

Https://gurus.pyimagesearch.com/

LOGIN (HTTPS://GURUS.PYIMAGESEARCH.COM/)

COMMUNITY (HTTP://COMMUNITY.PYIMAGESEARCH.COM/)

PyImageSearch Gurus Course

([HTTPS://GURUS.PYIMAGESEARCH.COM/](https://gurus.pyimagesearch.com/))  

0.1: Getting yourself oriented

Welcome to PyImageSearch
Gurus, Hieyong!

Course Progress

Ready to continue the course?

Click the button below to
continue your journey to
computer vision guru.

I'm ready,
let's go!
(/pyimagesearch-
gurus-
course/)

Resources & Links

- [PyImageSearch Gurus Community](#)
(<https://community.pyimagesearch.com/>)
- [PyImageSearch Virtual](#)

Search

how the course is laid out.

- Review the 4 steps to success in PyImageSearch Gurus.

Getting yourself oriented

One of the reasons I created the PyImageSearch Gurus course was because I wanted to *bridge the gap* between *college level coursework* and *actual, real-world computer vision applications*. In order to accomplish this, I needed to:

1. Describe advanced computer vision algorithms in an intuitive, easy to understand manner.
2. Provide (and explain) *actual source code* that you can use to perform various computer vision techniques.

You see, I wanted to build PyImageSearch Gurus such that each lesson was *practical, hands on*, and most importantly, *something you could read and take away actual value from*.

This orientation lesson is no different.

So, before I explain what the PyImageSearch Gurus course is and how it works, let's first have a bit of fun and ***learn how to detect faces in images***.

Detecting faces in images

Our goal in this section is to write Python + OpenCV code to detect the presence of a face in an image. Specifically, we'll be using this example image:



FIGURE 1: OUR INPUT IMAGE THAT WE ARE GOING TO DETECT FACES IN.

After our Python script runs, we should be able to (1) detect the face in the image and (2) draw a rectangle (i.e., a *bounding box*) surrounding the face. To see how we can do this, let's investigate some code:

detect_face.py	Python
<pre>1 # import the necessary packages 2 import cv2 3 4 # load our image and convert it to grayscale 5 image = cv2.imread("orientation_example.jpg") 6 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) 7 8 # load the face detector and detect faces in the image 9 detector = cv2.CascadeClassifier("haarcascade_frontalface_ 10 rects = detector.detectMultiScale(gray, scaleFactor=1.05, 11 minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE) 12 13 # loop over the faces and draw a rectangle surrounding each 14 for (x, y, w, h) in rects: 15 cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 16 17 # show the detected faces 18 cv2.imshow("Faces", image) 19 cv2.waitKey(0)</pre>	

Lines 5 and 6 load our input image from disk and convert it to grayscale. **Lines 9-11** handle the actual face detection process. Then, **Lines 14 and 15** draw the bounding box surrounding the

face in the image while **Lines 18 and 19** display our final output image.

We can run the `detect_face.py` script using the following command:

<code>detect_face.py</code>	Shell
1 \$ <code>python detect_face.py</code>	

Where we can see the following output:



FIGURE 2: WE HAVE SUCCESSFULLY DETECTED THE FACE IN THE INPUT IMAGE!

Notice how my face has been detected in the image — ***all with less than 20 lines of Python code!***

Pretty neat, right?

We'll be covering face detection and face recognition in ***much more detail*** later in the PyImageSearch Gurus course, but I just wanted to give you a *taste* of the practical, hands-on material you'll be learning.

PyImageSearch Gurus course structure

The PyImageSearch Gurus course is primarily comprised of three components:

Component #1: High-level **modules**, discussing topics such as *Automatic License Plate Recognition*, *Deep Learning*, *Face Recognition*, and **much more**:

Module 6: Automatic License Plate Recognition		
Lessons		Status
54	6.1: What is ANPR?	<input checked="" type="checkbox"/>
55	6.2: The problem with ANPR datasets	<input checked="" type="checkbox"/>
56	6.3: Localizing license plates in images	<input checked="" type="checkbox"/>
57	6.4: Segmenting characters from the license plate	<input checked="" type="checkbox"/>
58	6.5: Scissoring the license plate characters	<input checked="" type="checkbox"/>
59	6.6: Our first try at recognizing license plate characters	<input checked="" type="checkbox"/>
60	6.7: Gathering our own license plate characters	<input checked="" type="checkbox"/>
61	6.8: Improving our license plate classifier	<input checked="" type="checkbox"/>
62	6.9: Tips on classifying your own license plates	<input checked="" type="checkbox"/>

FIGURE 3: TOPICS INSIDE THE PYIMAGESEARCH GURUS COURSE ARE BROKEN INTO HIGHER-LEVEL MODULES.

There are **13 modules** inside PyImageSearch Gurus, so we'll be covering *a lot* of computer vision techniques.

Component #2: Each module is then broken down into **lessons** and **sub-lessons**:

Module 6: Automatic License Plate Recognition		
Lessons		Status
54	6.1: What is ANPR?	<input checked="" type="checkbox"/>
55	6.2: The problem with ANPR datasets	<input checked="" type="checkbox"/>
56	6.3: Localizing license plates in images	<input checked="" type="checkbox"/>
57	6.4: Segmenting characters from the license plate	<input checked="" type="checkbox"/>
58	6.5: Scissoring the license plate characters	<input checked="" type="checkbox"/>
59	6.6: Our first try at recognizing license plate characters	<input checked="" type="checkbox"/>
60	6.7: Gathering our own license plate characters	<input checked="" type="checkbox"/>
61	6.8: Improving our license plate classifier	<input checked="" type="checkbox"/>
62	6.9: Tips on classifying your own license plates	<input checked="" type="checkbox"/>

Every module consists of multiple lessons & sub-lessons

FIGURE 4: EACH MODULE CONSISTS OF MULTIPLE LESSONS AND SUB-LESSONS.

These lessons detail, piece-by-piece, how to accomplish the overall goal of the module. For example, lessons and sub-lessons inside our *Face Recognition* module explain how to extract features from a face, train a classifier, and then use this model to recognize faces in images and video streams.

Similarly, lessons inside the *Deep Learning* module start by explaining the basics of Neural Networks, then eventually move up to more advanced network architectures, such as Convolutional Neural Networks. Some modules have more lessons than others, depending on the complexity of the topic.

Component #3: Most lessons also include a **quiz** for you to test your knowledge:

Quizzes		Status
1	Loading, Displaying, and Saving Images Quiz	<input checked="" type="checkbox"/>

FIGURE 5: AN EXAMPLE OF A QUIZ APPEARING AT THE BOTTOM OF A LESSON.

You **do not** have to take each quiz to move on to the next lesson in the course — but you **do** have to take (and pass) each quiz to be eligible for the Certificate of Completion at the end of the course.

After you complete each lesson/associated quiz, you'll see a **green checkmark** appear next to it on the course homepage:

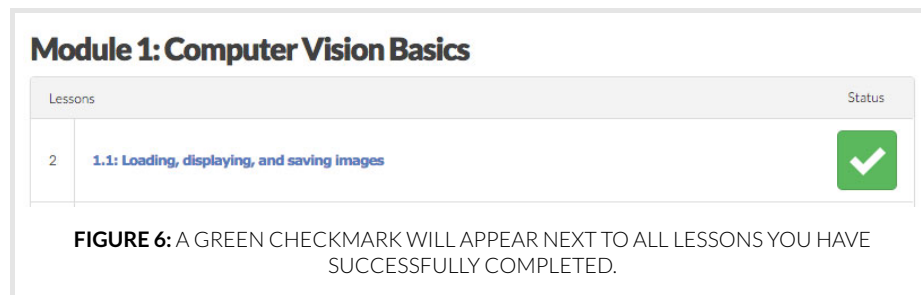
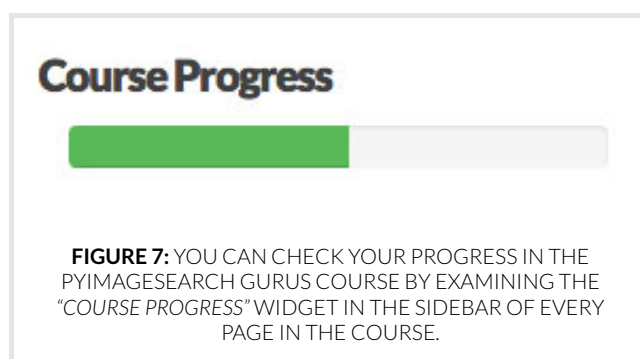


FIGURE 6: A GREEN CHECKMARK WILL APPEAR NEXT TO ALL LESSONS YOU HAVE SUCCESSFULLY COMPLETED.

You can also see your course progress on the upper-right sidebar of all pages in the course:



You'll also notice that after completing each lesson, you'll be awarded an *achievement* and corresponding set of points:

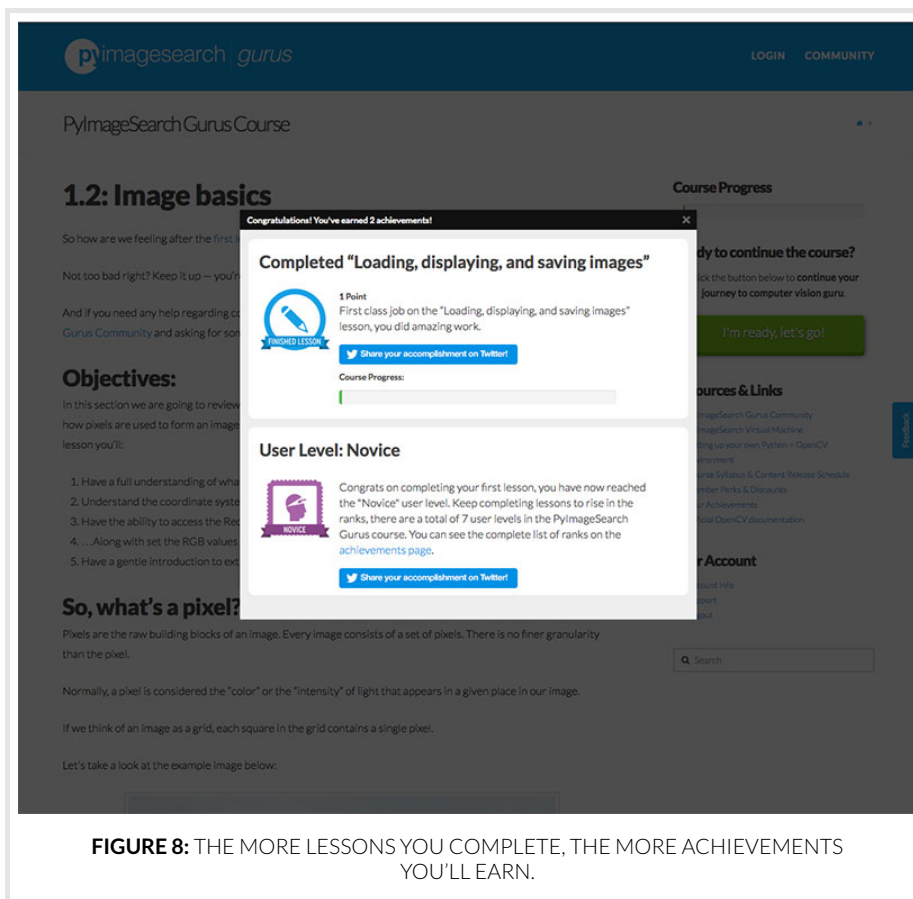


FIGURE 8: THE MORE LESSONS YOU COMPLETE, THE MORE ACHIEVEMENTS YOU'LL EARN.

The more lessons and modules you complete, the more points you are awarded, allowing you to move up in the ranks:

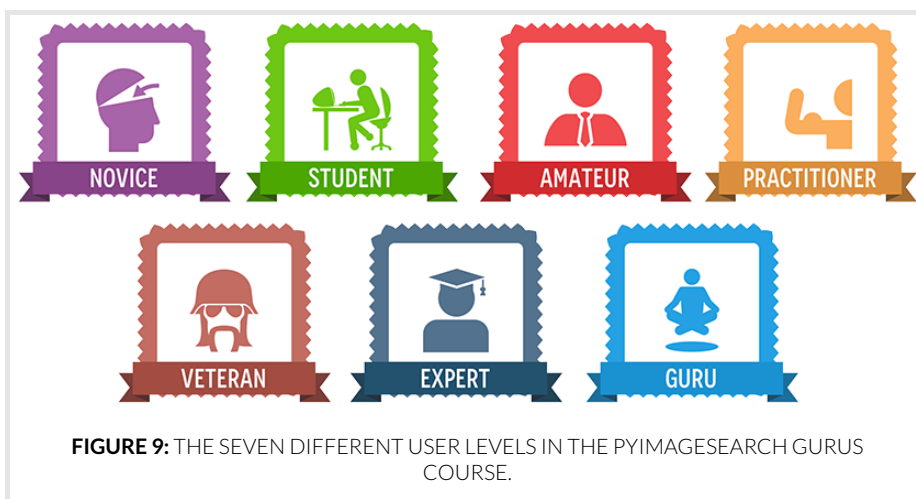


FIGURE 9: THE SEVEN DIFFERENT USER LEVELS IN THE PYIMAGESEARCH GURUS COURSE.

Your goal is to reach the **Guru** user level, which you can do by completing all lessons and quizzes inside the course.

After you have completed all lessons inside the course, you are awarded the *PyImageSearch Gurus Certificate of Completion*:

Adrian Rosebrock

successfully completed and passed all lessons associated with:

PyImageSearch Gurus

an actionable, real-world course on computer vision offered
by PyImageSearch.com

CERTIFICATE OF COMPLETION
Awarded 23 March, 2016

AWARDED BY
Adrian Rosebrock, Owner & Director of PyImageSearch Gurus

FIGURE 10: YOU'LL BE AWARDED THE PYIMAGESEARCH GURUS CERTIFICATE OF COMPLETION AFTER SUCCESSFULLY PASSING EVERY LESSON IN THE COURSE.

Which you can also embed on your LinkedIn profile page to demonstrate your completion of the course:



FIGURE 11: YOUR PYIMAGESEARCH GURUS CERTIFICATE OF COMPLETION CAN BE EMBEDDED INTO YOUR LINKEDIN PROFILE.

Syllabus and lesson release schedule

From a timeline perspective, the PyImageSearch Gurus course is meant to run for 6 months. Each month, based on your signup date, a new set of lessons will be released according to the **syllabus and content release schedule** (<https://gurus.pyimagesearch.com/course-syllabus-content-release-schedule/>).

This schedule is a *pre-defined timetable* that is *specifically designed* to help you learn computer vision in the most optimal, efficient manner.

If you purchased the ***instant access membership*** to PyImageSearch Gurus, the lesson release schedule does not apply to you — you have *immediate access* to *every lesson* inside PyImageSearch Gurus starting as soon as you finish this orientation.

4 steps to success in PyImageSearch Gurus

One of my favorite quotes is from former United States Senator, Byron Dorgan, who said:

“Working harder and working smart sometimes can be two different things.”

Not only is this great life advice, but it’s also *extremely applicable* to studying and teaching yourself a new skill. Sitting in front of a screen, book, or educational video for hours on end may teach you something — *but it might not be the optimal way to learn*.

In order to help you *study smarter* and make the most of your time invested in PyImageSearch Gurus, I've created the **4 steps to success inside PyImageSearch Gurus**.

#1. Setup your development environment

It goes without saying that we'll be doing a lot of programming and developing inside this course. In order to work through our lessons, we need to have a development environment setup and configured to use various packages and libraries, including Python, OpenCV, scikit-learn, Keras, *and many more*.

To get you on the path to PyImageSearch Guru ASAP, I have created a **downloadable Ubuntu virtual machine** (<https://gurus.pyimagesearch.com/pyimagesearch-virtual-machine/>) that comes with all the necessary libraries and packages you'll need inside this course pre-configured and pre-installed. All you have to do is download the virtual machine, set it up, and you'll be up and running in a matter of minutes:

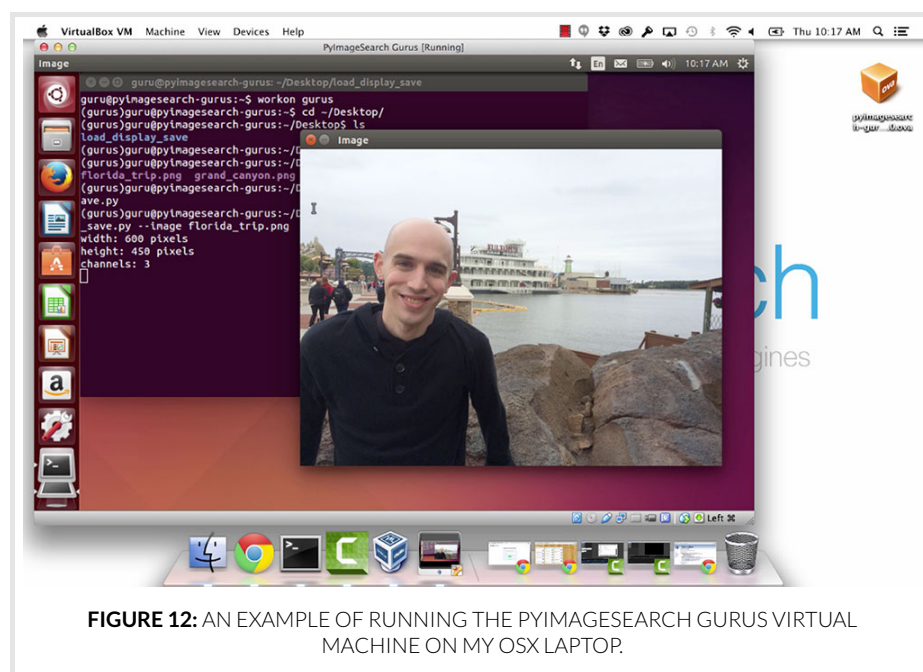


FIGURE 12: AN EXAMPLE OF RUNNING THE PYIMAGESEARCH GURUS VIRTUAL MACHINE ON MY OSX LAPTOP.

That said, if you want to use your own native system, I have created **setup instructions** (<https://gurus.pyimagesearch.com/setting-up-your-python-opencv-development-environment/>) for both OSX and

Ubuntu (and other Debian-based systems). Using these instructions, you can setup your own development environment or even recreate the virtual machine I mentioned above.

If you are a Windows user, **I highly recommend that you use either the Ubuntu virtual machine or configure your own Unix-based system.** Computer vision development is best performed on a Unix machine, so definitely consider this when configuring your environment. If you choose to use Windows inside the course, I and the PyImageSearch Gurus community will only be able to give limited advice and suggestions if you run into problems.

#2. Introduce yourself

PyImageSearch Gurus is more than just a computer vision course — it's also a *community* of like-minded developers, researchers, and students.

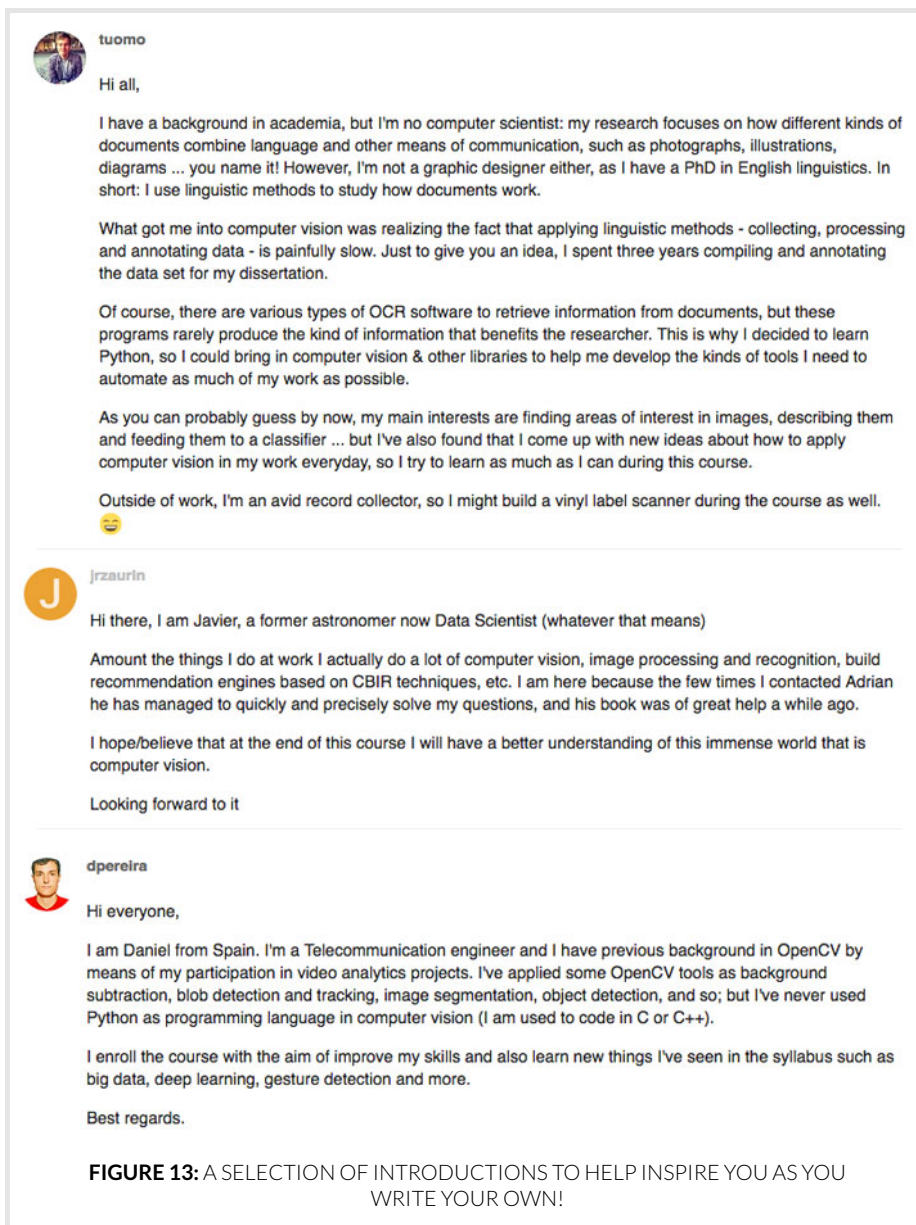
Your next step in the orientation process is to head over to the **PyImageSearch Gurus Community**

(<https://community.pyimagesearch.com/>) and introduce yourself in the **Introductions thread**

(<https://community.pyimagesearch.com/t/introduce-yourself/20>). Tell us who you are, what your background is, and why you decided to join the PyImageSearch Gurus course.

Go ahead, don't be shy!

To help inspire you, I've included a few of my favorite introductions below:



As you can see, some introductions are short and to the point. Others are longer and more detailed. But they all accomplish the same thing — *they tell us who you are and why you joined!*

I know it may sound silly, like you're back in high school going through orientation for the first time — **but we're family here.** We should all get to know each other.

So go ahead and click here to introduce yourself.
(<https://community.pyimagesearch.com/t/introduce-yourself/20>)

#3. Create an accountability thread

Now that you've introduced yourself, you should also create an **Accountability thread**

(<https://community.pyimagesearch.com/c/accountability>).

Inside this thread, detail your **goals** and **what you hope to achieve after finishing this course**.

To create an Accountability thread, **just click this link** (<https://community.pyimagesearch.com/c/accountability>) to head to the Accountability section of the community, then click the **New Topic** button:

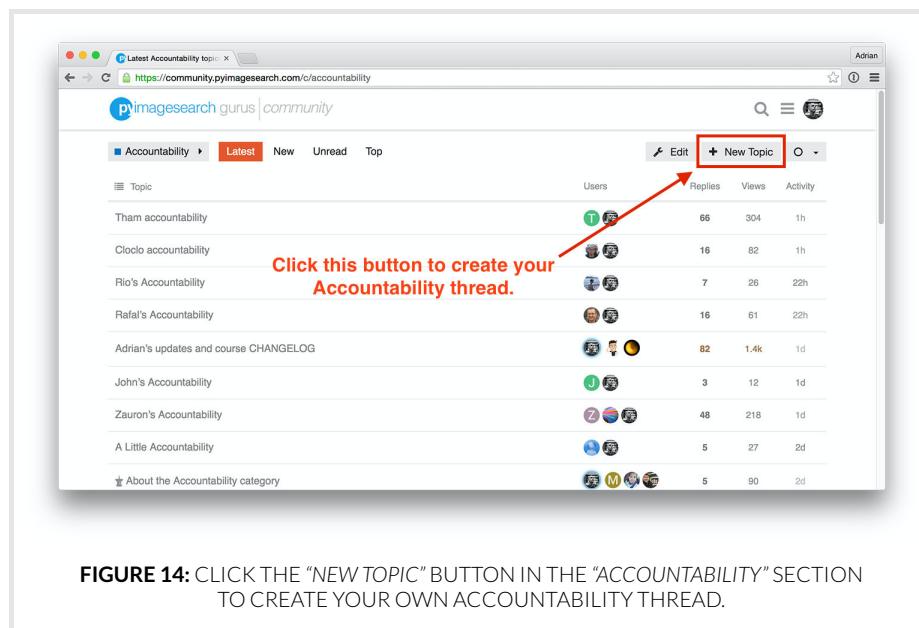


FIGURE 14: CLICK THE “NEW TOPIC” BUTTON IN THE “ACCOUNTABILITY” SECTION TO CREATE YOUR OWN ACCOUNTABILITY THREAD.

Give your thread a name, such as “*Hieyong’s accountability*” and list out your goals for the week. I recommend starting with (at a bare minimum) two goals:

1. Configuring your development environment.
2. Going through Lessons 1.1-1.6.

Make sure you keep this thread updated each week as you work through the course!

I have taken *many* online courses in my life, and I can tell you that *no other technique* has motivated me to work harder and faster than publicly posting my updates and having others keep

me accountable — I think the same will be true for you as well.

Make a habit of posting in your accountability once per week with your updates. You can find an [example of my accountability thread here \(https://community.pyimagesearch.com/t/adrians-updates-and-course-changelog/21\)](https://community.pyimagesearch.com/t/adrians-updates-and-course-changelog/21) if you need inspiration.

#4. Get started!

This step seems so obvious, but too many times I feel that it's left unsaid. Each day I get at least 20-30 emails from students who are *interested in computer vision* but *don't know where to start*. That's a fair question — but my reply is always the same 3-step process:

1. Install OpenCV.
2. Go through a few basic tutorials on the PyImageSearch blog (which I of course link them to).
3. Identify what you *did* and *did not* like as you went through the tutorials — use this to guide your future explorations in computer vision.

Contrary to what you may read on other forums, blogs, or websites, you *do not need* a degree in computer science to learn computer vision. You *do not need* an in-depth understanding of mathematics to understand computer vision. And you sure as hell *do not need* an intensive college-level education to comprehend computer vision algorithms.

All you need to do is simply *get started*.

Getting started is *by far* the most important step — and that's exactly why you're here inside this course. I'm honestly incredibly excited (and honored) to be here and help you on your journey to become a computer vision guru.

BONUS: Watch my changelog

As I fix bugs on the website, update lessons, and release new features, I'll be using [my accountability thread](https://community.pyimagesearch.com/t/adrians-updates-and-course-changelog/21) (<https://community.pyimagesearch.com/t/adrians-updates-and-course-changelog/21>) as a changelog. Be sure to keep an eye on this thread for more insider updates and behind-the-scenes action.

Have fun. Learn a lot. And enjoy the course!

Summary

Congratulations on completing your first lesson, Hieyong!

Let's review what you accomplished.

First, you learned how to detect faces in images using OpenCV. We then discussed the PyImageSearch Gurus course structure and how it's divided into three primary components:

1. Modules
2. Lessons (and sub-lessons)
3. Quizzes

You **do not** need to take each quiz to move on to the next lesson — but you **do** need to take (and pass) each quiz to be eligible for the Certificate of Completion.

Finally, we reviewed the 4 steps to success inside PyImageSearch Gurus.

Now that you've completed this lesson, let's move on to the next step — **taking your first quiz**. Click the link below to start the quiz (don't worry, there are no wrong answers). After you finish the quiz, you'll be free to start the course!

Downloads:

[Download the Code](#)

(https://gurus.pyimagesearch.com/protected/code/orientation/getting_y

Quizzes	Status
1	<p data-bbox="188 521 555 555">Getting Yourself Oriented Quiz</p> <p data-bbox="188 633 817 779">(https://gurus.pyimagesearch.com/quizzes/getting-yourself-oriented-quiz/)</p>

[Next Lesson →](#)

(<https://gurus.pyimagesearch.com/lessons/loading-displaying-and-saving-images/>).