



# Building a Pokedex in Python: Getting Started (Step 1 of 6)

by **Adrian Rosebrock** on March 10, 2014 in **Building a Pokedex, Examples of Image Search Engines, Tutorials**

1

1



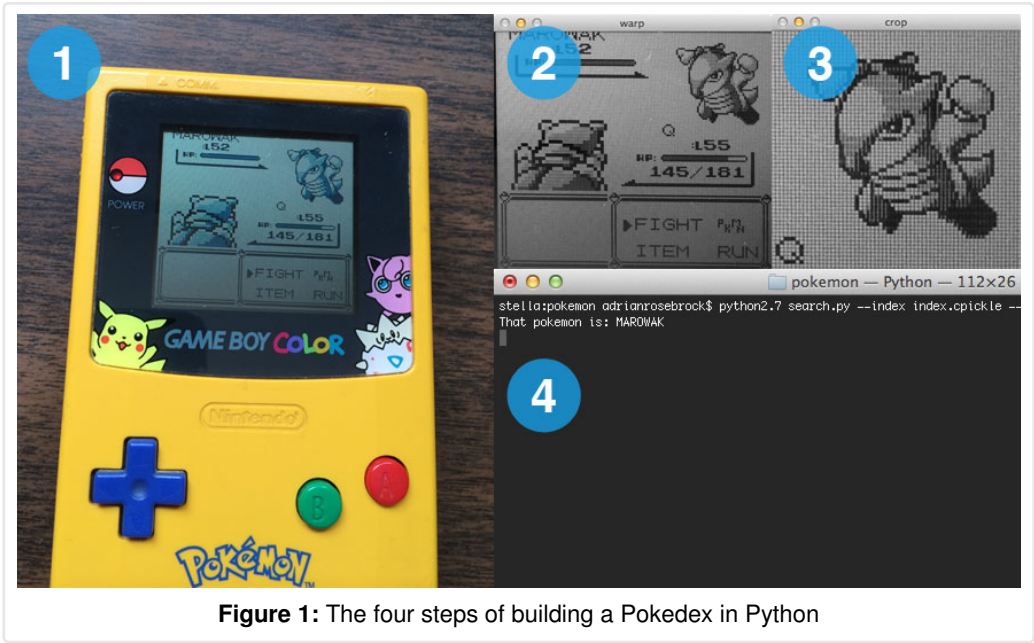
“ So you’re telling me that we are going to build a Pokedex in Python that can **visually** recognize Pokemon? Just like in the TV show? – Ash Ketchum

Yes, Ash. That’s exactly what I’m telling you. The next series of blog posts will be dedicated to building a Pokedex that can visually recognize Pokemon from the original Red, Blue, and Green (JP) games for the Game Boy.

Don’t believe me? Check out this video:

Building a Real life Pokedex using Python and OpenCV

## Motivation: Visually Recognizing Pokemon



I’ve been playing Pokemon (the original Red and Blue) since I was a little kid. I love these games and I know them front to back. In fact, every year my friend and I hold the *Pokethon*. We drink some beer and do a speed run of Pokemon Blue as fast as we can. This past year I managed to beat the entire game in 2 hours, 46 minutes. Was I pretty happy with that achievement? You bet. But there’s always room for improvement. Next year’s goal is sub 2 hours, 30 minutes.

I’m getting a little off topic here. A few years back I decided that I wanted to beat Pokemon Green. The Green version was only available in Japan, and as you might guess, all the text is in Japanese. I can’t read Japanese, but I still managed to get through the game.

And I thought to myself, this is what Ash Ketchum must have felt like.

Here I am, playing this game. I can’t read the text, I can’t follow the actual story of the game. But through random encounters and battles with rivals, I’m able to see and experience a lot of different Pokemon.

Now, I obviously know what each Pokemon is based on their sprite images. But let’s pretend that I didn’t.

How might I identify these Pokemon?

And how might I easily and conveniently lookup any stats related to the Pokemon?

Perhaps the most obvious strategy would be to go to Google and type in the name of the Pokemon in Japanese (character by character, according to what is on my screen), translate it, and then lookup the stats in english.

But we study computer vision on this blog. So instead we’re going to build a computer vision solution.

## Your Phone is Your Pokedex

Free 21-day crash course on computer vision & image search engines



Figure 2: An example query image containing a Game Boy screen and Pokemon.

If you’re anything like me, you have your phone with you wherever you go. It’s probably sitting next to you when you are playing Pokemon, just in case you get a txt or a call.

And in all likelihood, your phone has a camera.

So here’s what’s going to happen:

1. We are playing Pokemon on our Game Boy (either, Red, Blue, or Green version) when we encounter a Pokemon that we do not know.
2. We would like to identify this Pokemon, so we take out our phone and snap a photo of our Game Boy screen.
3. We then take this picture, apply our computer vision and image search engine techniques, and we are able to automatically identify the Pokemon.

Again, if you don’t believe me, just take a look at the video above.

## How are we going to do this?

In order to even “find” the Game Boy screen in our query image we are first going to make use of many image processing and computer vision techniques. For example, we’ll be using techniques such as:

- Edge detection
- Thresholding
- Perspective warping

After we have our Game Boy screen, we’ll crop out the Pokemon and then *describe* the Pokemon. We’ll characterize the shape of the Pokemon by using shape descriptors. In this way we can abstractly represent the Pokemon using only a list of numbers (i.e. an image feature vector).

Don’t worry. I’ll go through every and every detail as we work through this series. By the time we are finished, you’ll have a firm understanding of how to use shape descriptors to describe objects in images. And in this case, the “objects” will be Pokemon.

## What’s next?

Building our Pokedex will be a six step process:

1. An introduction to building a Pokedex (that’s this blog post)
2. Scraping the web and building our Pokemon sprite database
3. Indexing our database using shape descriptors

Free 21-day crash course on computer vision & image search engines

- 4. Finding the Game Boy screen in our image (Part 1)
- 5. Finding the Game Boy screen (Part 2)
- 6. Searching and identifying the Pokemon

## Libraries You Will Need

I will assume that you have the following libraries installed:

- [NumPy and SciPy](#)
- [OpenCV](#) with Python bindings
- [Scikit-Image](#)
- [requests](#)
- [BeautifulSoup](#)

Don't want spend a bunch of time installing these packages?

No problem!

I have a pre-configured Ubuntu virtual machine with all the necessary computer vision and image processing pre-installed!

[Click here](#) and check out the eBook + virtual machine package to learn more.

## Gotta Identify 'Em All

Are you ready to embark on your very own Pokemon identification journey?

I know I am.



If you would like to receive an email update when new blog posts in this series are published, please enter your email address in the form below. Not only will you get a notification sent directly to your inbox, I'll also send you a **FREE 11-page Resource Guide** on Computer Vision and Image Search Engines, including **exclusive techniques** that I don't post on this blog! Sound good? If so, enter your email address and below and I'll be sure to let you know when new posts are published.

Email address:

Your email address

SIGN UP

### Resource Guide (it's totally free).



Enter your email address below to get my **free 11-page Image Search Engine Resource Guide PDF**. Uncover **exclusive techniques** that I don't publish on this blog and start building image search engines of your own!

Your email address

DOWNLOAD THE GUIDE!

💡 **blue version, database, green version, image processing, pokedex, pokemon, red version, scraping, shape**

< Charizard Explains How To Describe and Quantify an Image Using Feature Vectors

Building a Pokedex in Python: Scraping the Pokemon Sprites (Step 2 of 6) >

### 7 Responses to *Building a Pokedex in Python: Getting Started (Step 1 of 6)*



**Aman Khatri** August 23, 2016 at 1:25 am #

Free 21-day crash course on computer vision & image search engines

REPLY ↩



▼ ▼ OFF the topic:

Hi Adrian,I want to build a code which takes user input and gives output accordingly.the code should display the whole color chart,the user selects the particular color and gives a image as an input,output contains only that particular color displayed in the image,rest all color are either black or grey.How should i approach the code?Thanks in advance.



**Adrian Rosebrock** August 24, 2016 at 12:20 pm #

REPLY ↩

In essence, it sounds like you are trying to apply [color thresholding](#). I would suggest starting there and building the app as a command line application. You can then find a GUI framework that you like using (I'm not a GUI developer). TKinter, Qt, and maybe even Kivy would be good choices.

### Trackbacks/Pingbacks

- [Building a Pokedex in Python: Scraping the Pokemon Sprites \(Step 2 of 6\) - PyImageSearch](#) - March 24, 2014  
[...] Step 1: Building a Pokedex in Python: Getting Started (Step 1 of 6) [...]
- [Building a Pokedex in Python: Indexing our Sprites using Shape Descriptors \(Step 3 of 6\) - PyImageSearch](#) - April 7, 2014  
[...] Step 1: Building a Pokedex in Python: Getting Started (Step 1 of 6) [...]
- [Building a Pokedex in Python: Finding the Game Boy Screen \(Step 4 of 6\) - PyImageSearch](#) - April 21, 2014  
[...] Step 1: Building a Pokedex in Python: Getting Started (Step 1 of 6) [...]
- [Python and OpenCV Example: Warp Perspective and Transform](#) - May 5, 2014  
[...] Step 1: Building a Pokedex in Python: Getting Started (Step 1 of 6) [...]
- [Comparing Shape Descriptors for Similarity using Python and OpenCV](#) - May 19, 2014  
[...] explored what it takes to build a Pokedex using computer vision. Then we scraped the web and built up a database of Pokemon. We've indexed our database of [...]

### Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

### Resource Guide (it's totally free).

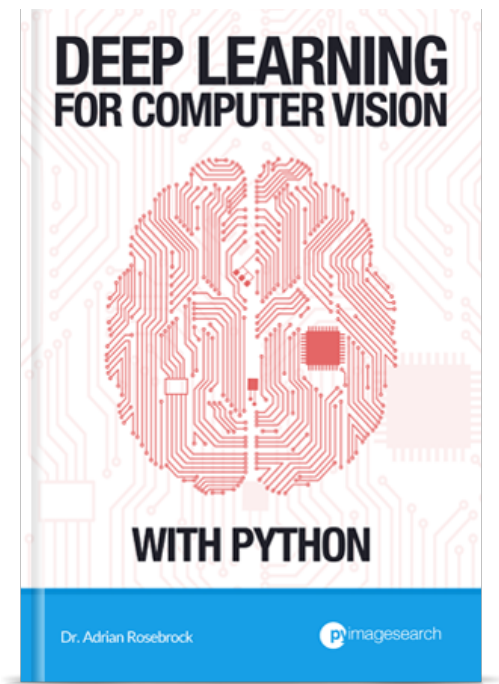


Click the button below to get my **free 11-page Image Search Engine Resource Guide PDF**. Uncover **exclusive techniques** that I don't publish on this blog and start building image search engines of your own.

Download for Free!

### Deep Learning for Computer Vision with Python Book

Free 21-day crash course on computer vision & image search engines



You're interested in deep learning and computer vision, *but you don't know how to get started*. Let me help. [My new book will teach you all you need to know about deep learning.](#)

CLICK HERE TO PRE-ORDER MY NEW BOOK

You can detect faces in images & video.



Are you interested in **detecting faces in images & video**? But **tired of Googling for tutorials** that *never work*? Then let me help! I guarantee that my new book will turn you into a **face detection ninja** by the end of this weekend. [Click here to give it a shot yourself.](#)

CLICK HERE TO MASTER FACE DETECTION

PyImageSearch Gurus: NOW ENROLLING!

The PyImageSearch Gurus course is *now enrolling!* Inside the course you'll learn how to perform:

- Automatic License Plate Recognition (ANPR)
- Deep Learning
- Face Recognition
- *and much more!*

Click the button below to learn more about the course, take a tour, and get 10 (FREE) sample lessons.

TAKE A TOUR & GET 10 (FREE) LESSONS

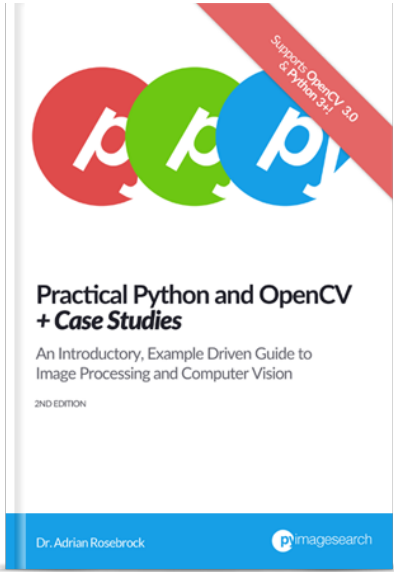
Hello! I'm Adrian Rosebrock.

Free 21-day crash course on computer vision & image search engines



I'm an entrepreneur and Ph.D who has launched two successful image search engines, [ID My Pill](#) and [Chic Engine](#). I'm here to share my tips, tricks, and hacks I've learned along the way.

Learn computer vision in a single weekend.



Want to learn computer vision & OpenCV? I can teach you in a **single weekend**. I know. It sounds crazy, but it's no joke. My new book is your **guaranteed, quick-start guide** to becoming an OpenCV Ninja. So why not give it a try? [Click here to become a computer vision ninja](#).

CLICK HERE TO BECOME AN OPENCV NINJA

Subscribe via RSS



**Never miss a post!** Subscribe to the PyImageSearch RSS Feed and keep up to date with my image search engine tutorials, tips, and tricks

POPULAR
<b>Install OpenCV and Python on your Raspberry Pi 2 and B+</b> FEBRUARY 23, 2015
<b>Home surveillance and motion detection with the Raspberry Pi, Python, OpenCV, and Dropbox</b> JUNE 1, 2015
<b>Install guide: Raspberry Pi 3 + Raspbian Jessie + OpenCV 3</b> APRIL 18, 2016
<b>How to install OpenCV 3 on Raspbian Jessie</b> OCTOBER 26, 2015
<b>Basic motion detection and tracking with Python and OpenCV</b> MAY 25, 2015
<b>Accessing the Raspberry Pi Camera with OpenCV and Python</b> MARCH 30, 2015
<b>Install OpenCV 3.0 and Python 2.7+ on Ubuntu</b> JUNE 22, 2015

Search

Search...

Q

Find me on [Twitter](#), [Facebook](#), [Google+](#), and [LinkedIn](#).  
© 2017 PyImageSearch. All Rights Reserved.

Free 21-day crash course on computer vision & image search engines