

PYIMAGESEA
RCH

[Click here to download the source code to this post](#)
<https://app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi>)

[DEEP LEARNING \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/DEEP-LEARNING/\)](#).

[DLIB \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/DLIB/\)](#).

[FACE APPLICATIONS \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/FACES/\)](#).

[TUTORIALS \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/TUTORIALS/\)](#).

Face clustering with Python

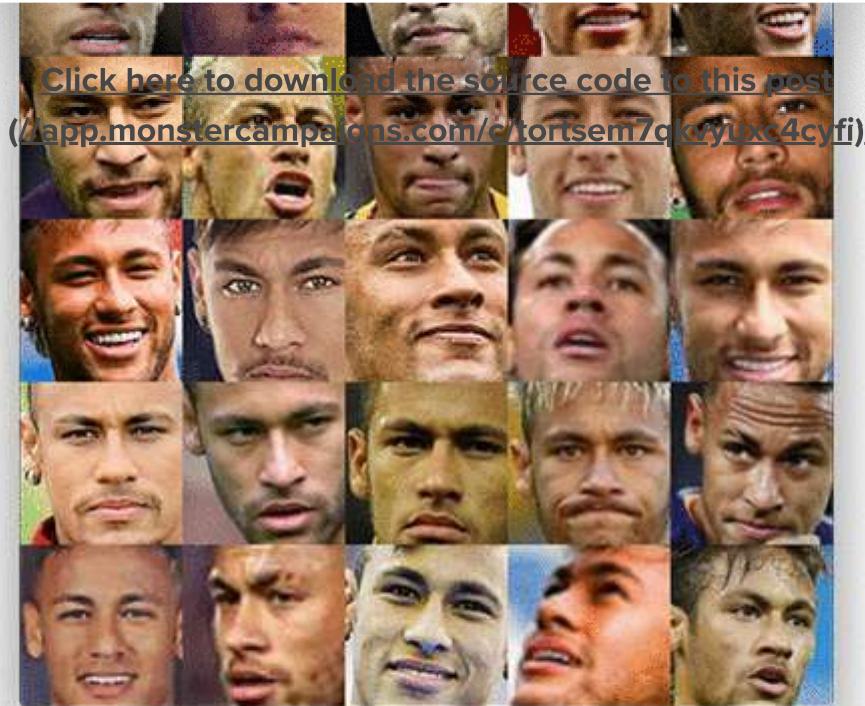
by [Adrian Rosebrock](#) (<https://www.pyimagesearch.com/author/adrian/>) on July 9, 2018

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.



Today's blog post is inspired by a question from PyImageSearch reader, Leonard Bogdonoff.

After I published my previous post on [Face recognition with OpenCV and deep learning](https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/) (<https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>), Leonard wrote in and asked:

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

The perpetrators flee the bank with the cash hidden under their clothes, taking off their masks, and dumping them in nearby trash can as to not appear “suspicious” in public.

[Click here to download the source code to this post](#)

Will they get away with the crime? (<http://app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi>)

Maybe.

But security cameras installed at nearby gas stations, restaurants, and red lights/major intersections capture all pedestrian activity in the neighborhood.

After the police arrive their detectives could leverage face clustering to find all unique faces across *all* video feeds in the area — given the unique faces, detectives could: (1) manually investigate them and compare them to bank teller descriptions, (2) run an automated search to compare faces to a known database of criminals, or (3) apply good ole’ detective work and look for suspicious individuals.

This is a fictitious example of course, but I hope you see the value in how face clustering could be used in real-world situations.

To learn more about face clustering, and how to implement it using Python, and deep learning, just keep reading.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

faces we want to recognize along with (2) the names that correspond to each face (i.e., the “class labels”).

[Click here to download the source code to this post](#)

([//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi](http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi))

But in *face clustering* we need to perform **unsupervised learning** — we have only the *faces themselves* with *no names/labels*. From there we need to *identify* and *count* the number of unique people in a dataset.

In the first part of this blog post, we’ll discuss our face clustering dataset and the project structure we’ll use for building the project.

From there I’ll help you write two Python scripts:

- 1 One to extract and quantify the faces in a dataset
- 2 And another to cluster the faces, where each resulting cluster (ideally) represents a unique individual

From there we’ll run our face clustering pipeline on a sample dataset and examine the results.

Configuring your development environment

In our previous [face recognition \(<https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>\)](https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/) post, I explained how to configure your development

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a free 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!

If you have a GPU, you'll want to install dlib with CUDA bindings which is also described in [this previous post](#) (<https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>). (<http://app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi>)

Our face clustering dataset



[\(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_dataset.jpg\)](https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_dataset.jpg)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

- Luis Suarez (https://en.wikipedia.org/wiki/Luis_Su%C3%A1rez) [Click Here to download the source code to this post](#) (<http://app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi>)

In total, there are 129 images in the dataset.

Our goal will be to extract features quantifying each face in the image and cluster the resulting “facial feature vectors”. Ideally each soccer player will have their own respective cluster containing just their faces.

Face clustering project structure

Before we get started, be sure to grab the downloadable zip from the “**Downloads**” section of this blog post.

Our project structure is as follows:

```
Face clustering with Python
1. | $ tree --dirsfirst
2. |
3. |   dataset [129 entries]
4. |     00000000.jpg
5. |     00000001.jpg
6. |     00000002.jpg
7. |     ...
8. |     00000126.jpg
9. |     00000127.jpg
10. |    00000128.jpg
```

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

- encode_faces.py : This is our first script — it computes face embeddings for all faces in the dataset and outputs a serialized encodings file.
[Click here to download the source code to this post](#)
- encodings.pickle : Our face embeddings serialized pickle file.
- cluster_faces.py : The magic happens in this script where we'll cluster similar faces and ideally find the outliers.

Encoding faces via deep learning



[-0.08, 0.45, ..., -0.97]

(<https://pyimagesearch.com/wp->

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

[and-deep-learning\(\)](#), so be sure to refer to it if you have any questions on the network itself. Our `encode_faces.py` script will contain all code used to extract a 128-d feature vector representation for each face.

Click here to download the source code to this post (<https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

To see how this process is performed, create a file named `encode_faces.py`, and insert the following code:

```
Face clustering with Python
1. # import the necessary packages
2. from imutils import paths
3. import face_recognition
4. import argparse
5. import pickle
6. import cv2
7. import os
8.
9. # construct the argument parser and parse the arguments
10.ap = argparse.ArgumentParser()
11.ap.add_argument("-i", "--dataset", required=True,
12.    help="path to input directory of faces + images")
13.ap.add_argument("-e", "--encodings", required=True,
14.    help="path to serialized db of facial encodings")
15.ap.add_argument("-d", "--detection-method", type=str, default="cnn",
16.    help="face detection model to use: either `hog` or `cnn`")
17.args = vars(ap.parse_args())
```

Our required packages are imported on **Lines 2-7**. Take note of:

- `paths` from my [imutils \(https://github.com/jrosebr1/imutils/blob/master/imutils/paths.py\)](https://github.com/jrosebr1/imutils/blob/master/imutils/paths.py)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Histogram of Oriented Gradients (HOG) method to detect the faces in an input image prior to quantifying the face. ~~The CNN method is more accurate (but slower), whereas the HOG method is faster (but less accurate).~~ [Click here to download the source code to this post \(//app-monstercampaigns.com/c/tortsem7qkvyuxc4cyfi\)](#)

If you're unfamiliar with command line arguments and how to use them, please refer to my [previous post \(\)](#).

I'll also mention that if you think this script is running slow or you would like to run the face clustering post in real-time without a GPU you should *absolutely* be setting

--detection-method to hog instead of cnn . While the CNN face detector is more accurate, it's *far* too slow to run in real-time without a GPU.

Let's grab the paths to all the images in our input dataset:

```
Face clustering with Python
19.  # grab the paths to the input images in our dataset, then initialize
20.  # out data list (which we'll soon populate)
21.  print("[INFO] quantifying faces...")
22.  imagePaths = list(paths.list_images(args["dataset"]))
23.  data = []
```

On **Line 22**, we create a list of all imagePaths in our dataset using the dataset path provided in our command line argument.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

32). Then we swap color channels in the image because dlib assumes `rgb` ordering rather than OpenCV's default `BGR`. (Line 33).

[Click here to download the source code to this post](https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)

([//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi](https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi))

Now that the image has been processed, let's detect all the faces and grab their bounding box coordinates:

```
Face clustering with Python
35. |     # detect the (x, y)-coordinates of the bounding boxes
36. |     # corresponding to each face in the input image
37. |     boxes = face_recognition.face_locations(rgb,
38. |         model=args["detection_method"])
```

We must *detect* the actual location of a face in an image *before* we can quantify it. This detection takes place on **Lines 37 and 38**. You'll notice that the `face_recognition` API is very easy to use.

Note: We are using the CNN face detector for higher accuracy, but it will take a significantly longer time to run if you are using a CPU rather than a GPU. If you want the encoding script to run faster or your system, and your system does not have enough RAM or CPU power for the CNN face detector, use the HOG + Linear SVM method instead.

Let's get to the "meat" of this script. In the next block, we'll compute the facial encodings:

```
Face clustering with Python
40. |     # compute the facial embedding for the face
41. |     encodings = face_recognition.face_encodings(rgb, boxes)
```

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

3 Line 128-a encoding itself

[Click here to download the source code to this post](#)

Then we add the dictionary to our data list (Line 47). We'll use this information later when we want to visualize which faces belong to which cluster.

To close out this script, we simply write the data list to a serialized pickle file:

```
Face clustering with Python
49. | # dump the facial encodings data to disk
50. | print("[INFO] serializing encodings...")
51. | f = open(args["encodings"], "wb")
52. | f.write(pickle.dumps(data))
53. | f.close()
```

Using our command line argument, `args ["encodings"]`, as the path + filename, we write the data list to disk as a serialized pickle file (**Line 51-53**).

Running the face encoding script

Before proceeding, scroll to the “**Downloads**” section to download code + images. You may elect to use your own dataset of images — that’s totally fine too, just be sure to provide the appropriate path in the command line arguments.

Then, open a terminal and activate your Python virtual environment (if you are using one) containing the libraries and packages you installed earlier in this post.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

```
14. | [INFO] processing image 129/129
15. | dataset/00000035.jpg
16. | [INFO] serializing encodings...
```

[Click here to download the source code to this post](#)

(<http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

This process can take a while and you can track the progress with the terminal output.

If you're working with a GPU it will execute in quickly — in the order of 1-2 minutes. Just be sure that you installed dlib with CUDA bindings to take advantage of your GPU (as I mentioned above and described [in this post](https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/) (<https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>)).

However, if you're just executing the script on your laptop with a CPU, the script may take 20-30 minutes to run.

Clustering faces

Now that we have quantified and encoded all faces in our dataset as 128-d vectors, the next step is to *cluster* them into groups.

Our hope is that each unique individual person will have their own separate cluster.

The problem is, many clustering algorithms such as [k-means](https://en.wikipedia.org/wiki/K-means_clustering) (https://en.wikipedia.org/wiki/K-means_clustering) and [Hierarchical Agglomerative Clustering](https://en.wikipedia.org/wiki/Hierarchical_clustering) (https://en.wikipedia.org/wiki/Hierarchical_clustering), require us to specify the number of clusters we seek *ahead of time*.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

[\(https://en.wikipedia.org/wiki/Chinese_Whispers_\(clustering_method\)\)](https://en.wikipedia.org/wiki/Chinese_Whispers_(clustering_method))

[Click here to download the source code to this post](#)

We'll be using DBSCAN (<https://en.wikipedia.org/w/index.php?title=DBSCAN&oldid=749101741>) massive datasets you should consider using the Chinese whispers algorithm as it's linear in time.

The DBSCAN algorithm works by grouping points together that are closely packed in an N -dimensional space. Points that lie close together will be grouped together in a single cluster.

DBSCAN also naturally handles outliers, marking them as such if they fall in low-density regions where their "nearest neighbors" are far away.

Let's go ahead and implement face clustering using DBSCAN.

Open up a new file, name it `cluster_faces.py`, and insert the following code:

```
Face clustering with Python
1. # import the necessary packages
2. from sklearn.cluster import DBSCAN
3. from imutils import build_montages
4. import numpy as np
5. import argparse
6. import pickle
7. import cv2
8.
9. # construct the argument parser and parse the arguments
10. ap = argparse.ArgumentParser()
11. ap.add_argument("-e", "--encodings", required=True,
12.     help="path to serialized db of facial encodings")
13. ap.add_argument("-i", "--jobs", type=int, default=-1.
```

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

- `--jobs` : DBSCAN is multithreaded and a parameter can be passed to the constructor containing the number of parallel jobs to run. A value of `-1` will use all CPUs available (and is also the default for this command line argument).

Let's load the face embeddings data:

```
Face clustering with Python
17.  | # load the serialized face encodings + bounding box locations from
18.  | # disk, then extract the set of encodings to so we can cluster on
19.  | # them
20.  | print("[INFO] loading encodings...")
21.  | data = pickle.loads(open(args["encodings"], "rb").read())
22.  | data = np.array(data)
23.  | encodings = [d["encoding"] for d in data]
```

In this block we've:

- Loaded the facial encodings data from disk (**Line 21**).
- Organized the data as a NumPy array (**Line 22**).
- Extracted the 128-d encodings from the data , placing them in a list (**Line 23**).

Now we can cluster the encodings in the next code block:

Face clustering with Python

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Now let's determine the unique faces found in the dataset!

[Click here to download the source code to this post](#)

Referring to **Line 31**, `(/app/mustercampaigns.com/tensorflow/face/4.pyf)` contains the label ID for all faces in our dataset (i.e., which cluster each face belongs to). To find the unique faces/unique label IDs, we simply use NumPy's `unique` function. The result is a list of unique `labelIDs`.

On **Line 32** we count the `numUniqueFaces`. There could potentially be a value of `-1` in `labelIDs` — this value corresponds to the “outlier” class where a 128-d embedding was too far away from any other clusters to be added to it. Such points are called “outliers” and could either be worth examining or simply discarding based on the application of face clustering.

In our case, we excluded negative `labelIDs` in this count because we know for a fact that our dataset only contains images of 5 people. Whether or not you make such assumptions is highly dependent on your project.

The goal of our *next three code blocks* is to generate face montages of the unique soccer/futbol players in our dataset.

We begin the process by looping over all of the unique `labelIDs`:

```
Face clustering with Python
35. | # loop over the unique face integers
36. | for labelID in labelIDs:
37. |     # find all indexes into the `data` array that belong to the
38. |     # current label ID, then randomly sample a maximum of 25 indexes
39. |     # from the set
```

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

```

49.     for i in idxs:
50.         # load the input image and extract the face ROI
51.         image = cv2.imread(data[i][imagePath])
52.         (top, right, bottom, left) = data[i]['face']
53.         face = image[top:bottom, left:right]
54.
55.         # force resize the face ROI to 96x96 and then add it to the
56.         # faces montage list
57.         face = cv2.resize(face, (96, 96))
58.         faces.append(face)

```

We begin looping over all `idxs` in our random sample on **Line 49**.

Inside the first part of the loop, we:

- Load the `image` from disk and extract the `face` ROI (**Lines 51-53**) using the bounding box coordinates found during our face embedding step.
- Resize the face to a fixed 96×96 (**Line 57**) so we can add it to the `faces` montage (**Line 58**) used to visualize each cluster.

To finish out our top-level loop, let's build the montage and display it to the screen:

```

    Face clustering with Python
60.     # create a montage using 96x96 "tiles" with 5 rows and 5 columns
61.     montage = build_montages(faces, (96, 96), (5, 5))[0]
62.
63.     # show the output montage
64.     title = "Face ID #{}".format(labelID)

```

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

cluster montage.

[Click here to download the source code to this post](#)

Face clustering results

Be sure to use the “**Downloads**” section of this blog post to download the code and data necessary to run this script.

This script requires just one [command line argument](#)

(<https://pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/>) — the path to the encodings file. To perform face clustering for soccer/futbol players, just enter the following command in your terminal:

```
Face clustering with Python
1. | $ python cluster_faces.py --encodings encodings.pickle
2. | [INFO] loading encodings...
3. | [INFO] clustering...
4. | [INFO] # unique faces: 5
5. | [INFO] faces for face ID: -1
6. | [INFO] faces for face ID: 0
7. | [INFO] faces for face ID: 1
8. | [INFO] faces for face ID: 2
9. | [INFO] faces for face ID: 3
10. | [INFO] faces for face ID: 4
```

Five face cluster classes are identified. The face ID of -1 contains any outliers found. You’ll be presented with the cluster montage on your screen. To generate the next face cluster montage just press a key (with the window in focus so that OpenCV’s highgui module can capture your

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.



(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_0.jpg)

Figure 3: Face clustering with Python grouped similar faces for the World Cup player Neymar Jr.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

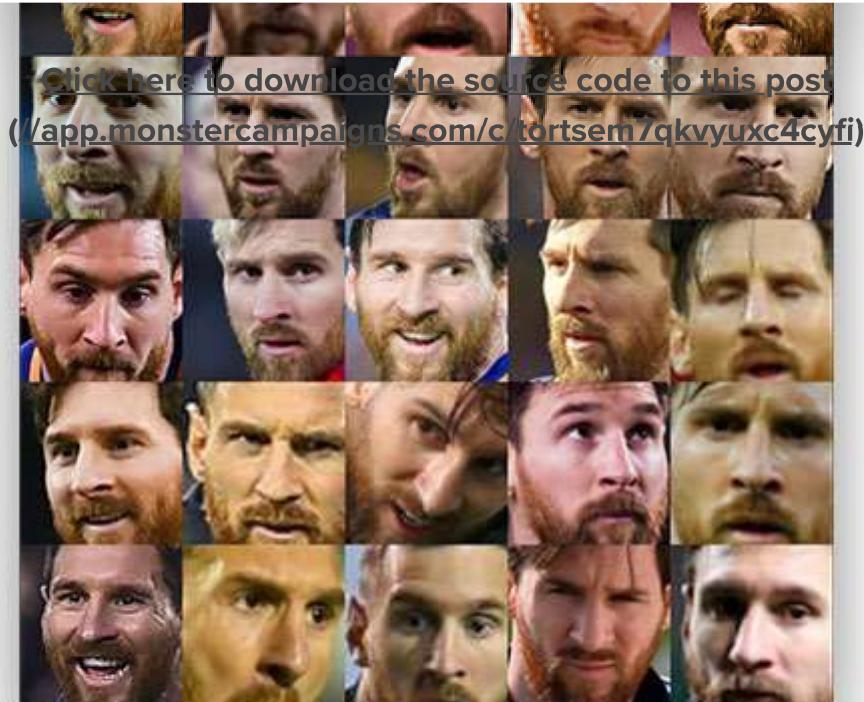
Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!



[\(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_1.jpg\)](https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_1.jpg)

Figure 4: Images of Lionel Messi's face have been grouped together for being similar after running our face clustering with Python script.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.



(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_2.jpg)

Figure 5: Face clustering via Python and the face_recognition library identifies a cluster of 2018 World Cup player, Mohamed Salah.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.



(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_3.jpg)

Figure 6: Our Python face clustering script allows us to find similar face pictures and identify outliers. In this case, we found similar pictures of 2018 World Cup player Luis Suarez.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.



(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_4.jpg)

Figure 7: Cristiano Ronaldo is a 2018 World Cup soccer player. All 25 pictures of Cristiano are grouped together by our Python face clustering script.

And finally, the unknown faces are presented (it is actually displayed first, but I'm providing

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!



[\(https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_neg_1.jpg\)](https://pyimagesearch.com/wp-content/uploads/2018/07/face_clustering_id_neg_1.jpg)

Figure 8: This picture of Lionel Messi didn't get clustered together and is presented as an "Unknown face" as it does not belong to any other cluster. Our Python face clustering algorithm did a reasonably good job clustering images and only mis-clustered this face picture.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Unlike face recognition, which is a **supervised learning** task, face clustering is an **unsupervised learning** task.

[Click here to download the source code to this post](#)

(<http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

With face recognition we have both:

- 1 The faces of people
- 2 And their *names* (i.e., the class labels)

But in face clustering we have only the faces — we do not have their corresponding names as well. Lacking the names/class labels we can leverage only unsupervised learning algorithms, in this case, clustering techniques.

To cluster the actual faces into groups of individuals we choose to use the [DBSCAN algorithm](#) (<https://en.wikipedia.org/wiki/DBSCAN>). Other clustering algorithms could be used as well — Davis King (the creator of dlib) suggests using the [Chinese whispers algorithm](#) ([https://en.wikipedia.org/wiki/Chinese_Wispers_\(clustering_method\)](https://en.wikipedia.org/wiki/Chinese_Wispers_(clustering_method))).

To learn more about face recognition and computer vision + face applications be sure to refer to the first two blog posts in this series:

- [Face recognition with OpenCV, Python, and deep learning](#)
(<https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

```
1 # construct the head model that will be
2 # placed on top of the
3 # base model
4 headModel = baseModel.output
5 headModel = cv2.dnn.blobFromImage(
6     headModel,
7     activation_scale_factor=1,
8     headModel.shape,
9     headModel.mean,
10    headModel.std,
11    swapRB=True)
12 # place the head model on top of the
13 # base model
14 # the activation function
15 model = cv2.dnn.concatenate(baseModel, headModel)
16 outputs = model.forward()
17
18 # loop over the outputs
19 # them so we can visualize
20 # them
```

Download the Source Code and FREE 17-page Resource Guide

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Click here to download the source code to this post
(//app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi)



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:

An interview with Francois Chollet

(<https://www.pyimagesearch.com/2018/07/02/an-interview-with-francois-chollet/>)

Next Article:

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!

**Javier Perez**

July 9, 2018 at [Click here to download the source code to this post](#) (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470470>) (<https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

Hi,

good post, thanks!

Just for fun, in case you didn't know it already: the image that was not clustered with the others doesn't belong to Lionel Messi, is actually a dopplegänger (see here:

<http://www.spiegel.de/sport/fussball/messi-doppelgaenger-iraner-reza-parastesh-sorgt-fuer-chaos-a-1146672.html> (<http://www.spiegel.de/sport/fussball/messi-doppelgaenger-iraner-reza-parastesh-sorgt-fuer-chaos-a-1146672.html>)

So, in fact, the clustering algorithm worked really well! 😊

Regards

**Adrian Rosebrock**

July 9, 2018 at 2:03 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470481>)

Hah! No way! That is so cool, I had no idea it was doggleganger. Thank you for sharing.

**jorge nunez**

July 9, 2018 at 8:54 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470505>)

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Start my email course!



July 9, 2018 at 2:02 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470480>)

Click here to download the source code to this post

(<http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

Very surprising indeed!



SXW

July 9, 2018 at 2:27 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470484>)

Love this post! Thanks!



Adrian Rosebrock

July 9, 2018 at 4:05 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470488>)

Thanks, I'm glad you liked it! 😊

×

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!

recognition using a model trained *only* on that cluster of faces. As to whether or not it improves accuracy or not that's up to you to decide.

[Click here to download the source code to this post](https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)

([//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi](https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi))



rivendil

July 9, 2018 at 8:24 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470502>)

Hey Adrian. Thanks for the great post!

Greetings from Argentina.



Adrian Rosebrock

July 10, 2018 at 8:18 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470579>)

Thank you, I'm glad you enjoyed it! 😊



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!



Adrian Rosebrock [Click here to download the source code to this post](#)

July 10, 2018 at 12:17 pm (<https://www.pyimagesearch.com/comments/2018/07/09/face-clustering-with-python/#comment-470578>)

Wow! I had no idea. Thank you for sharing Mohamad.



Thakur Rohit

July 10, 2018 at 12:26 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470518>)

Hey, I am wondering how can be save the obtained clusters which is shown as a montage file?



Adrian Rosebrock

July 10, 2018 at 8:14 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470575>)

You can use the “cv2.imwrite” function. If you are new to OpenCV, no worries, but I would recommend reading [Practical Python and OpenCV](#) (<https://www.pyimagesearch.com/practical-python-opencv/>) to help you get up to speed.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!



Adrian Rosebrock [Click here to download the source code to this post](#)

July 10, 2018 at 10:10 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470573>)

Thanks Sourabh, I appreciate that. Be sure to see the introduction to this post where I discuss a real-world scenario.



Sourabh Mane

July 11, 2018 at 1:31 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470725>)

My mistake, i jumped directly to code. And one more thing how can we use Chinese whisper algorithm in this code?? What changes i have to make??



Adrian Rosebrock

July 11, 2018 at 5:38 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470790>)

I don't have any examples pre-made that use the Chinese whispers clustering algorithm. You'll want to refer to the [dlib documentation](http://dlib.net/ml.html) (<http://dlib.net/ml.html>) and replace DBSCAN

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Email address

Start my email course!

**Aravind****Click here to download the source code to this post**[July 10, 2018 at 7:57 am](https://app.monstercampaigns.com/c/tortsem7gkvyuxc4cyfi)[\(https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470562\)](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470562)

Thanks, the post really useful and informative.?

**Adrian Rosebrock**[July 10, 2018 at 8:07 am](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470564)

Thanks Aravind, I'm glad you liked the post 😊

**Ubirajara**[July 10, 2018 at 9:01 am](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470623)

Hi, Adrian,

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.



July 10, 2018 at 9:09 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470631>)
[Click here to download the source code to this post](#)

(<http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

Are you referring to Amazon Rekognition? Or something else?



Ubirajara

July 10, 2018 at 9:11 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470633>)

Sorry, I am wrong.

In this case, Amazon Rekognition is not used.



Andy

July 11, 2018 at 2:19 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470779>)

Hi Adrian, Thanks for awesome post! When we have new faces, should we put into the dataset and re-run the clustering? Or is there any way to run the clustering algorithm to see new face belong to existing face, or totally new and we should create new group?



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Click here to download the source code to this post
(//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)



Aakash Nandrajog

July 12, 2018 at 4:25 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470824>)

Hey,

I'm unable to import build_montages

I have installed imutils, upgrade the imutils but still doesn't work

please help me



Adrian Rosebrock

July 12, 2018 at 6:58 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-470877>)

Are you using a Python virtual environment? Perhaps you installed the imutils library globally and not into the Python virtual environment. Also make sure you don't accidentally have a Python packaged name "imutils" in your working directory as that would cause a problem as

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!

My question is, can I use this same code to cluster images of any other types of objects, such as images of flowers or images of cars? What if the dataset is a random dataset with some faces, flowers, cars and other objects? For a set of dissimilar objects, will the algorithm work as is or some retraining will be needed
?

Thanks again for the post.

Milind



Adrian Rosebrock

July 17, 2018 at 8:07 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-471277>)

If you intend on clustering objects of various class you'll want to quantify each of the images in the same manner. Feature extraction, whether by traditional extractors or transfer learning via CNNs would be a good approach here. I would suggest referring to the [PyImageSearch Gurus course](https://www.pyimagesearch.com/pyimagesearch-gurus/) (<https://www.pyimagesearch.com/pyimagesearch-gurus/>) for examples of such applications.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!

I have a [dedicated post on face alignment](#) to download the source code to this post (<https://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/>). Give that a read, it should help with your project.

**Abhi**

[July 19, 2018 at 7:31 pm](#) (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-471480>)

Does this work with pictures with multiple faces in it? Thanks! Awesome post btw!

**Adrian Rosebrock**

[July 20, 2018 at 6:24 am](#) (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-471513>)

Yes, the code will loop over all faces in a set of input images and extract embeddings for each of the faces.



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

No, provided you have already computed the embeddings for a particular dataset you would not have to re-run encode them. You would need to update the logic in the code to handle this use case, but again, there is no need to re-encode faces.

**Bud**

[September 14, 2018 at 11:00 pm](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-478185) (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-478185>)

Hi Adrian, I applied the sklearn's DBSCAN for videos as you mentioned. It worked for small videos, like (1 minute), but it shows a lot of unique labels for longer videos (like 10minutes). For example, if a video of length 10minutes has 5 people in it, after clustering, it shows 14 people.

Is there a better solution for larger dataset?

**Adrian Rosebrock**

[September 17, 2018 at 2:55 pm](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-478701) (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-478701>)

Hi Bud — did you extract 128-d facial embeddings for each face from every frame and then



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.



November 25, 2018 at 2:09 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-488935>)

Click here to download the source code to this post

(//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)

I am also doing something similar. My questions have to do with 1) finding the right frame rate to extract images from to lower compute resources, 2) finding the right frame image resizing for the same reason and 3) extracting more faces per frame (I am getting too few). Perhaps for #3 I need to increase the number_of_times_to_upsample?



Moshe Chen

October 12, 2018 at 5:15 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-482254>)

Adrian, thanks for this great article, love it. I have followed the above and compiled dlib for GPU. I am using an Amazon P3-2Xlarge GPU instance. everything went fine, the code compiles, software installed and running. The only problem is that when I ran encode_faces.py it didn't use the GPU, it used a CPU and completed in about 15min. What am I missing? do i need to add anything to the command line when running it? I have used the following syntax:
python encode_faces.py --dataset dataset --encodings encodings.pickle
How can i make sure that the GPU is used?

Thanks a lot for all your great guidance

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!



Adrian Rosebrock [Click here to download the source code to this post](#)

October 16, 2018 [\(16 comments\)](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-482810)

Awesome, congrats on resolving the issue Moshe!



Victor (<http://victormeunier.com>)

January 9, 2019 at 2:49 pm [\(1 comment\)](https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-495294)

Hi Adrian, thanks for your work. I've been using a few of your tutorial and it's always really clear and great quality.

I'm trying to cluster faces using OPTICS (which is supposedly fixing some of DBSCAN's flaws). I've done the encoding of the faces and fed them to OPTICS. However, the clustering seems abnormally long (it's still running and has been for at least a good hour).

I know a lot of things could be the root cause of my problem but, do you have any idea? How long does it take for you with DBSCAN?

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

**Hina**

January 3, [Click here to download the source code to this post](#) (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-625941>) (<https://app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi>)

Hello Adrian,

Many thanks for your tutorial, it really stands out from others.

I want to try facial clustering recognition using LBP and k-means. Do you have any implementation of k-means? Do you have any advice on how to combine LBP with k-means? What would be different from this tutorial?

Thanks a lot!

**Adrian Rosebrock**

January 16, 2020 at 11:00 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-656665>)

I would refer to the **[PyImageSearch Gurus course](#)**

(<https://www.pyimagesearch.com/pyimagesearch-gurus/>) which covers both LBPs and k-means.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Your machine is running out of memory. Resize your images to smaller dimensions before detecting faces and extracting the face embeddings.

[**Click here to download the source code to this post**](#)

([//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi](http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi))



Sara

[February 28, 2019 at 12:13 pm \(<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-503996>\)](#)

I tried to do this with a different dataset and all the images are classified as noise.



Huang-Yi Li

[March 4, 2019 at 11:33 pm \(<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-504856>\)](#)

If I try to use some features like eyes, nose, mouth..etc to classify several types. For example, I divides 20 kinds of right eye, 20 kinds of left eye, 15 kinds of mouth. Then I can divide people as $20 \times 20 \times 15 = 6000$ clusters. Do you think this feasible? Indeed, I don't know how classify those facial features as several types. Could you give some advises?



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!

[python/#comment-505088](#)

[Click here to download the source code to this post](#)

Great post 😊

[\(/app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi\)](http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)

Let's say I wanted to then fetch the locations for each unique face cluster. How would I go about fetching each location from the data array and assigning it to its own array based on the clusters?

Thanks



Adrian Rosebrock

March 8, 2019 at 5:55 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-505599>)

I'm not sure what you mean by "fetching each location". Do you mean the location of the face in an input image?



Ben

March 12, 2019 at 4:43 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-506352>)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!

That's not really a computer vision program, more so a general programming question. You should look into the "os.makedirs" and "shutil.copy" functions.

[**Click here to download the source code to this post**](#)

([//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi](http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi))



Adeeb (<http://adeeb.com>)

March 23, 2019 at 9:52 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-508788>)

Hi Adrian,

Is there a DBSCAN “metric” value (or another argument) that is a little more sensitive?

When i run this code on my dataset of 13,000 face images, I only get 1 unique, probably because my images are lower quality than the ones provided in this tutorial.

Is there a way to up the sensitivity on DBScan at all? I'm just trying to compare things to dlibs chines whispers algorithm



Akshay Nevrekar

April 27, 2019 at 3:02 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-508788>)



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!

▼ ▼ June 15, 2019 at 11:31 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-521961>)

Click here to download the source code to this post

(<http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)

Hello

first i wanna thank you for very good example

i have problem i can run encode_faces.py in hog algorithm but when i run it in CNN after 40 images i get cuda out of memory error and i don't know how to fix this

my Laptop config is

core-i7 6700HQ

GTX 1060 6GB

RAM 32GB



Adrian Rosebrock

June 19, 2019 at 2:12 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-522344>)

Reduce your image dimensions before encoding them. Either use `cv2.resize` or `imutils.resize`.

×

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!

person? I tried the features matches but it's not quite effective...

Thank you

Click here to download the source code to this post
(//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)



Adrian Rosebrock

June 26, 2019 at 1:57 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-523197>)

Not that a know of but you should look into “person re-identification” algorithms as that may help you.



River

July 9, 2019 at 11:58 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-525067>)

Hi,

Great tutorial. I'm totally new & managed to follow, with cuda too. My problem is, your source dataset works perfectly just as in the guide, but when I use my own I get 0 unique faces. I feel like I'm missing an obvious step? Am I supposed to somehow prepare my dataset beforehand?

Thank you!

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Email address

Start my email course!

anyone 😊)

Click here to download the source code to this post
(//app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi)



Jay

September 30, 2019 at 2:05 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-555533>)

Thanks for sharing to community, i appreciate you.

your model is working because images sizes are small, if images are big in size like 400KB and above then your model doesn't works.

do you have workaround with that if there are larger image size??

Any help is greatly appreciated.



Adrian Rosebrock

October 3, 2019 at 12:34 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-557458>)

The file size of the image on disk has nothing to do with the model. If you're running into



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!

**Adrian Rosebrock**

October 17, 2018 at 10:11 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-564819>)
<https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>

Sorry, I'm not sure what you mean. Could you elaborate?

**Rushikesh Sabde**

October 24, 2019 at 8:07 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-566822>)

Hi,

Can I use montages images as a dataset for facial recognition and how ??

**Paul Zikopoulos**

December 21, 2019 at 12:55 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-606089>)

My gut tells me it could be a result of a newer version of Python or some of the libraries (I made a

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

**Adrian Rosebrock**

December 26, 2018 at 12:49 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-617904>)
<https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>

Insert the following statement:

```
print(type(image))
```

My guess is that the “image” is “None”, in which case the image was not properly read from disk OR you accidentally re-assigned “image” to “None” somewhere in your code. Double-check your image paths.

**Kesavan T**

January 16, 2020 at 12:49 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-656916>)

Hii Adrian, your pyimagesearch community is helping great for my project.

I have gone through both the face recognition(with dlib and face_regonition and the latest one with opencv alone)by your community. In that extracting embeddings from a image ,I'm comfortable with the latest one(Face-recognition with opencv alone) because it's fast in

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a free 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Hii Adrain, your community is doing great, It helps me a lot in my final year project.
[Click here to download the source code to this post](#)

~~(//app.monstercampaigns.com/c/tortsem7akvyuxc4cyfi)~~
Can we make these clusters into separate folder instead of montages?

Changing the model to 'hog' alone is enough? or Should we make any other changes in the code?

Thank you in advance



Jason

February 20, 2020 at 12:35 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-745587>)

Hi Adrian!

Love PyImageSearch. I'm currently receiving a 512-D embedding. When I try to cluster a set of 1900 faces using DBScan, it appears that there are no matches.

Do you have any suggestions as to how one could fix this? Is this because the data is too sparse/high dimensional?

Post Reply

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

[Click here to download the source code to this post](#)



Adrian Rio <https://www.pyimagesearch.com/c/tortsem7qkvyuxc4cyfi>)

March 4, 2020 at 1:37 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-765476>)

You are welcome, Rio 😊



Halim

March 9, 2020 at 5:39 pm (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-765880>)

Hello, I have a question. Now that image has been clustered, how to test new unseen image belong to which cluster?



Mashaqi Lamees

April 13, 2020 at 1:58 am (<https://www.pyimagesearch.com/2018/07/09/face-clustering-with-python/#comment-779445>)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

Email address

Start my email course!

[Click here to download the source code to this post](#)
Before you leave a comment
<https://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>

Hey, Adrian here, author of the PyImageSearch blog. I'd love to hear from you; however, I have made the decision to no longer offer free 1:1 help over blog post comments. I simply do not have the time to moderate and respond to them all.

To that end, myself and my team are doubling down our efforts on supporting our paying customers, writing new books and courses, and authoring high quality Computer Vision, Deep Learning, and OpenCV content for you to learn from.

I'd be happy to help you with your question or project, but [I have to politely ask you to purchase one of my books or courses first. \(https://www.pyimagesearch.com/books-and-courses/\)](#)

Why bother becoming a PyImageSearch customer?

- You'll receive a **great education** through my premium content
- You'll receive **priority support**

X

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Click here to download the source code to this post
(//app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi)

Similar articles

MACHINE LEARNING TUTORIALS

k-NN classifier for image classification

August 8, 2016

(<https://www.pyimagesearch.com/2016/08/08/k-nn-classifier-for-image-classification/>)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!

input).

[Click here to download the source code to this post](#)

March 18, 2019

(<http://app.monstercampaigns.com/c/tortsem7qkvyuxc4cyfi>)
(<https://www.pyimagesearch.com/2019/03/18/im-writing-a-book-on-computer-vision-and-the-raspberry-pi-and-i-need-your-input/>) →

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

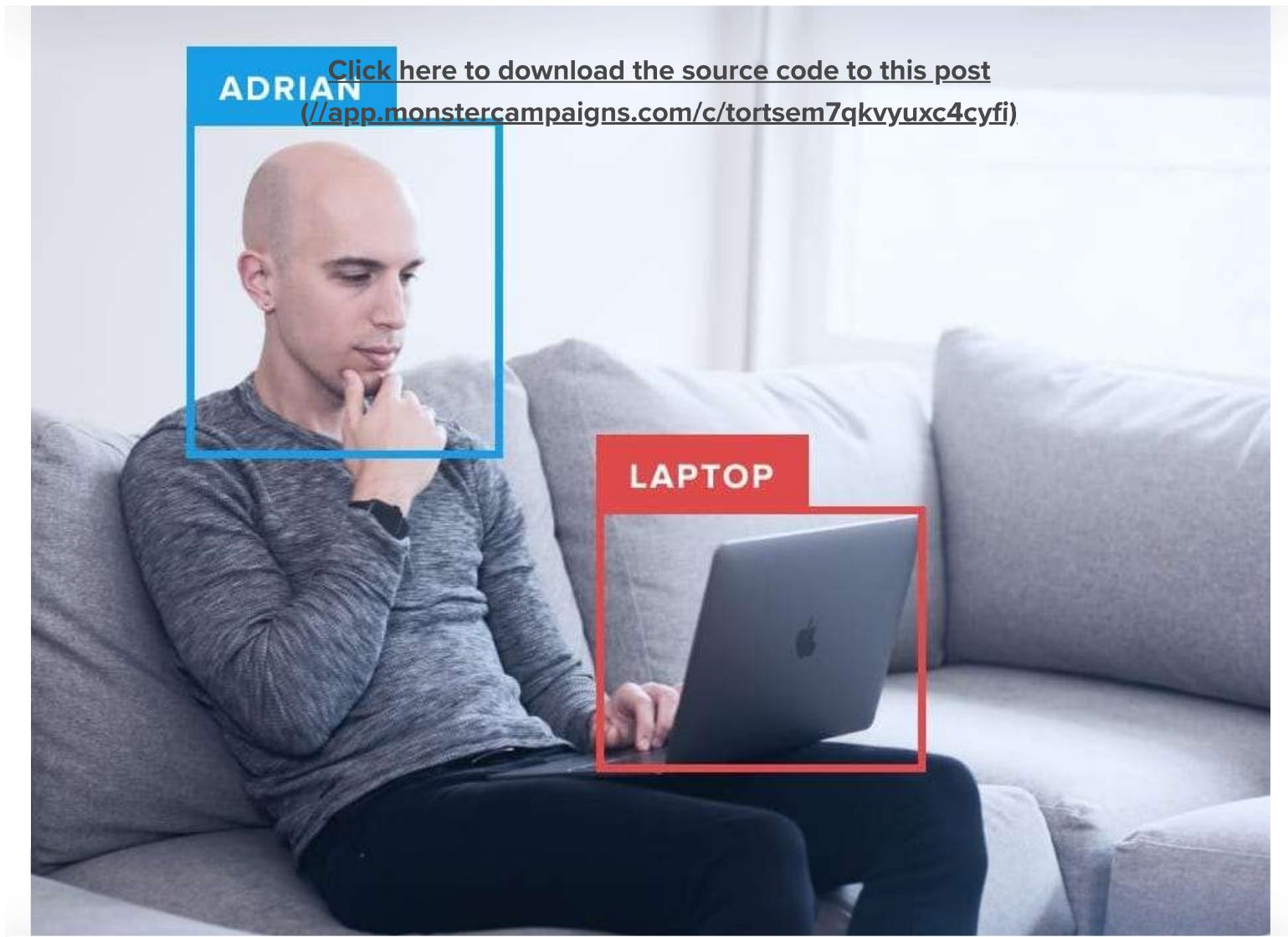
Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!



Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning**.

Sound good? Enter your email below to get started.

Click here to download the source code to this post**Topics**Deep Learning

(<https://www.pyimagesearch.com/category/deep-learning-2/>)

Dlib Library

(<https://www.pyimagesearch.com/category/dlib/>)

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/>)

Machine Learning and Computer Vision

(<https://www.pyimagesearch.com/category/machine-learning-2/>)

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/>)

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to **give you the best possible introduction to computer vision and deep learning.**

Sound good? Enter your email below to get started.

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

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

[Raspberry Pi for Computer Vision](https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/)
[\(https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/\)](https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/)

[Blog](https://www.pyimagesearch.com/topics/) (<https://www.pyimagesearch.com/topics/>)
[Contact](https://www.pyimagesearch.com/contact/) (<https://www.pyimagesearch.com/contact/>)
[\(/app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi\)](https://app.monstercampaigns.com/c/tortsem7qkvuyxc4cyfi)

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

[f](https://www.facebook.com/pyimagesearch) (<https://www.facebook.com/pyimagesearch>) [Twitter](https://twitter.com/PyImageSearch)
[in](https://www.linkedin.com/pub/adrian-rosebrock/2a/873/59b) (<http://www.linkedin.com/pub/adrian-rosebrock/2a/873/59b>) [YouTube](https://www.youtube.com/channel/UCoQK7OVcIVy-nV4m-SMCk_Q/videos) (https://www.youtube.com/channel/UCoQK7OVcIVy-nV4m-SMCk_Q/videos)

© 2020 PyImageSearch (<https://www.pyimagesearch.com>). All Rights Reserved.

Free 17-day email crash course on Computer Vision, OpenCV, and Deep Learning

Interested in Computer Vision, Deep Learning, and OpenCV, but don't know where to start?

Let me help. I've created a *free* 17-day email crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning.

Sound good? Enter your email below to get started.

Email address

Start my email course!