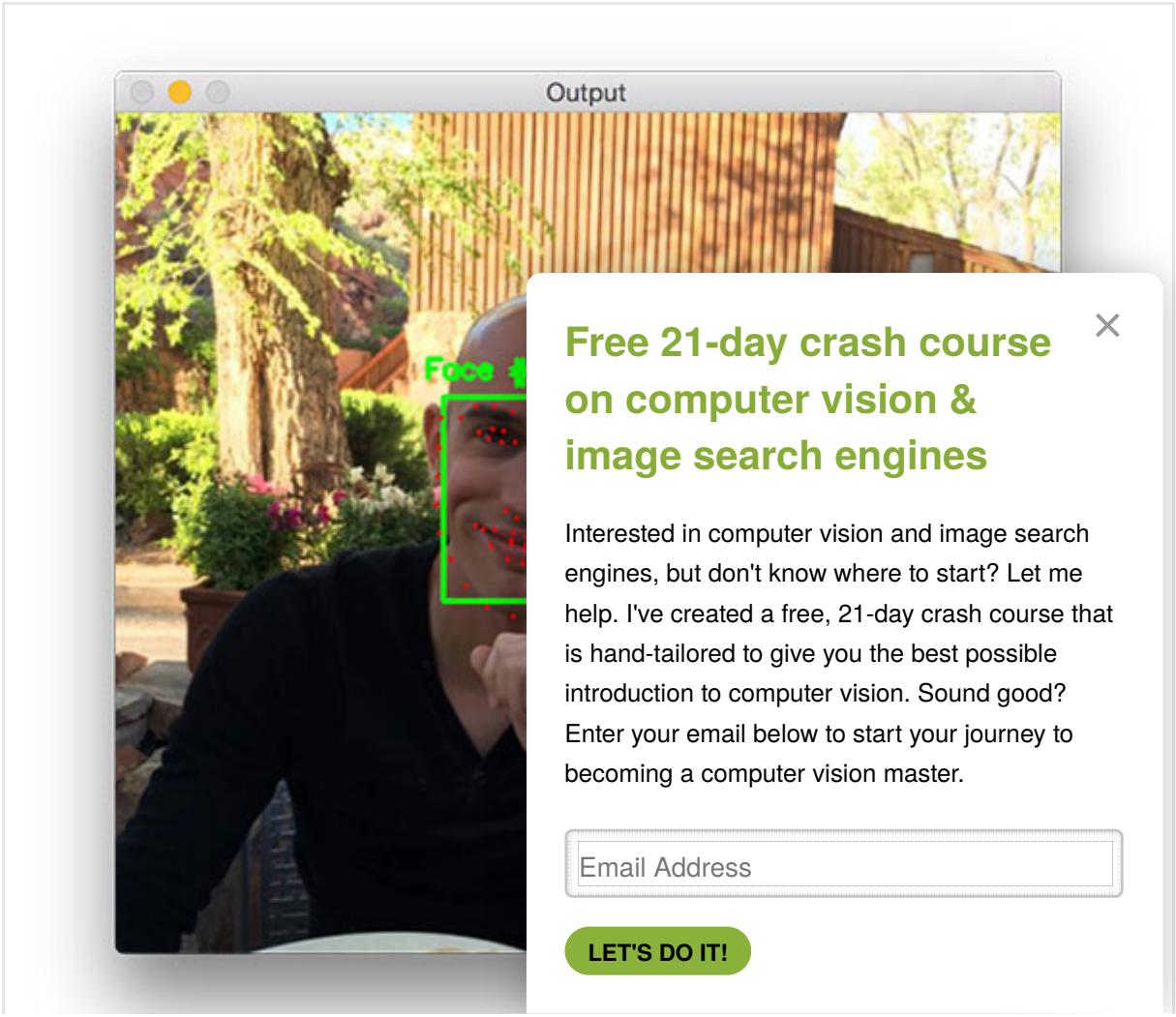
Navigation

pyimagesearch
be awesome at building image search engines

Facial landmarks with dlib, OpenCV, and Python

by **Adrian Rosebrock** on April 3, 2017 in **dlib**, **Facial Landmarks**, **Libraries**, **Tutorials**



Last week we learned how to install and configure dlib on our system with Python bindings.

Today we are going to use dlib and OpenCV to detect ***facial landmarks*** in an image.

Facial landmarks are used to localize and represent salient regions of the face, such as:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more.

In today's blog post we'll be focusing on the **basics of facial landmarks**, including:

1. Exactly *what* facial landmarks are and *how* they work.
2. How to detect and extract facial landmarks from an image using dlib, OpenCV, and Python.

In the next blog post in this series we'll take
to extract *specific* facial regions based on t

To learn more about facial landmarks, *ju*

Looking for the source code
Jump right to the downloads

Facial landmarks with dli

The first part of this blog post will discuss f
vision applications.

From there, I'll demonstrate how to detect a
and Python.

Finally, we'll look at some results of applying facial landmark detection to images.

What are facial landmarks?

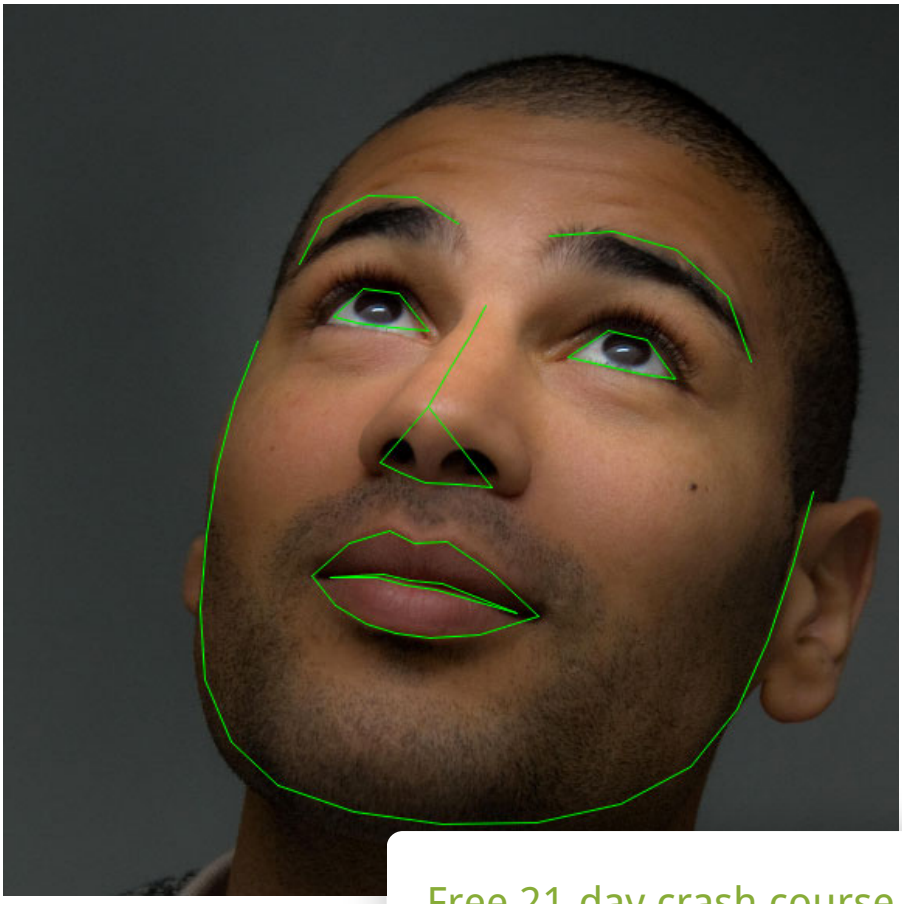


Figure 1: Facial landmarks are use
im

Detecting facial landmarks is a *subset* of th
(and normally an ROI that specifies the obj
localize key points of interest along the sha

In the context of facial landmarks, our goal
using shape prediction methods.

Detecting facial landmarks is therefore a tw

- **Step #1:** Localize the face in the imag
- **Step #2:** Detect the key facial structure

Face detection (Step #1) can be achieved i

We could use OpenCV's built-in Haar cascades.

We might apply a pre-trained HOG + Linear SVM object detector specifically for the task of face detection.

Or we might even use deep learning-based algorithms for face localization.

In either case, the actual algorithm used to detect the face in the image doesn't matter. Instead, what's important is that through some method we obtain the face bounding box (i.e., the *(x, y)*-coordinates of the face in the image).

Given the face region we can then apply **Step #2: detecting key facial structures in the face region.**

There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

The facial landmark detector included in th
Millisecond Face Alignment with an Ensem
Sullivan (2014).

This method starts by using:

1. A training set of labeled facial landmar
   labeled, specifying **specific** *(x, y)*-coor re.
2. *Priors*, of more specifically, the *probab*

Given this training data, an ensemble of re
landmark positions directly from the *pixel in* s
taking place).

The end result is a facial landmark detecto
in *real-time* with **high quality predictions**

For more information and details on this specific technique, be sure to read the paper by Kazemi and Sullivan linked to above, along with the official dlib announcement.

Understanding dlib's facial landmark detector

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of ***68 (x, y)-coordinates*** that map to facial structures on the face.

The indexes of the 68 coordinates can be visualized on the image below:

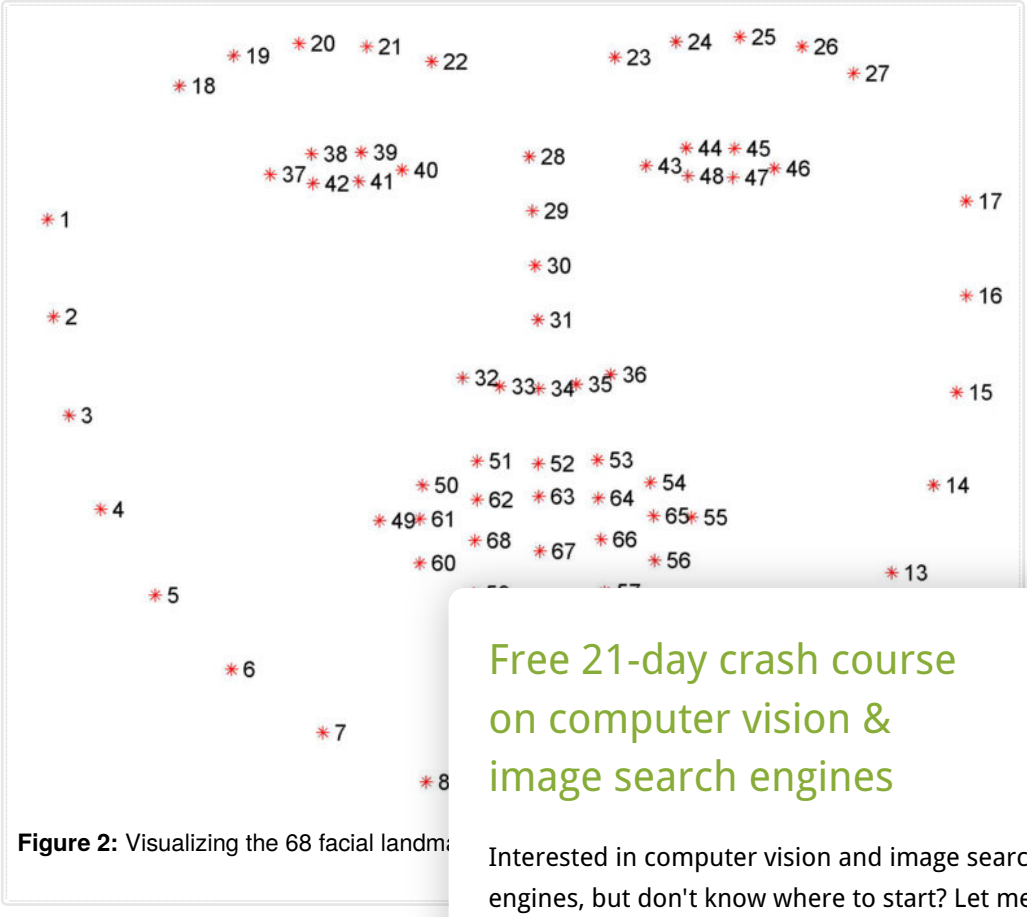


Figure 2: Visualizing the 68 facial landm

These annotations are part of the 68 point k predictor was trained on.

It's important to note that other flavors of fa point model that can be trained on the HEL

Regardless of which dataset is used, the sa shape predictor on the input training data – landmark detectors or custom shape predic

In the remaining of this blog post I'll demonstrate how to detect these facial landmarks in images.

Future blog posts in this series will use these facial landmarks to extract *specific* regions of the face, apply face alignment, and even build a blink detection system.

Detecting facial landmarks with dlib, OpenCV, and Python

In order to prepare for this series of blog posts on facial landmarks, I've added a few convenience functions to my imutils library, specifically inside face_utils.py.

We'll be reviewing two of these functions inside `face_utils.py` now and the remaining ones next week.

The first utility function is `rect_to_bb` , short for "rectangle to bounding box":

Facial landmarks with dlib, OpenCV, and Python — Python

```python
def rect_to_bb(rect):
    # take a bounding predicted by dlib and convert it
    # to the format (x, y, w, h) as we would normally do
    # with OpenCV
    x = rect.left()
    y = rect.top()
    w = rect.right() - x
    h = rect.bottom() - y

    # return a tuple of (x, y, w, h)
    return (x, y, w, h)
```

This function accepts a single argument, `r`
rectangle produced by a dlib detector (i.e.,

The `rect` object includes the *(x, y)*-coord

However, in OpenCV, we normally think of
as a matter of convenience, the `rect_to_b`
it into a 4-tuple of coordinates.

Again, this is simply a matter of conveinenc

Secondly, we have the `shape_to_np` funct

Facial landmarks with dlib, OpenCV, and

```python
def shape_to_np(shape, dtype="int"):
    # initialize the list of (x, y)
    coords = np.zeros((68, 2), dtyp

```

```
    # loop over the 68 facial landmarks and convert them
    # to a 2-tuple of (x, y)-coordinates
    for i in range(0, 68):
        coords[i] = (shape.part(i).x, shape.part(i).y)

    # return the list of (x, y)-coordinates
    return coords
```

The dlib face landmark detector will return a `shape` object containing the 68 *(x, y)*-coordinates of the facial landmark regions.

Using the `shape_to_np` function, we cam convert this object to a NumPy array, allowing it to "play nicer" with our Python code.

Given these two helper functions, we are now ready to detect facial landmarks in images.

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

Facial landmarks with dlib, OpenCV, and Python — Python

```
# import the necessary packages
from imutils import face_utils
import numpy as np
import argparse
import imutils
import dlib
import cv2

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required=True,
    help="path to facial landmark p
ap.add_argument("-i", "--image", re
    help="path to input image")
args = vars(ap.parse_args())
```

Lines 2-7 import our required Python pack

We'll be using the `face_utils` submodule
detailed above.

We'll then import `dlib` . If you don't alread
the instructions in my previous blog post to

Lines 10-15 parse our command line argu

- `--shape-predictor` : This is the path
  can download the detector model here
  post to grab the code + example image
- `--image` : The path to the input image

Now that our imports and command line arguments are taken care of, let's initialize dlib's face detector and facial landmark predictor:

Facial landmarks with dlib, OpenCV, and Python — Python

```
17 # initialize dlib's face detector (HOG-based) and then create
18 # the facial landmark predictor
19 detector = dlib.get_frontal_face_detector()
20 predictor = dlib.shape_predictor(args["shape_predictor"])
```

Line 19 initializes dlib's pre-trained face detector based on a modification to the standard Histogram of Oriented Gradients + Linear SVM method for object detection.

Line 20 then loads the facial landmark predictor using the path to the supplied `--shape-predictor` .

But before we can actually detect facial landmarks, we first need to detect the face in our input image:

Facial landmarks with dlib, OpenCV, and Python — Python

```
22 # load the input image, resize it, and convert it to grayscale
23 image = cv2.imread(args["image"])
24 image = imutils.resize(image, width=500)
25 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
26
27 # detect faces in the grayscale image
28 rects = detector(gray, 1)
```

Line 23 loads our input image from disk via OpenCV, then pre-processes the image by resizing to have a width of 500 pixels and convertin

Line 28 handles detecting the bounding bo

The first parameter to the `detector` is our with color images as well).

The second parameter is the number of im image prior to applying the detector (this it of times on the image).

The benefit of increasing the resolution of t allow us to detect *more* faces in the image the more computaitonally expensive the de

Given the *(x, y)*-coordinates of the faces in detection to each of the face regions:

Facial landmarks with dlib, OpenCV, and Python — Python

```python
# loop over the face detections
for (i, rect) in enumerate(rects):
    # determine the facial landmarks for the face region, then
    # convert the facial landmark (x, y)-coordinates to a NumPy
    # array
    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)

    # convert dlib's rectangle to a OpenCV-style bounding box
    # [i.e., (x, y, w, h)], then draw the face bounding box
    (x, y, w, h) = face_utils.rect_to_bb(rect)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

    # show the face number
    cv2.putText(image, "Face #{}".format(i + 1), (x - 10, y - 10),
        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    # loop over the (x, y)-coordinates for the facial landmarks
    # and draw them on the image
    for (x, y) in shape:
        cv2.circle(image, (x, y), 1, (0, 0, 255), -1)

# show the output image with the face detections + facial landmarks
cv2.imshow("Output", image)
cv2.waitKey(0)
```

We start looping over each of the face detections on **Line 31**.

For each of the face detections, we apply facial landmark detection on **Line 35**, giving us the 68 *(x, y)*-coordinates that map to the specific facial features in the image.

Line 36 then converts the dlib `shape` obje

Lines 40 and 41 draw the bounding box su
while **Lines 44 and 45** draw the index of th

Finally, **Lines 49 and 50** loop over the dete
individually.

Lines 53 and 54 simply display the output

Facial landmark visualizations

Before we test our facial landmark detector on
of `imutils` which includes the `face_util`

Facial landmarks with dlib, OpenCV, and — l

```
$ pip install --upgrade imutils
```

Note: *If you are using Python virtual environments, make sure you upgrade the* `imutils` *inside the virtual environment.*

From there, use the ***"Downloads"*** section of this guide to download the source code, example images, and pre-trained dlib facial landmark detector.

Once you've downloaded the .zip archive, unzip it, change directory to `facial-landmarks` , and execute the following command:

Facial landmarks with dlib, OpenCV, and Python — Shell

```
$ python facial_landmarks.py --shape-predictor shape_predictor_68_face_landmarks.dat
    --image images/example_01.jpg
```

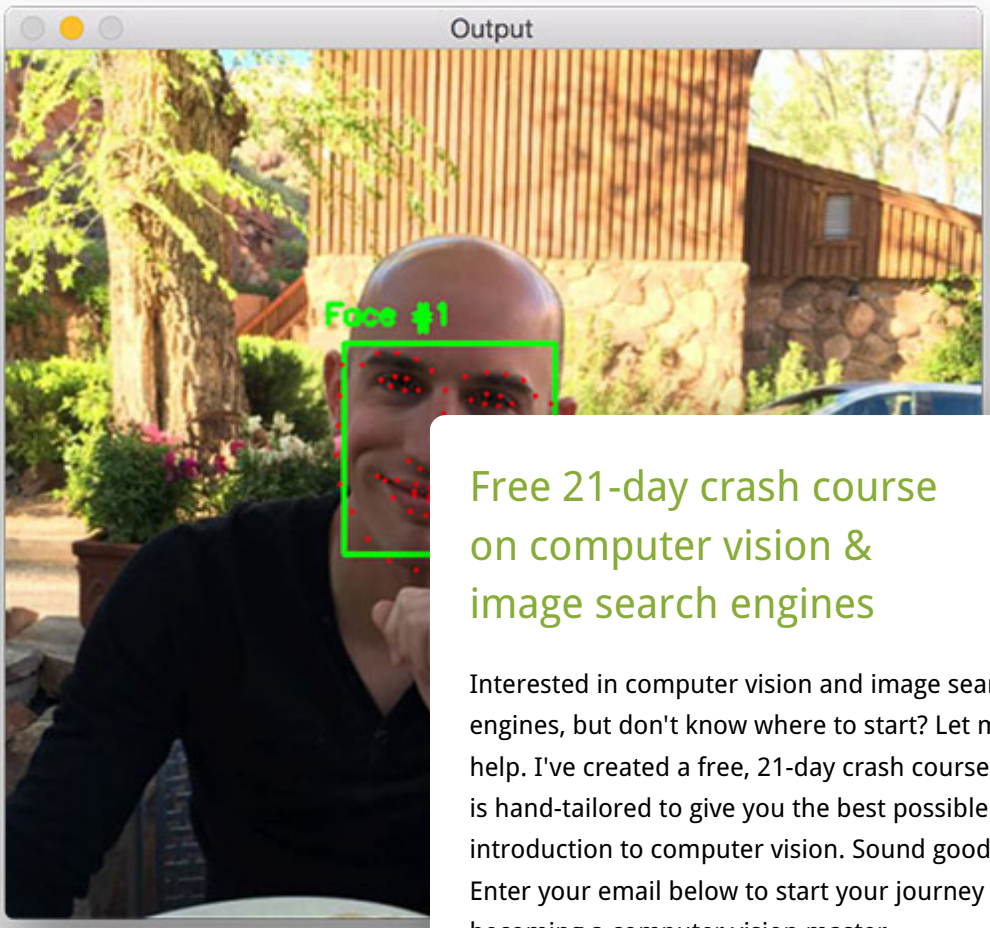


Figure 3: Applying facial landma

Notice how the bounding box of my face is drawn in *green* while each of the individual facial landmarks are drawn in *red*.

The same is true for this second example image:

```
Facial landmarks with dlib, OpenCV, and Python                    Shell
1 $ python facial_landmarks.py --shape-predictor shape_predictor_68_face_landmarks.dat
2     --image images/example_02.jpg
```

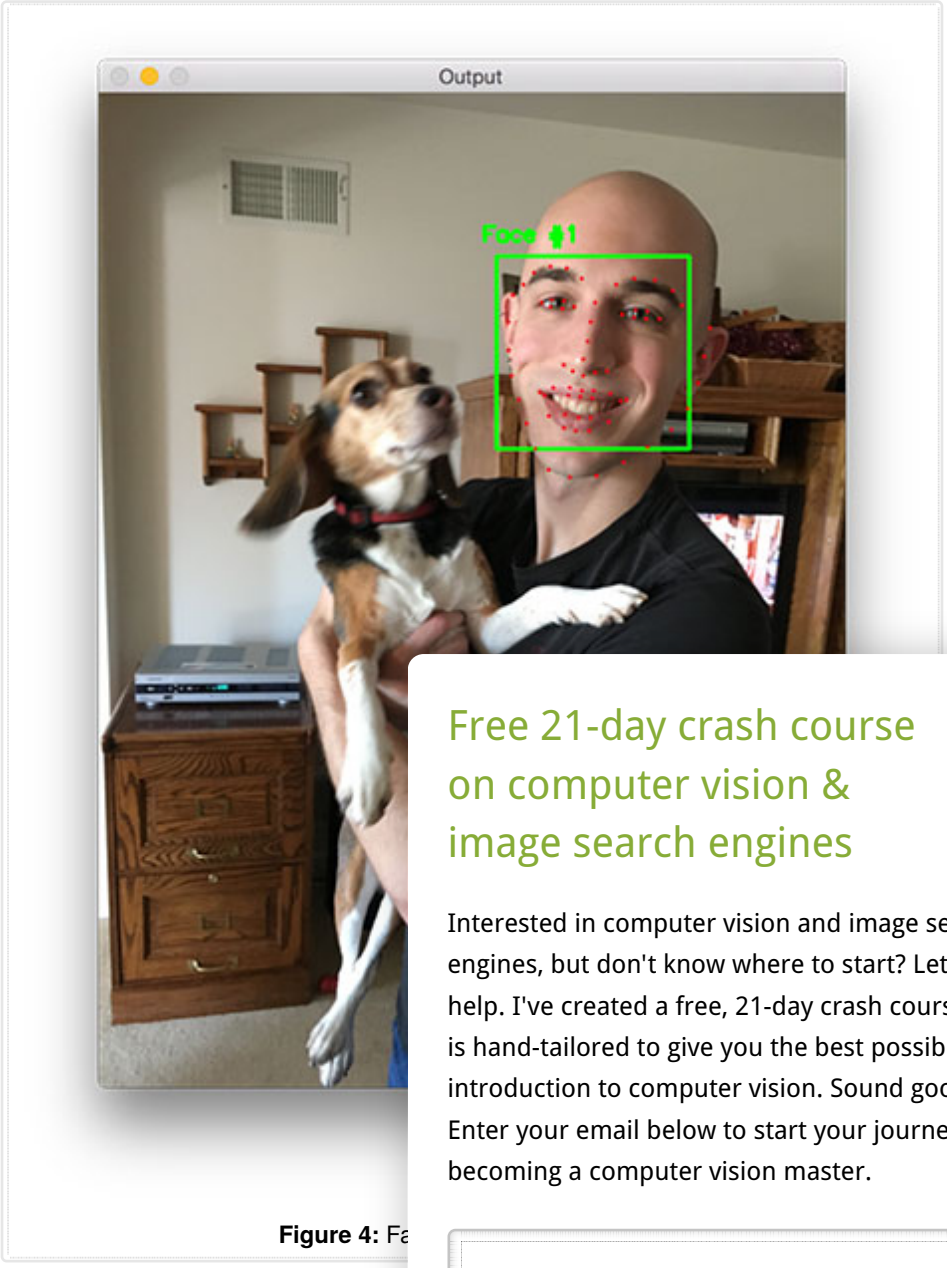


Figure 4: Fa

Here we can clearly see that the red circles
jawline, mouth, nose, eyes, and eyebrows.

Let's take a look at one final example, this time with multiple people in the image:

Facial landmarks with dlib, OpenCV, and Python Shell

```
$ python facial_landmarks.py --shape-predictor shape_predictor_68_face_landmarks.dat
    --image images/example_03.jpg
```

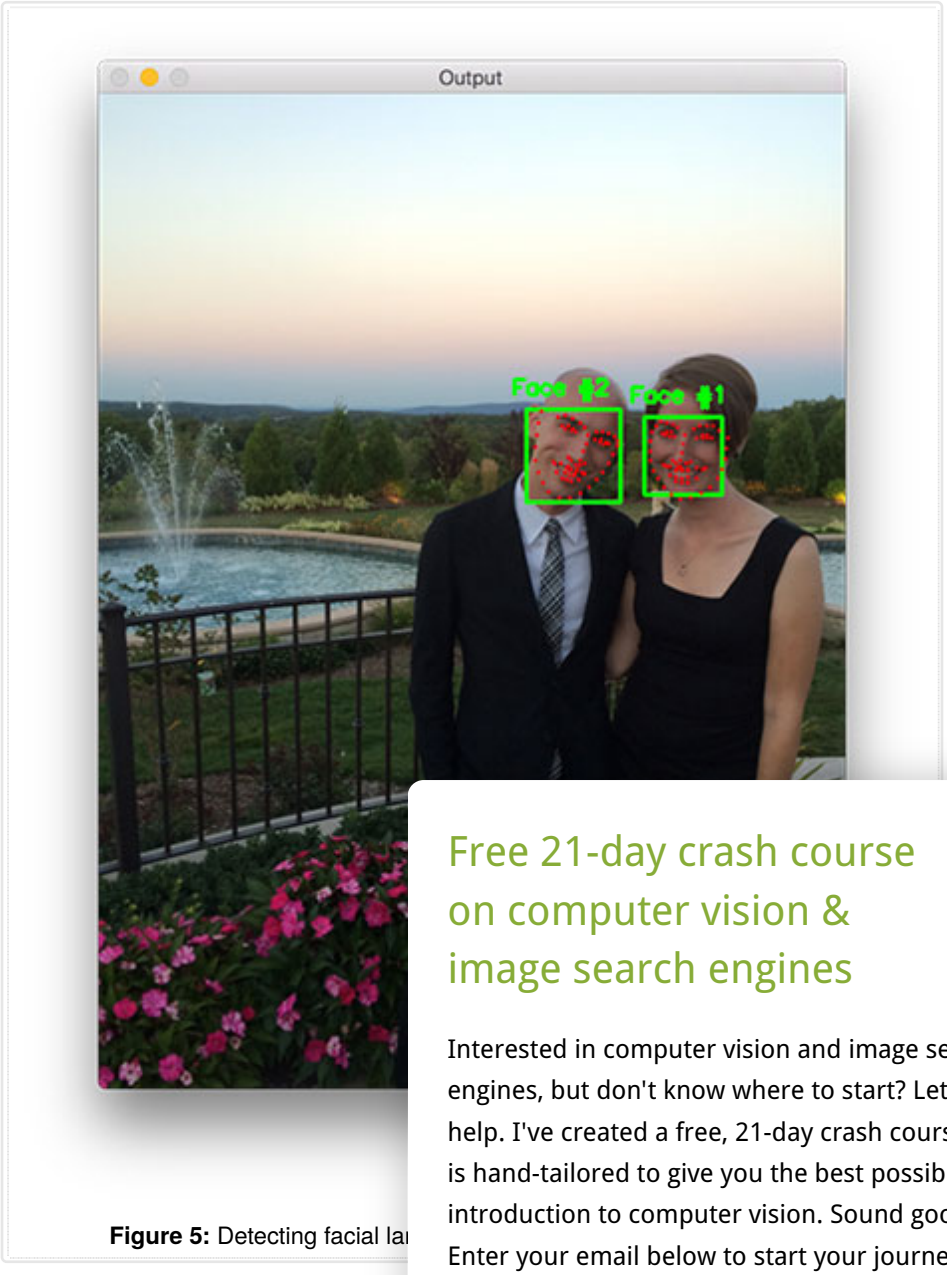


Figure 5: Detecting facial la

For both people in the image (myself and T / but also *annotated* via facial landmarks as

Summary

In today's blog post we learned what facial landmarks are and how to detect them using dlib, OpenCV, and Python.

Detecting facial landmarks in an image is a two step process:

1. First we must localize a face(s) in an image. This can be accomplished using a number of different techniques, but normally involve either Haar cascades or HOG + Linear SVM detectors (but any approach that produces a bounding box around the face will suffice).
2. Apply the shape predictor, specifically a facial landmark detector, to obtain the *(x, y)*-coordinates of the face regions in the face ROI.

Given these facial landmarks we can apply a number of computer vision techniques, including:

- Face part extraction (i.e., nose, eyes, mouth, jawline, etc.)
- Facial alignment
- Head pose estimation
- Face swapping
- Blink detection
- *…and much more!*

In next week's blog post I'll be demonstrating how to access each of the face parts *individually* and extract the eyes, eyebrows, nose, mouth, and jawline features simply by using a bit of NumPy array slicing magic.

To be notified when this next blog post g *in the form below!*

Downloads:

If you would like to dow se enter your email addres the code, I'll also send Computer Vision and In **techniques** that I don't email address and I'll s

Email address:

Your email address

DOWNLOAD THE C

dlib, **face detection**, **face regions**, **facial landmarks**

‹ How to install dlib

Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python ›

126 Responses to *Facial landmarks with dlib, OpenCV, and Python*

Anto April 3, 2017 at 11:24 am #

Superrrrrbbbbbbbbbb blog .searcl
explained.looking forward for face recogn
go ahead soon..!!!!
awesome blog….!!!!!!

Adrian Rosebrock April 3, 20

Thanks Anto!

elliot August 20, 2017 at 3:06

thanks for providing such nice code and helping people.
Can you please guide me how to save extracted landmarks in .mat file so i could use it in matlab.
thanks

Adrian Rosebrock August 21, 2017 at 3:41 pm #

REPLY

You should be able to use the savemat function in SciPy.

Jaechang Shim June 6, 2017 at 3:15 am #

REPLY

It's great!! Working very well, Thanks a lot!!

Danny April 3, 2017 at 3:51 pm #

REPLY

Thank you so much Adrian!!

Mário April 3, 2017 at 8:37 pm #

Very good job Adrian.
All of your explanation are very useful.
This one, in special, is very important for m
Thank you a lot!!!

Adrian Rosebrock April 5, 20

Thank you Mário! 🙂 I wish yo

Abkul Orto April 4, 2017 at 4:47 am #

This is a clear, clean, and EXCEL

Any plan to include this concept and the Deep learning version of training and implementation in your up coming Deep learning book?

Adrian Rosebrock April 5, 2017 at 12:01 pm #

REPLY

Thanks Abkul, I'm glad you enjoyed the tutorial!

I don't plan on covering how to train custom facial landmark detectors via deep learning inside Deep Learning for Computer Vision with Python, but I will consider it for a future tutorial.

Dimitri April 4, 2017 at 6:33 am #

REPLY

This blog is a goldmine. Thank you so much for writing this.

Adrian Rosebrock April 5, 2017 at 12:00 pm #

REPLY

I'm glad you're enjoying the blog Dimitri, I'm happy to share 🙂

Thimira Amaratunga April 4, 201

Hi Adrian,

This is a great article. Cant wait for next we
individually.

After some experimenting (and with your h
features. Here is the method I used,
http://www.codesofinterest.com/2017/04/ex

Undoubtedly, this method I used could use
So, waiting for your article 🙂

Adrian Rosebrock April 5, 20

REPLY

Nice job Thimira. The method I will demonstrate in next week's blog post is similar, but uses the face_utils sub-module of imutils for convenience. I'll also be demonstrating how to draw semi-transparent overlays for each region of the face.

Thimira Amaratunga April 5, 2017 at 12:53 pm #

REPLY

Awesome! can't wait for next weeks post 🙂

Neeraj Kumar April 4, 2017 at 2:58 pm #

REPLY

Hey Adrian,

I have already configured dlib with your previous week blog and now when i am trying to run "python facial_landmarks.py –shape-predictor shape_predictor_68_face_landmarks.dat \ –image images/example_01.jpg" command my ubuntu terminal is showing error "python: can't open file 'facial_landmarks.py' : [Errno 2] no such file or directory ".

PS : I have already downloaded your code and files and i am running my code inside that 'facial-landmarks' folder. All the files are present as well.

Neeraj Kumar April 4, 2017 at 3:13 p

Dear Adrian,

Fixed the previous issue by providing the fu

Thanks and Regards
Neeraj Kumar

Rehan Shaikh September 4, 20

How did you managed to rem
Where do we have to specify the path

Neeraj Kumar April 4, 2017 at 3:25 p

REPLY

Dear Adrian,

I tried working for side faces but its not working, can you please guide what can be the possibilities for side face landmark detection and yes i was also trying working on your example_02.jpg there imutils.resize() method was giving some error.
Attribute error : 'NoneType' object has no attribute 'shape'.

Thanks and Regards
Neeraj Kumar

Adrian Rosebrock April 5, 2017 at 11:56 am #

REPLY

If you're getting a "NoneType" error it's because you supplied an invalid path to the input image. You can read more about NoneType errors in this blog post.

Neeraj Kumar April 7, 2017 at 6:02 am #

REPLY

Fixed Buddy. Thanks a ton.
can you please help me out with – how can i detect landmarks in video and compare with existing dataset of images.

Adrian Rosebrock

I will be discussing lar
blog post.

Manh Nguyen April 5, 2017 at 2:01 a

I hope next post you can use infra

Adrian Rosebrock April 5, 2017 at 11:49 am #

REPLY

I don't have any plans right now to cover IR cameras, but I'll add it to my potential list of topics to cover.

Sachin April 5, 2017 at 3:57 am #

REPLY

Nice article Adrian! Btw shouldn't the shape points in Figure 2 be 0 based?

Adrian Rosebrock April 5, 2017 at 11:49 am #

REPLY

Figure 2 was provided by the creators of the iBUG dataset. They likely used MATLAB which is 1-indexed rather than 0-indexed.

Oli April 6, 2017 at 4:06 am #

REPLY

I also came across this. I have created an image and printed the index numbers as they are with dlib and python here: http://cvdrone.de/dlib-facial-landmark-detection.html

Parag Jain April 5, 2017 at 10:47 am #

Isn't Independent Component Ana
that approach different from this? Advantag

Mansoor April 5, 2017 at 11:30 am #

Adrian, i'm a huge fan! i don't kno
I don't know but i am having trouble runnin
contain face_utils. I think it is not upgradin

Adrian Rosebrock April 5, 2017 at 11:46 am #

REPLY

Make sure you run:

```
$ pip install --upgrade imutils
```

To make sure you have the latest version of imutils installed. You can check which version is installed via pip freeze

addouch April 6, 2017 at 3:23 pm #

REPLY

amazing adrian

I hope next time to show us how to recognize emotions on image

Adrian Rosebrock April 8, 2017 at 12:54 pm #

REPLY

I won't be using facial landmarks to recognize emotions, but I do cover how to recognize emotions via deep learning inside Deep Learning for Computer Vision with Python.

tony April 6, 2017 at 3:53 pm #

Hi Adrian ,thanks for this great po

how dlib eye landmarks can be used to det

Adrian Rosebrock April 8, 20

Hi Tony — I'll be covering how

dlib in the next two weeks. Stay tuned!

bumbu April 10, 2017 at 8:59 am #

REPLY

May we have a tutorial about appl

classify some dataset, thanks sir, you are super!!!

Adrian Rosebrock April 12, 2017 at 1:17 pm #

REPLY

I demonstrate how to use Keras for deep learning applied to MNIST in this post. I'll also be discussing deep learning applied to custom datasets in lots of detail in my upcoming book, Deep Learning for Computer Vision with Python.

Rijul Paul April 24, 2017 at 4:38 am #

REPLY

Hey Adrian,thanks for this blog post.IS there a way so that we can create our own custom shape predictor?

Adrian Rosebrock April 24, 2017 at 9:32 am #

REPLY

Yes, but you will have to use the dlib library + custom C++ code to train the custom shape predictor.

Benu April 27, 2017 at 6:14 am #

I've tried my best to play with the p
result is never as close as shape_predictor
custom data of face similar to that of ibug.

wiem April 28, 2017 at 5:37 am #

Hi Adrean,
I'm following your post about Facial landma
amazing. thank you a lot for such helpful a
landmarks in a file ! So that I'm ask you ho
Thank you

Adrian Rosebrock April 28, 2017 at 9:13 am #

REPLY

Sure, that's absolutely possible. I would use pickle for this:

```python
f = open("landmarks.pickle", "wb")
f = write(pickle.dumps(shape))
f.close()
```

This will save the NumPy array shape which contains the *(x, y)*-coordinates for each facial landmark to disk.

Yang April 29, 2017 at 6:53 am #

REPLY

Hello, Adrian ,the blog is very useful, thanks for this great blog.

Adrian Rosebrock May 1, 2017 at 1:39 pm #

REPLY

Thank you Yang!

Ameer May 7, 2017 at 4:59 pm #

Hello Adrian
I was wondering if you did any tut. on face
thanks

Adrian Rosebrock May 8, 20

Hi Ameer — I'll be doing a blo

pravallika May 16, 2017 at 6:03 am #

hey adrian,
i am trying to achieve real-time face – reco

code with the .compute_face_descriptor(frame , shape) it gives an error that the arguments are not based on c++.please give me a solution sir

wiem May 16, 2017 at 6:42 am #

REPLY

Hi Adrian ,
Thanks a lot for your explanation. It is very useful. However I'm newer in python and I'm trying to save those face landmarks in matrix so I can manipulate them instead of the original image. Would you give me some suggestion. How can I do such thing ?
Thank you Adrian

Adrian Rosebrock May 17, 2017 at 9:59 am #

REPLY

Once you have the shape representing the facial landmarks you can save them to disk via JSON, pickle, HDF5, etc. I would recommend pickle if you are new to Python. I would also suggest working through Practical Python and OpenCV so you can learn the fundamentals of computer vision and OpenCV.

Samuel May 17, 2017 at 3:35 pm #

REPLY

Hello, i see you used dlib face/obj
dlib.rectangle object to bouding values like
draw rectangle, but my problem is i need t
and correct me if i am wrong there i need t
detector require this rectangle, but i cant fi
object, "[(386, 486) (461, 561)]" this sampl
doesnt, i cant event find out that while i wa
more than 4 hours and with no result, is th
impossible?

Adrian Rosebrock May 18, 2

I will look into this and see wh

Vikram Voleti June 30, 2017 at 11:43 am #

REPLY

You can (sort of) implement bb_to_rect on your own. I wanted to do the same, and figured it out after some probing:

For example, if you want to align an image called “frame” whose face you have already detected as a rectangle with (x, y, w, h) values known, you can do it thus:

fa.align(frame, frame, dlib.rectangle(int(x), int(y), int(x + w), int(y + h)))

Here, I used a BGR image as “frame”, I had used

(x, y, w, h) = faceCascade.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=5)

to detect face rectangle, having already defined faceCascade as:

faceCascade = cv2.CascadeClassifier(‘haarcascade_frontalface_default.xml’)

Adrian Rosebrock July 5, 2017 at 6:38 am #

REPLY

Thanks for sharing Vikram!

Sai kumar May 23, 2017 at 2:32 am #

REPLY

Hi,
I am student of you rather than saying a hu
I have a small doubt we are getting coordir
If i want only a specific points in landmarks
like
Nose tip
Left-eye left corner
Right-eye right corner
Corner of lips
These points required to estimate the pose
Please help me this

Adrian Rosebrock May 25, 2

REPLY

If you want only specific facial

which discusses the indexes of each of the facial landmark points (and how to extract them).

jvf May 23, 2017 at 6:28 pm #

REPLY

I decide to run your program again.

python '/home/…/facial_landmarks.py' –shape-predictor '/home /…/shape_predictor_68_face_landmarks.dat' –image '/home/…/30.jpg'

I have the error again. Now python see this dat file but I have the error anyway:

Error
Illegal instruction (core dumped)

What can I do?

Adrian Rosebrock May 25, 2017 at 4:23 am #

REPLY

It's hard to say what the exact error is without having physical access to your machine, but I think the issue is that you may have compiled the dlib library against a different Python version than you are importing it into. Are you using Python virtual environments?

jvf June 1, 2017 at 3:37 pm #

Yes, I use virtual environe
this error.

I have error in this line:

rects = detector(gray,1)

Output:
Illegal instruction (core dumped)

What can I do to solve this problen

Adrian Rosebrock

REPLY

It sounds like you might have compiled dlib against a different version of Python than you are importing it into. Try uninstalling and re-installing dlib, taking care to ensure the same Python version is being used.

Jon May 25, 2017 at 4:43 pm #

REPLY

Fantastic stuff. Thanks for all you've done!
I am doing face detection / recognition on IR images. This means I cannot use the standard features for detection or recognition. I am trying to build my own detctor using the dlib "train_object_detector.py" and it is working really well – mostly.
I have a training set that are faces of one size and the detector is getting faces of similar sizes but completely missing smaller face images.

So my question is how does the system work to detect faces of different sizes. Do you need to have training samples of all the sizes that you expect to be finding? Or does the system take in the training images and resize them?

If you could clarify how this process works and what kind of training set I need and how it works to find faces of different sizes, I would really appreciate it. I have the recognizer working well, I just need to find the faces.

I am using the python dlib, specifically:
http://dlib.net/train_object_detector.py.html

Thank you, Jon Hauris

Adrian Rosebrock May 28, 2

HOG-based object detectors
the objects across the dataset have the
you don't need to worry about the size
scale during training and then image p
images.

Moises Rocha May 30, 2017 at 2:21

Good afternoon, your tutorials are

I am developing a code for facial recognitic

precision.
To get faster I made 37 proportions based on 12 points. But the variation of the face angle also varies the result. I'm limiting the angle 5 degrees to each side horizontally. I will now try to record the proportions for each angle ie a series of frames for the same face. Thus the comparison would be between several faces at the same angle.

If you can give me a light, I thank you.

mux June 1, 2017 at 10:25 am #

REPLY

Hi,
I got this error when trying to run your code:

usage: facial_landmarks.py [-h] -p SHAPE_PREDICTOR -i IMAGE
facial_landmarks.py: error: argument -p/–shape-predictor is required
An exception has occurred, use %tb to see the full traceback.

Can you help me please! thank you

Adrian Rosebrock June 4, 2017 at 6:23 am #

REPLY

Please read up on command line arguments before continuing. You need to supply the command line arguments to

Jack han June 2, 2017 at 5:02 am #

How to open shape_predictor_68_

Adrian Rosebrock June 4, 20

I'm not sure what you mean b
section of this blog post to download th
file.

REPLY

Jack han June 7, 2017 at 5:37 am #

Hi，Adrian. How to train shape_predictor_68_landmarks model?And do you have train demo?It's perfect to have directions to train model.I want to train my model.Thanks!

Adrian Rosebrock June 9, 2017 at 1:51 pm #

REPLY

I don't have any demos of this, but you would need to refer to the dlib documentation on training a custom shape predictor.

Mayank June 6, 2017 at 10:30 pm #

REPLY

Great article, thank you for you efforts.

Is there any way by which i can get more than 68 points of facial landmark. I see some paper mentioning 83 points or more. Is there any library that can help me find some points on forehead ? I am trying to find golden ratio for a face to score it. Thanks!

Adrian Rosebrock June 9, 20

You would need to train your
The pre-trained facial landmark predict
dataset comes to mind as a good start

Caroline Wang September 20,

Hi Mayank,

I'm facing the same issue! Would like t
Have you found any solution?

Thanks!

shravankumar June 12, 2017 at 1:3

REPLY

Hey Chief,

Thank you so much. the post is so clear and works super cool.

Raziye June 20, 2017 at 8:09 pm #

REPLY

Hi,do you have MATLAB or c++ cood for your work ?I try that use your code but I could not and I could not solve its error.
Thanks a lot

Adrian Rosebrock June 22, 2017 at 9:34 am #

REPLY

I only provide Python code here on the PyImageSearch blog. I do not have any MATLAB or C++ code.

SathyaNK June 23, 2017 at 6:39 am #

REPLY

Hi adrian…I'm having problem with the argument constructor, after giving the path to predictor and image when this line " args = vars(ap.parse_args())" is executed in ipython console it is giving this error
"In [46]: args = vars(ap.parse_args())
usage: ipython [-h] -p SHAPE_PREDICTO
ipython: error: argument -p/–shape-predict
An exception has occurred, use %tb to see
"
please help me with this problem

Adrian Rosebrock June 27, 2

If you're using ipython I would
and simply hardcoding the path to the

Pelin GEZER June 25, 2017 at 1:36 p

REPLY

I tried with the photo of man who has beard. It did not work well. How can we solve?

moises rocha June 26, 2017 at 7:13 pm #

REPLY

Hello, how are you?
I've been a big fan of your posts since I started working with python. Even more so when we made the post for Dlib on Raspberry. I was sad because you did not answer me and that you answered all the questions about your post.
I know I asked a very specific question but could answer me by email if that is the case. Even if it is a negative answer.

I do biometrics with haarcascade but I'm trying with landmarks.

I am developing a code for facial recognition but I am having difficulties in the matter of precision.
To get faster I made 37 proportions based on 12 points. But the variation of the face angle also varies. I'm limiting the angle 5 degrees to each side horizontally. I will now try to record the proportions for each angle ie a series of frames for the same face. Thus the comparison would be between several faces at the same angle.

Thank you for your attention.
bye

Adrian Rosebrock June 27, 2

Hi Moises, I do my best to ans
possibly can, but please keep in mind
receives 100's of emails and comment
always respond to them all.

That said, regarding your specific proje
face recognition but want to use facial
obtain better accuracy? Do I understar
tutorial I provided on face alignment.

moises rocha June 29, 20

Thank you response.
About the project I will explain bett

My code makes reasons like this:
Face 1
Comparison 1 = straight (point1 to point2) / straight (point4 to point8) = “1,2”
Comparison 2 = straight (point3 to point4) / straight (point5 to point6) = “0.8”

Face 2
Comparison 1 = straight (point1 to point2) / straight (point4 to point8) = “1,6”
Comparison 2 = straight (point3 to point4) / straight (point5 to point6) = “1,0”
..

So if the face is straight the comparison is accurate. However if the face is crooked it does not work.

Example of a crooked face facing left:
.. …..
| Head straight but face turned to left
——

Straight face:
…. ….
| Head straight and face straight
——-

The turning of the face is not a problem because the comparison is made by the proportion of the lines. If the head is crooked but the face is straight the code works well.

I hope you have explained it better

thank you

Adrian Rosebrock

Keep in mind that the
algorithms assume a frontal fa
tolerate a small degree of rotat
there becomes a point where t
still not sure I understand your
situation you are running into.

tarun July 5, 2017 at 9:57 am #

Hi Adrian,

Thanks for the wonderful tutorial. However if I wish to get a roi composed of eye lid, eye ball together, for example like in eye localization tasks where in the whole eye including continuous portion from eye lids to eye brows to eye is to be cropped, how do I do the same with facial landmarks code above

best regards
Tarun

Adrian Rosebrock July 7, 2017 at 10:11 am #

REPLY

I'm not sure I understand your question, but since facial landmarks can localize both eyes and eyebrows, you can simply compute the bounding box of both those regions and extract the ROI.

Avi July 11, 2017 at 3:07 am #

REPLY

Great tutorial. Thank a lot!
However, I have one confounding problem:
When running the code at:
rects = detector(gray, 1) I get the following

__ TypeError: No registered converter was
Python object of type str __
I investigated the error, upgraded libboost-

What confounds me is; in a separate work
detector and set rects = detector(img , 1) e
It works fine.
Redid this exercise on python console (line
Ran the program and the error turns up.
No spelling mistakes…
Any pointers, anything will help…
Thanks for your time

Adrian Rosebrock July 11, 20

REPLY

Hi Avi — I have to admit that I

sure what the exact issue is. I would suggest posting the issue on the official dlib forums.

wiem July 11, 2017 at 5:10 am #

REPLY

Hello Sir!
Thank you very much. All your explanations in this tutorial are very helpful.
https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
I am currently studying at the national school of engineers of Sousse, coming from Tunisia. I use facial markers to recognize emotionsfrom the image. I try to create a template (model) from all the landmarks I extracted from the images in the CK + dataset. I advocate using this model to qualify the emotion of the face by using it.
So I wonder if you could help me and guide me how to save the facial landmarks in a training model and how could I predict this pattern to detect facial emotion from benchmarks.
Thank you for your help
Sincerly

Adrian Rosebrock July 11, 2017 at 6:21 am #

REPLY

Is there a particular reason why you are using facial landmarks to predict emotions? As I show in Deep Learning for Computer Vision with Python, CNNs can give higher accuracy for emotion recognition and are much easier to use in this context.

khalid July 16, 2017 at 11:45 am #

hi , thanks for this great tutorial, pl
function crop but it doesn't work.
please if you have any idea, help me.
thanks a lot.

Adrian Rosebrock July 18, 20

REPLY

You can drop out the face usi

face = image[startY:endY, startX:endX]

I cover the basics of image processing

crop images), inside my book, Practical Python and OpenCV. I suggest you start there. I hope that helps!

Julien July 17, 2017 at 11:15 pm #

REPLY

Hi Adrian, thanks for a useful website. I just tried your code on a movie in which I want to annotate faces (cf. your real time post, feeding in a video file instead of the webcam input). When faces are detected, it works well, however the bottleneck is definitely face detection. Are there other "out of the box" solutions than the pre-trained HOG + Linear SVM object detector? You mention deep learning based approaches, is it something I could quickly deploy (i.e., are there pre-trained weights somewhere, and a pre-built network architecture, which would do a decent job?). Thank you for any hints!

Adrian Rosebrock July 18, 2017 at 9:50 am #

REPLY

Are you looking to simply increase the speed of the face detection? Use Haar cascades instead. They are less accurate than HOG + Linear SVM detectors, but much faster.

Julien July 20, 2017 at 7:06 p

sorry, I wasn't clear. The i
wondering if there are other metho
HOG+Linear SVM doesn't. I am nc
project.

Adrian Rosebrock

Sure, there are many
Deep learning-based face dete
face detection API. It really dep
go. Ideally, I would suggest ga
faces and environments the fa

Ankit September 9, 2017 at 2:20 pm #

REPLY

Hello Sir,
I am a huge fan of you.
You are doing wonder full work.

This code works very perfectly.

I want to know is how can I detect face if the image if the image is rotated by 90, 180 or 270 degrees?

And what if I can do this in a live video from the camera?

Adrian Rosebrock September 11, 2017 at 9:18 am #

REPLY

The HOG detector is not invariant to rotation so you'll need to rotate your image by 90, 180, and 270 degrees and then apply the detection to each of the rotated images.

I cover how to apply facial landmarks to real-time video in this post.

Junior September 13, 2017 at 3:39 pm #

REPLY

Hi adrian, thanks for this page.
How can I implement face detection with dl

Adrian Rosebrock Septembe

Face *detection*? Or face *recog*
face detection (i.e., detecting the locati

siva charan September 14, 2017 at 8:02 am #

REPLY

Hi Adrian,

Is it possible to recognize the faces in live cam using OpenCV.I need your suggestions.Currently i am working on face recognization.

Adrian Rosebrock September 14, 2017 at 1:15 pm #

REPLY

Yes, absolutely. I cover face recognition inside the PyImageSearch Gurus course. I would suggest starting there.

wiem September 21, 2017 at 10:12 pm #

REPLY

HI Andrian , thanks for this page, I want ask you if the precision of detected landmarks is related to the image size; because in this tutorial you change it into 500 ??
Cordialy
Wiem

Adrian Rosebrock Septembe

No, the precision of detected l
the image because it's very rare we ne
along their largest dimension. Instead,
faster.

Ashish hajagulkar September 24,

how I can do live facial landmark

Adrian Rosebrock September 24, 2017 at 8:42 am #

REPLY

Please see this blog post.

Tonmoy September 29, 2017 at 2:18 pm #

REPLY

Hello Adrian, have you written any blog on how to estimate head pose using facial landmarks? Please let me know. Cant seem to find any elegant solution.

Adrian Rosebrock October 2, 2017 at 10:08 am #

REPLY

Hi Tommy — I have not written a post on head pose estimation, but I will consider this for a future blog post topic.

Arfah S September 29, 2017 at 4:28 pm #

REPLY

This is one of the best articles ive ever come across!

Adrian Rosebrock Septembe

Thanks Arfah!

Darshil October 25, 2017 at 2:03 pm #

Hi Adrian,
Thanks for this page. This is a great tutoria

python facial_landmarks.py –shape-predic
images/example_01.jpg

I'm getting

Illegal instruction (core dumped)

Adrian Rosebrock October 26, 2017 at 11:47 am #

REPLY

Hi Darshil — I'm sorry it isn't running on your system. Can you try running this example program to see if it works?

Darshil October 29, 2017 at 2:15 am #

REPLY

Thank you Adrian for your reply. I tried running the code you sent, it is not giving any error but it is not giving any output either. I am using Ubuntu 14. Can I record and send you a video somewhere ,of what error I'm getting.

Adrian Rosebrock October 30, 2017 at 3:15 pm #

REPLY

Hi Darshil. cv2.imshow() should display the image on your computer screen. Alternatively you could use cv2.imwrite() to write the image to disk.

Darshil October 30, 2017 at 11:29 am #

REPLY

Hi Adrian, thanks for your

The code you sent is running witho
using Ubuntu 14. I know what the
facial_landmarks.py, video_facial_
drowsiness_detection.py and all th
execution reaches to "rects = detec
shows "Illegal instruction (core dun

Adrian Rosebrock

Hi Darshil — Did the
Without being on your system,
an email with details about you
version, and Python version.

Darshil November 3, 2017 at 12:41 pm #

I have dropped you an email. Please help

Eyshika November 3, 2017 at 12:53 pm #

REPLY

Am also facing error in parse.arg(). I havent left any space in middle still it shows ASSERTION ERROR

Adrian Rosebrock November 6, 2017 at 10:46 am #

REPLY

Hi Eyshika — please read up on command line arguments. You DO NOT need to edit any of the code.

Abdul hanan November 7, 2017 at 2:28 am #

REPLY

Hey there. This is one of the simple and well written tutorial about facial landmarks.I have a question regarding landmarks.

Can we find distance between two landma
according to 68-landmarks detection we sh
anyway to do ?

Adrian Rosebrock November

Compute the Euclidean distar
pixel distance. The distance in a meas
a simple calibration.

Jigyasu Bagai November 12, 2017 a

Hi Adrian great work , can you ple
classification or lip extraction is doing , I m

script ????

looking for an early reply

Adrian Rosebrock November 13, 2017 at 2:02 pm #

REPLY

You can evaluate the facial landmarks against a testing set; however, you cannot evaluate the performance without knowing the ground-truth locations of each facial coordinate.

ghazi November 19, 2017 at 3:28 pm #

REPLY

Hi Adrian great work, thank you for you efforts.

I have a mini project about lip reading authentication.

Does you have an idea about extracting letters from an image ?.

cordially

ghazi November 20, 2017 at 10:58 am #

REPLY

Hi Adrian great work.
I have a project about Lip reading authenti
by points of interest.
Well, your example help me very match in
extracting a letter from the picture?

Adrian Rosebrock November

I don't have any experience w
this publication which discusses the to

Nick December 13, 2017 at 8:51 pm #

REPLY

Hi Adrian!

Great work! Thank you for running this blog!

I just tried the code and I am getting an error: facial_landmarks.py: error: the following arguments are required: -i/–image

and then

facial-landmarks>–image images/example_01.jpg
‘–image’ is not recognized as an internal or external command,
operable program or batch file.

Would you be able to help me to resolve it? I didn’t edited the code.

Adrian Rosebrock December 15, 2017 at 8:31 am #

REPLY

Hey Nick, please see my reply to “mux” (June 1, 2017). Your issue here is that you are not properly supplying the command line arguments. Open up a command line and then execute the following command, just like I do in the blog post:

```
Python
1 $ python facial_landmarks.py --shape-predictor shape_predictor_68_face_land
2     --image images/example_01.jpg
```

You’ll want to make sure you are in the same directory as the facial_landmarks.py script.

Take a second to read up on how to us
and it will help out dramatically. I hope

Trackbacks/Pingbacks

Real-time facial landmark detection with Open
[…] We’ve started off by learning how to detec
Face Alignment with OpenCV and Python - Py
[…] our series of blog posts on facial landmarl
process […]

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Search...

Resource Guide (it's totally free).

Click the button below to get my **free 11-page Image Search Engine Resource Guide PDF**. Uncover **exclusive tech**
image search engines of your

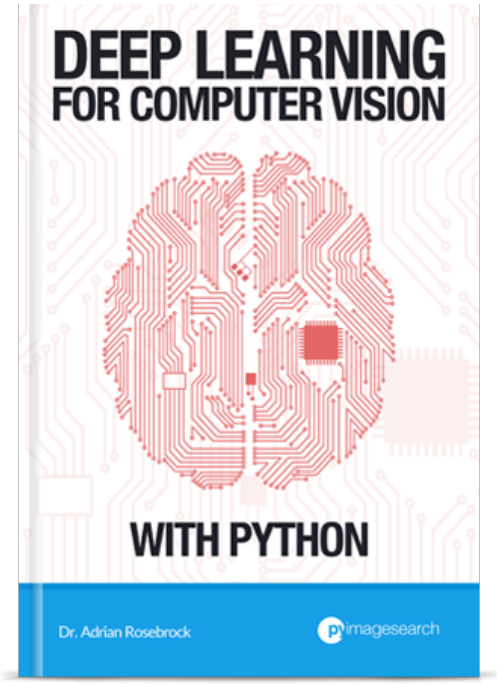
Down

Deep Learning for Computer Vision with P

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

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

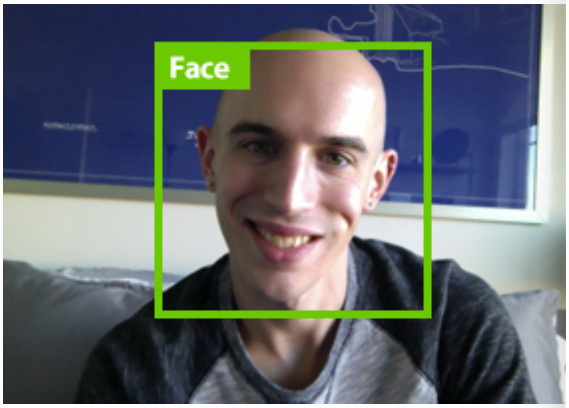
LET'S DO IT!



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 MASTER DEEP LEARNING

You can detect faces in images & video.



Are you interested in **detecting faces in images** *work?* Then let me help! I guarantee that my new 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.

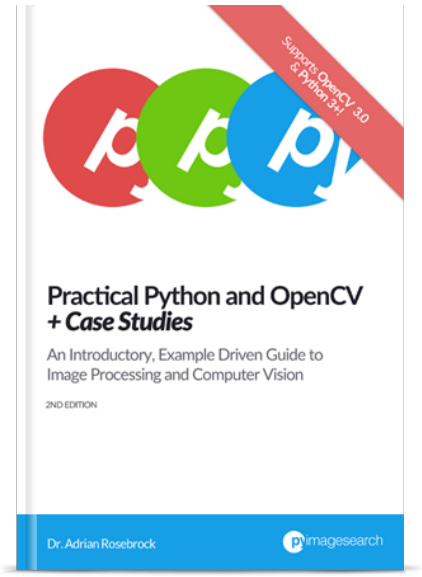
TAKE A TOUR & GET 10 (FREE) LESSONS

Hello! I'm Adrian Rosebrock.

I'm an entrepreneur and Ph.D
ID My Pill and Chic Engine. I'
the way.

ng

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, a

POPULAR

Install OpenCV and Python on your Raspberry P
FEBRUARY 23, 2015

Install guide: Raspberry Pi 3 + Raspbian Jessie
APRIL 18, 2016

Home surveillance and motion detection with th
JUNE 1, 2015

How to install OpenCV 3 on Raspbian Jessie
OCTOBER 26, 2015

Basic motion detection and tracking with Pytho
MAY 25, 2015

Ubuntu 16.04: How to install OpenCV
OCTOBER 24, 2016

Accessing the Raspberry Pi Camera with OpenCV and Python
MARCH 30, 2015

Find me on **Twitter**, **Facebook**, **Google+**, and **LinkedIn**.