

Python Puzzle Solver

This program takes two images as input, one reference and one piece to match. Start by opening the code and picking from one of the pairs of images, for example, p2 (piece 2) & s2 (source 2);



You can do this by changing lines 23 & 25 to:

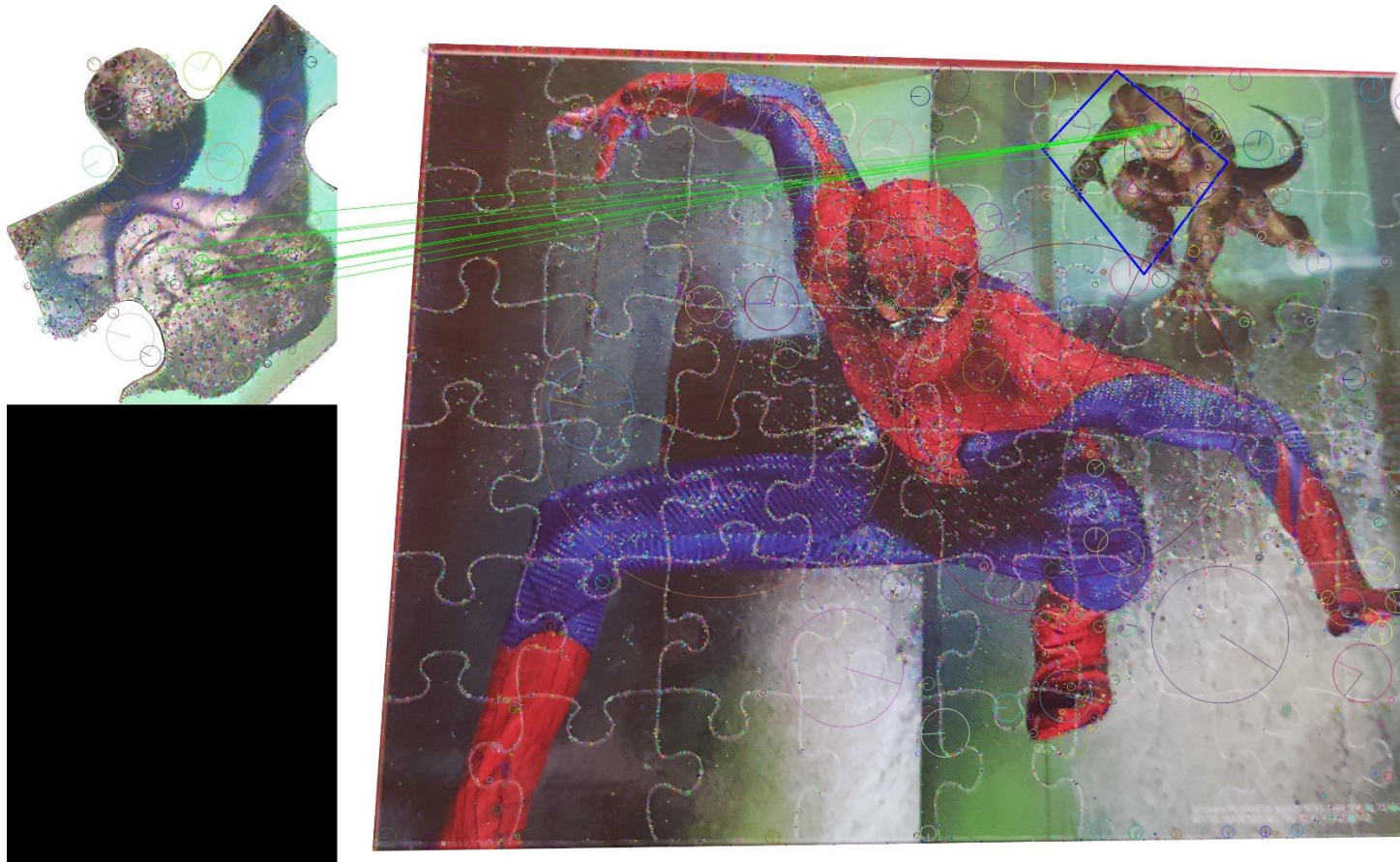
Line 23:

```
>>> canvas = s2
```

Line 25:

```
>>> piece = p2
```

When run, the code will change the “solution.jpg” file in the folder to this;



This image shows the good matches as green lines, and highlights the area in which most of the matches are placed.

```
# creates the SIFT detector instance
sift = cv2.xfeatures2d.SIFT_create()

img1 = cropped_piece.copy() # the piece, also queryImage
img2 = canvas.copy() # the canvas, also trainImage

# running SIFT on the two images
kp1, des1 = sift.detectAndCompute(img1, None)
kp2, des2 = sift.detectAndCompute(img2, None)
```

The usage of SIFT, courtesy of OpenCV, which is a glorious creation.

90% of my time with this program was spent trying to make it more efficient, but the base example I found was already about as good as it gets, so I just automated all of the image selection and opening of images, I also cleaned it up a lot. I also ported it from Python 2.7 to Python 3.6, which took a while and a

lot of research into the changes between Python 2.7 and Python 3.6, but also additional knowledge on OpenCV changes. I will be expanding on this a lot for my Major Project. Might make it an app, who knows.

~~Work distribution? I dunno, that Caleb guy didn't really carry his weight.~~

I'm by myself, so I did all the work.