How bad is bad

Jeren Suzuki

Last Edited April 17, 2013

Contents

1 Running 2 Edge-Detection Filters on an Image with Cut-off Fiducials

During our process of border-checking our image to see if we have any cut off fiducials, we want to make sure that our program is capable of handling fiducials at the edge. If it doesn't error out, at what point will the returned fiducial positions become unreliable? This is our short-term goal for this test.

Figure 4 reveals a basic check to see at what crop size do the convolution filters fail to return a fiducial position.

1.1 edge_truncate

I will point out here that it is important to have the `edge_truncate` keyword set for the convolution filters. Here is an example in Figure 2

Figure 3 shows us that there are significant benefits of using the keyword `edge_truncate` on both filters.

1.2 Another example of filters on an increasingly cropped image

1.3 Changing direction of edge filters

I thought perhaps the direction in which we were cropping and the location of where the filters identified had the fiducial had an effect on our threshold of "how-bad-can-it-get". To test this, I shifted the filters 180° to instead find an edge somewhere else in the fiducual. To be clear, to find an x-position of a fiducial we ran filters in the x direction to isolate the blue-outlined pixels. These positions are consistently the right 2 pixels of the fiducial's 2x2 center. If the edge detected are on the right and we're cropping the right of the image off, there's a chance that we're cropping off that detected edge 1 pixel early, so I modified the code so that it find the left 2 pixels of the center 2x2 so hopefully the fiducial could be more cut off and we'd still get fiducial positions. Results are shown in Figure 5

Compared to Figure 4, we see that in both instances the fiducials were only detected with 5 pixels in the image, regardless of which side of the fiducial we tried to identify.
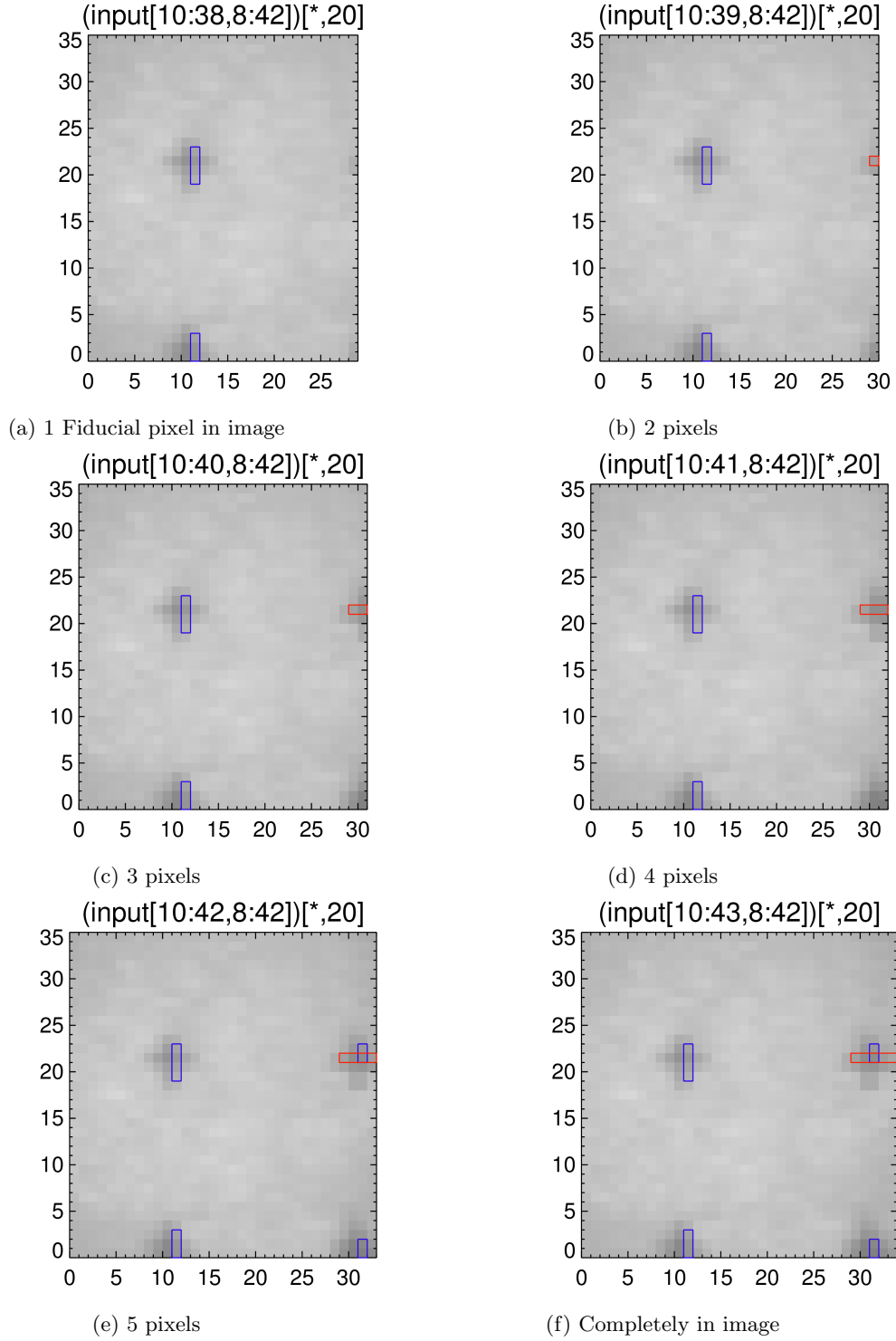
(a) 1 Fiducial pixel in image

(b) 2 pixels

(c) 3 pixels

(d) 4 pixels

(e) 5 pixels

(f) Completely in image

Figure 1: Result of `shift_diff(emboss(image,/edge_truncate),/edge_truncate)`. The top left is the most cropped image and the bottom right is the least cropped image.

(a) Both filters have keyword
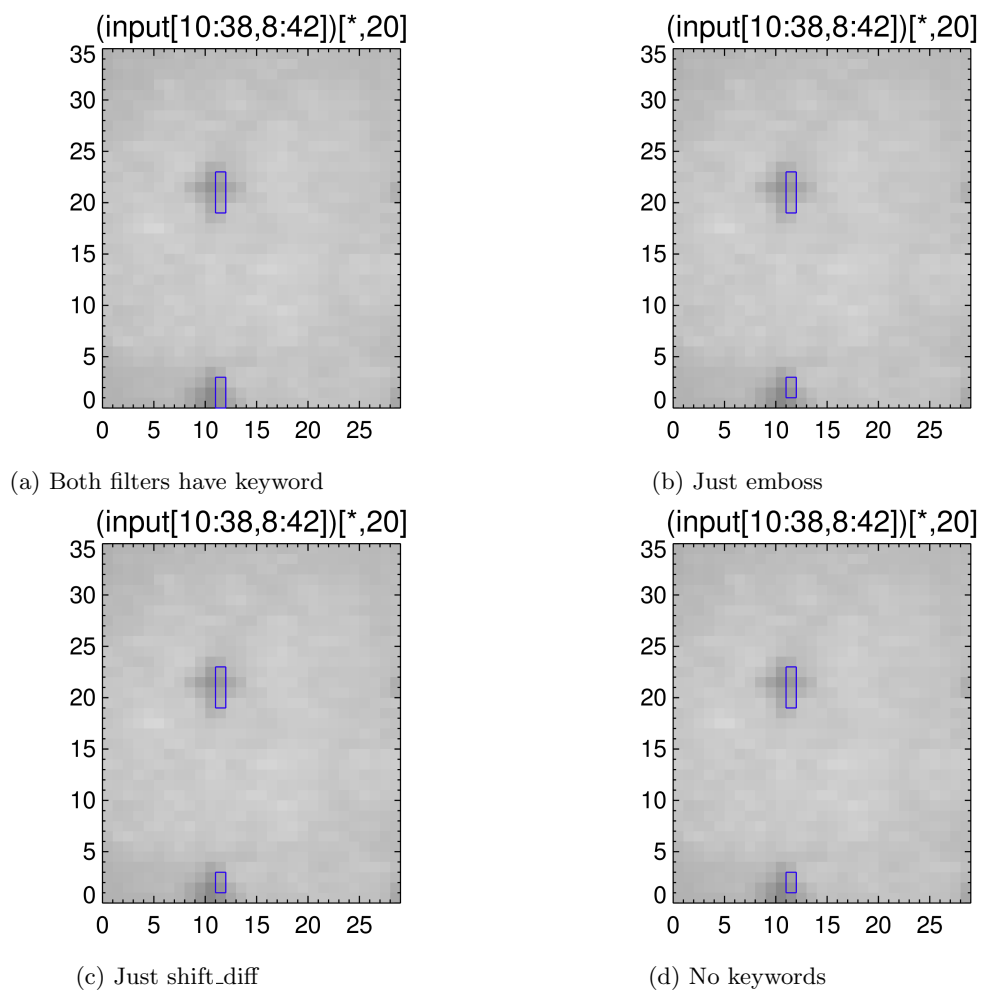
(b) Just emboss

(c) Just shift_diff

(d) No keywords

Figure 2: Only one of the images is different from the other. Can you guess which of the WHO THE WHAT NOW

(a) Both filters have keyword

(b) Just emboss

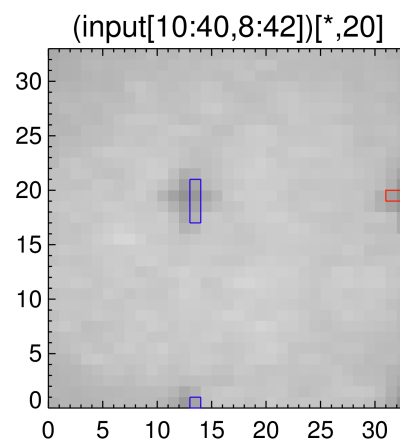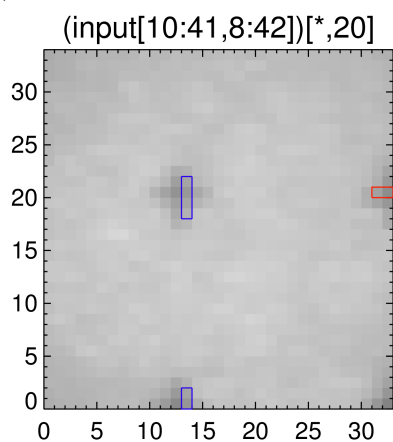(c) Just shift_diff
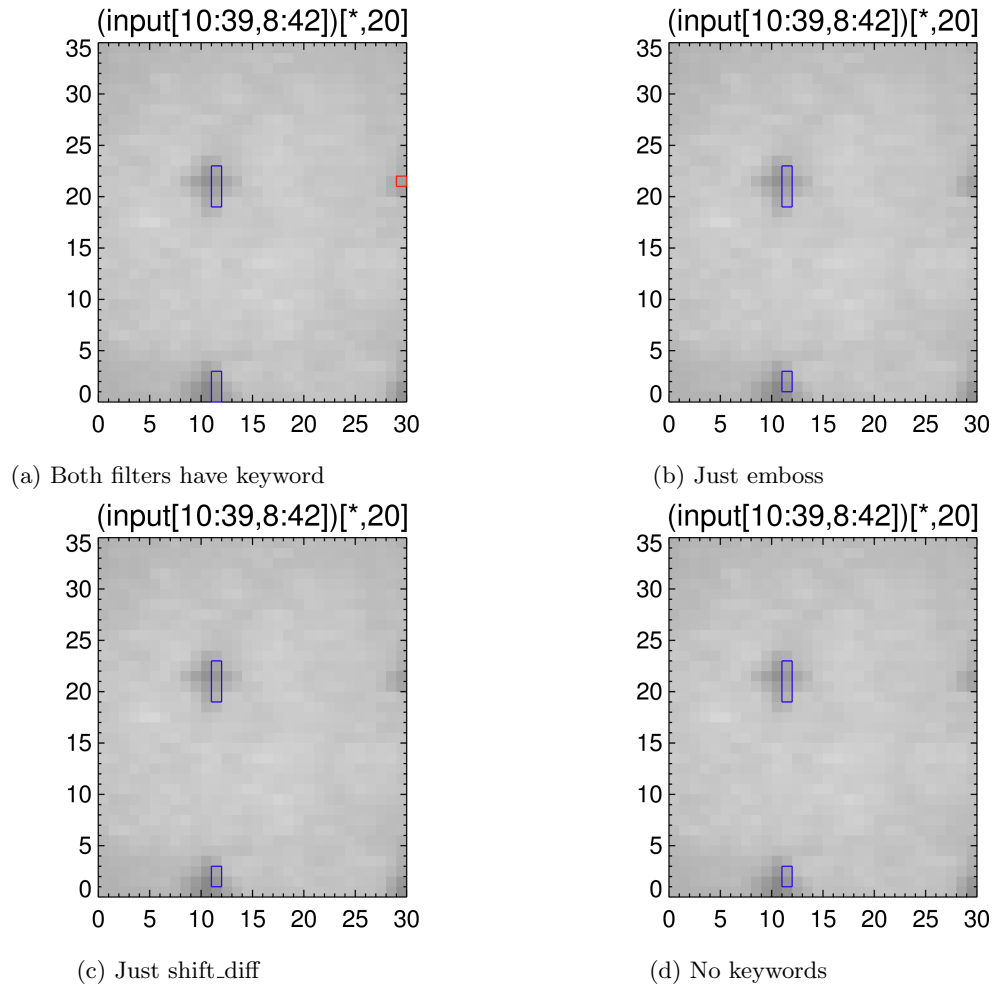
(d) No keywords

Figure 3: Another example, but this time we see that the one with both keyword sets has found a row that we can work with.
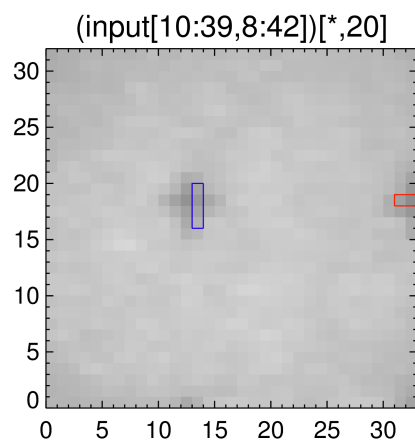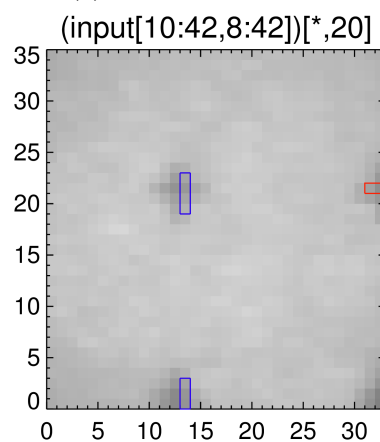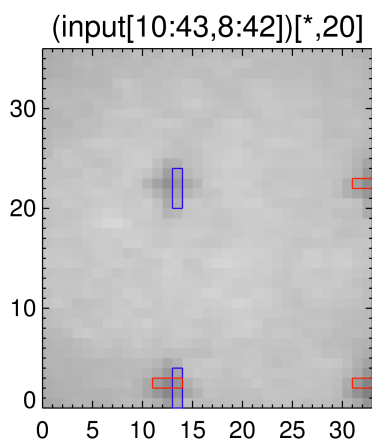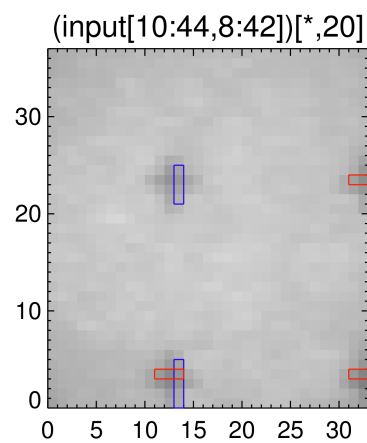
(a) 1 Fiducial pixel in image

(b) 2 pixels

(c) 3 pixels

(d) 4 pixels

(e) 5 pixels

(f) Completely in image

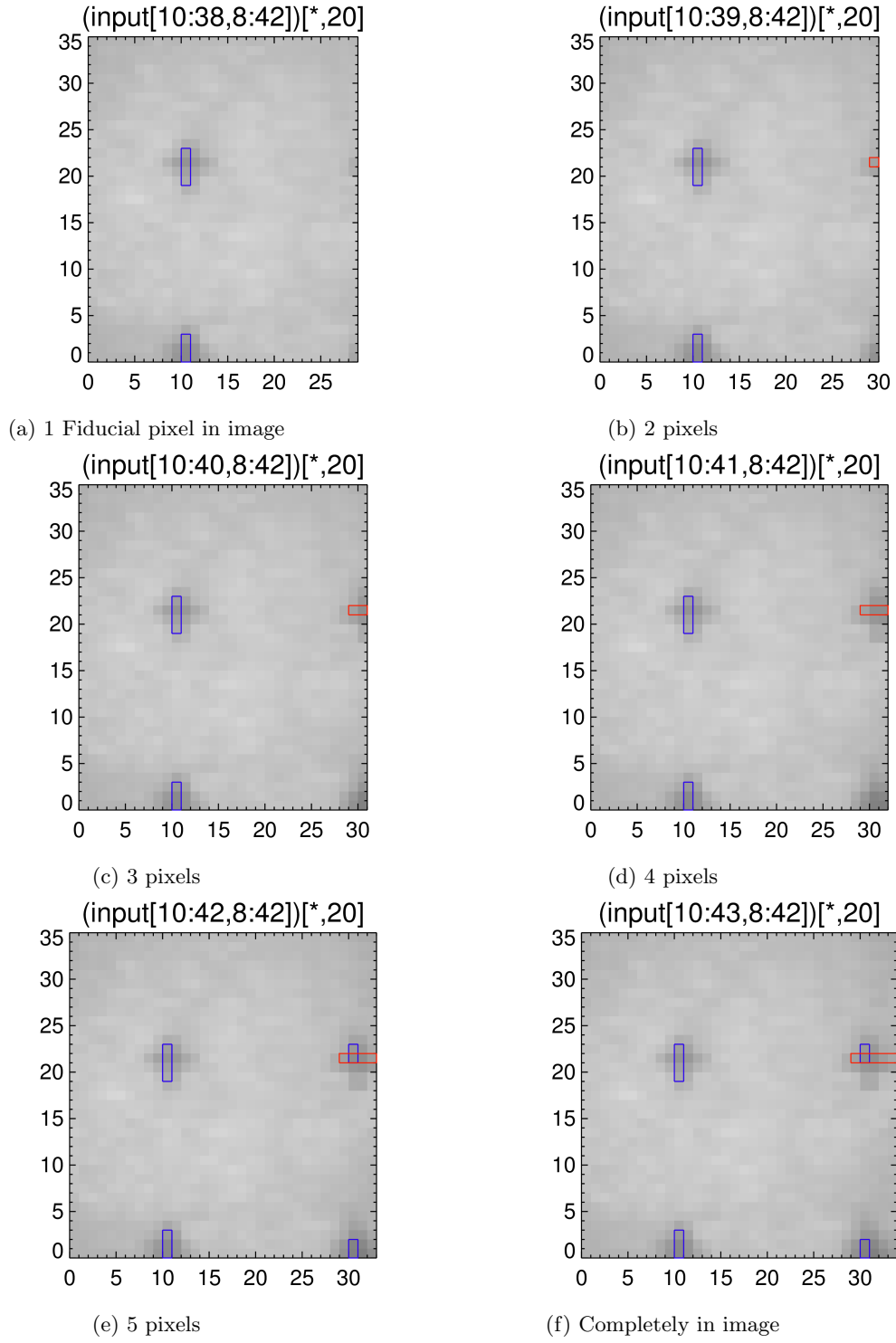Figure 4: This time, we're cropping the bottom less and less

5

(a) 1 Fiducial pixel in image

(b) 2 pixels

(c) 3 pixels

(d) 4 pixels

(e) 5 pixels

(f) Completely in image

Figure 5: Finding the left edge of the fiducual now, instead of the right edge of the fiducial