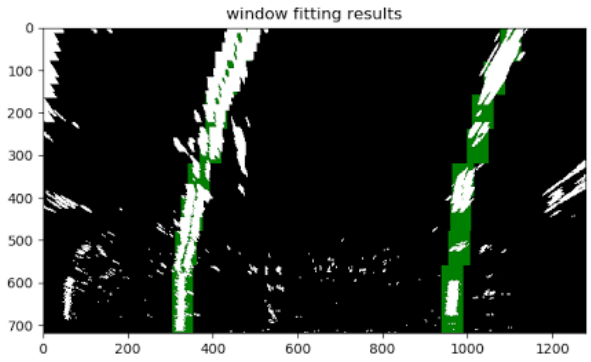
Sliding Window Search

Another way to approach the sliding window method is to apply a convolution, which will maximize the number of "hot" pixels in each window. A convolution is the summation of the product of two separate signals, in our case the window template and the vertical slice of the pixel image.

You slide your window template across the image from left to right and any overlapping values are summed together, creating the convolved signal. The peak of the convolved signal is where there was the highest overlap of pixels and the most likely position for the lane marker.

Now let's try using convolutions to find the best window center positions in a thresholded road image. The code below allows you to experiment with using convolutions for a sliding window search function. Go ahead and give it a try.



Sliding Window Search

```
4  import glob
5  import cv2
6
7  # Read in a thresholded image
8  warped = mpimg.imread('warped_examp
9  # window settings
10 window_width = 50
11 window_height = 80 # Break image in
12 margin = 100 # How much to slide le
13
14 def window_mask(width, height, img_
15     output = np.zeros_like(img_ref)
16     output[int(img_ref.shape[0]-(le
17     return output
18
19 def find_window_centroids(image, wi
20
21     window_centroids = [] # Store t
22     window = np.ones(window_width)
23
24     # First find the two starting p
25     # and then np.convolve the vert
26
```

RESET QUIZ

TEST RUN

SUBMIT ANSWER

NEXT