

Understanding of the problem statement:

The goal was to optimize/improve the code in terms of speed and readability.

What we have done:

We analysed the code and find out the function `form_p` was consuming a lot of time to compute as compared to all of the other functions as shown in the figure and hence it was a bottleneck in the codebase. So we implemented that function in c++ since c++ is a low level language.

What we have achieved:

With our implementation we were able to achieve almost 40 times speed up for `form_p` as you can see in the below figure.

We timed both the original implementation and our implementation of that function.

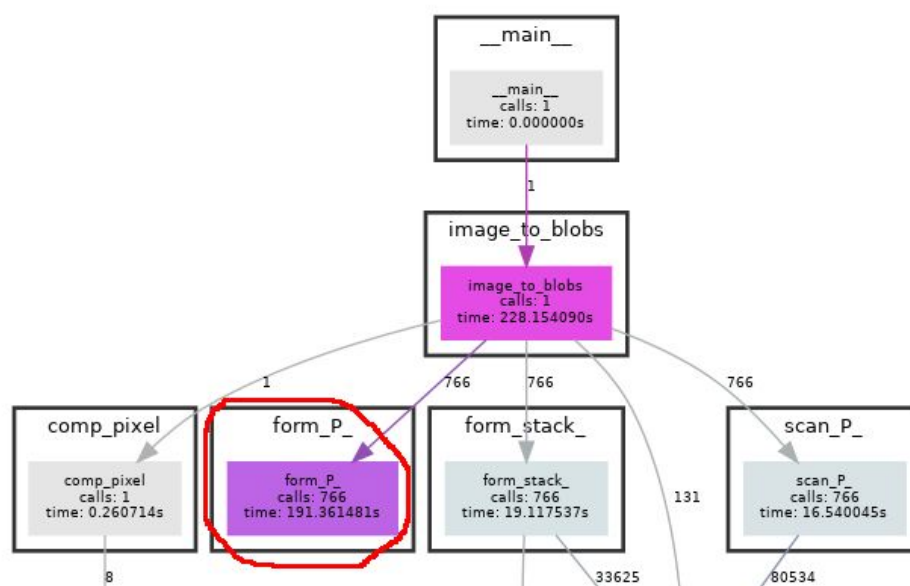
For 7 runs we got almost 40 times speed up in our implementation.

```
In [2]: from frame_blob import image_to_blobs
In [3]: from scipy.misc import face
In [4]: import cv2
In [5]: racoon = face()
In [6]: image = cv2.cvtColor(racoon, cv2.COLOR_BGR2GRAY)
In [7]: %timeit image_to_blobs(image)
13.8 s ± 25.8 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
In [8]: from frame_blob_improved import image_to_blobs
In [9]: %timeit image_to_blobs(image)
346 ms ± 3.35 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
In [10]: □
```

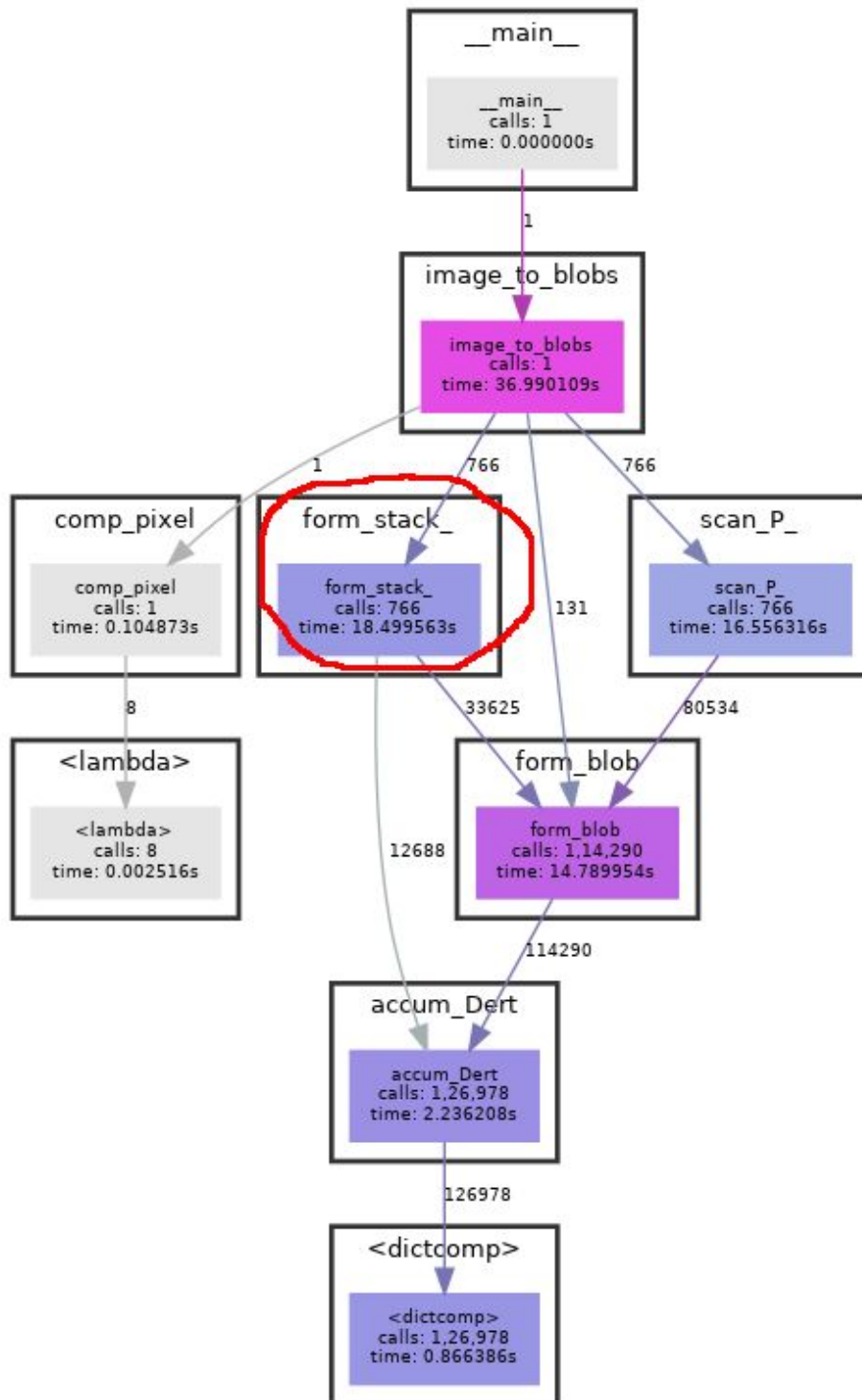
Total time for 7 runs in Original implementation: 13.8 seconds with 25.8 milliseconds of standard deviation

Total time for 7 runs in our implementation: 0.346 seconds with 3.35 milliseconds of standard deviation

Without Improvement:



With Improvement:



Performing analysis only on form_p (excluding all the function)