

CV ASSIGNMENT 4

Himansh Sheoran

20161211

Changing Number Of Iterations



Iteration 1

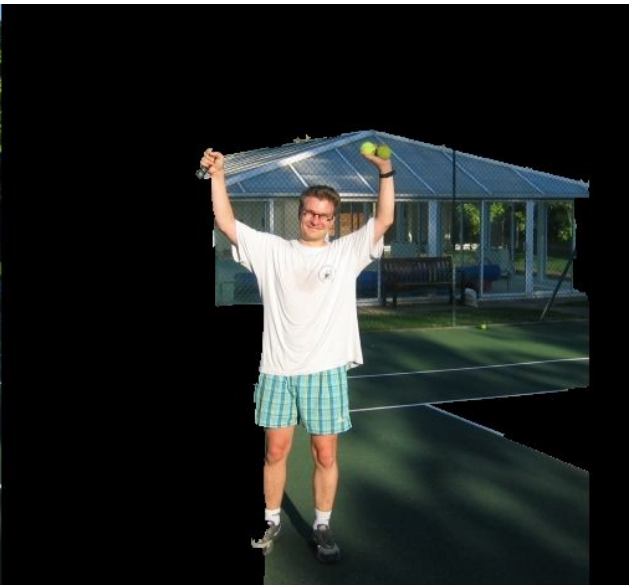


Several Iterations

Changing the number of Components



2 components



5 components



7 components

Changing Value of Gamma:



Gamma = 50



Gamma = 70



Gamma = 30

Code Snippets:

```
def construct_gc_graph(img,mask,gc_source,gc_sink,fgd_gmm,bgd_gmm,gamma,rows,cols,left_V,uyleft_V,
    bgd_indexes = np.where(mask.reshape(-1) == DRAW_BG['val'])
    fgd_indexes = np.where(mask.reshape(-1) == DRAW_FG['val'])
    pr_indexes = np.where(np.logical_or(mask.reshape(-1) == DRAW_PR_BG['val'],mask.reshape(-1) ==
    print('bgd count: %d, fgd count: %d, uncertain count: %d' % (len(bgd_indexes[0]), len(fgd_in
    edges = []
    gc_graph_capacity = []
    edges.extend(list(zip([gc_source] * pr_indexes[0].size, pr_indexes[0])))
    _D = -np.log(bgd_gmm.calc_prob(img.reshape(-1, 3)[pr_indexes]))
    gc_graph_capacity.extend(_D.tolist())
    edges.extend(list(zip([gc_sink] * pr_indexes[0].size, pr_indexes[0])))
    _D = -np.log(fgd_gmm.calc_prob(img.reshape(-1, 3)[pr_indexes]))
    gc_graph_capacity.extend(_D.tolist())
    edges.extend(list(zip([gc_source] * bgd_indexes[0].size, bgd_indexes[0])))
    gc_graph_capacity.extend([0] * bgd_indexes[0].size)
    edges.extend(list(zip([gc_sink] * bgd_indexes[0].size, bgd_indexes[0])))
    gc_graph_capacity.extend([9 * gamma] * bgd_indexes[0].size)
    edges.extend(list(zip([gc_source] * fgd_indexes[0].size, fgd_indexes[0])))
    gc_graph_capacity.extend([9 * gamma] * fgd_indexes[0].size)
    edges.extend(list(zip([gc_sink] * fgd_indexes[0].size, fgd_indexes[0])))
    gc_graph_capacity.extend([0] * fgd_indexes[0].size)
```

Code which constructs the t-links in the graph

```

img_indexes = np.arange(rows*cols,dtype=np.uint32).reshape(rows,cols)
temp1 = img_indexes[:, 1:]
temp2 = img_indexes[:, :-1]
mask1 = temp1.reshape(-1)
mask2 = temp2.reshape(-1)
edges.extend(list(zip(mask1, mask2)))
gc_graph_capacity.extend(left_V.reshape(-1).tolist())
temp1 = img_indexes[1:, 1:]
temp2 = img_indexes[:-1, :-1]
mask1 = temp1.reshape(-1)
mask2 = temp2.reshape(-1)
edges.extend(list(zip(mask1, mask2)))
gc_graph_capacity.extend(upleft_V.reshape(-1).tolist())
temp1 = img_indexes[1:, :]
temp2 = img_indexes[:-1, :]
mask1 = temp1.reshape(-1)
mask2 = temp2.reshape(-1)
edges.extend(list(zip(mask1, mask2)))
gc_graph_capacity.extend(up_V.reshape(-1).tolist())
temp1 = img_indexes[1:, :-1]
temp2 = img_indexes[:-1, 1:]
mask1 = temp1.reshape(-1)
mask2 = temp2.reshape(-1)
edges.extend(list(zip(mask1, mask2)))
gc_graph_capacity.extend(upright_V.reshape(-1).tolist())
gc_graph = ig.Graph(cols * rows + 2)
gc_graph.add_edges(edges)
return gc_graph,gc_source,gc_sink,gc_graph_capacity

```

Code which construct the n-links

```

def estimate_segmentation(mask,gc_graph,gc_source,gc_sink,gc_graph_capacity,rows,cols):
    mincut = gc_graph.st_mincut(gc_source,gc_sink, gc_graph_capacity)
    print('foreground pixels: %d, background pixels: %d' % (len(mincut.partition[0]), len(mincut.
pr_indexes = np.where(np.logical_or(mask == DRAW_PR_BG['val'], mask == DRAW_PR_FG['val'])))
img_indexes = np.arange(rows * cols,dtype=np.uint32).reshape(rows, cols)
mask[pr_indexes] = np.where(np.isin(img_indexes[pr_indexes], mincut.partition[0]),DRAW_PR_FG[
bgd_indexes = np.where(np.logical_or(mask == DRAW_BG['val'],mask == DRAW_PR_BG['val'])))
fgd_indexes = np.where(np.logical_or(mask == DRAW_FG['val'],mask == DRAW_PR_FG['val'])))
print('probable background count: %d, probable foreground count: %d' % (bgd_indexes[0].size,fg
return pr_indexes,img_indexes,mask,bgd_indexes,fgd_indexes

```

Code which estimate segmentations by using min-cut algorithm.

Interactive Segmentation:

