DSA PRESENTATION

IMAGE SEGMENTATION

OVERVIEW

INTRODUCTION

IMAGE SEGMENTATION IS A COMPUTER VISION TECHNIQUE THAT INVOLVES PARTITIONING AN IMAGE INTO MULTIPLE DISTINCT REGIONS OR SEGMENTS BASED ON CERTAIN CHARACTERISTICS, SUCH AS COLOR, TEXTURE, OR SHAPE. THE GOAL OF IMAGE SEGMENTATION IS TO SEPARATE DIFFERENT OBJECTS OR REGIONS WITHIN AN IMAGE TO FACILITATE THEIR ANALYSIS, UNDERSTANDING, OR FURTHER PROCESSING.

PROBLEM STATEMENT

TO DEVELOP AN IMAGE SEGMENTATION ALGORITHM THAT COMBINES SPEED, EASE OF IMPLEMENTATION, AND OVERSEGMENTATION. THE ALGORITHM SHOULD EFFICIENTLY PARTITION IMAGES INTO REGIONS BY MINIMIZING AN ENERGY FUNCTION, HANDLING COMPLEX OBJECT SHAPES, AND PROVIDING ACCURATE FOREGROUND–BACKGROUND SEPARATION, WHILE MAINTAINING HIGH SPEED AND SIMPLICITY FOR PRACTICAL IMPLEMENTATION.

ANALYSIS OF THEOREM

GRAPH CUTS

- CONSTRUCT A GRAPH(V,E) FROM THE GIVEN IMAGE WITH V AS NODE AND E AS EDGES WITH WEIGHTS AND DIRECTION

- WEIGHTS FOR A GIVEN NODE I(X,Y) WITH OTHER NODE J (X,Y) IS GIVEN BY :

CRITERIA

$$\frac{1}{\sqrt{(x_i-x_j)^2+(y_i-y_j)^2)}}\, e^{-\frac{\|f(x_i,y_i)-f(x_j,y_j)\|}{2\sigma^2}}$$

HOW TO ASSIGN WEIGHTS

SUPER NODES

WE WILL HAVE TWO SUPER NODES

- FOR BACKGOUND
- FOR FOREGROUND

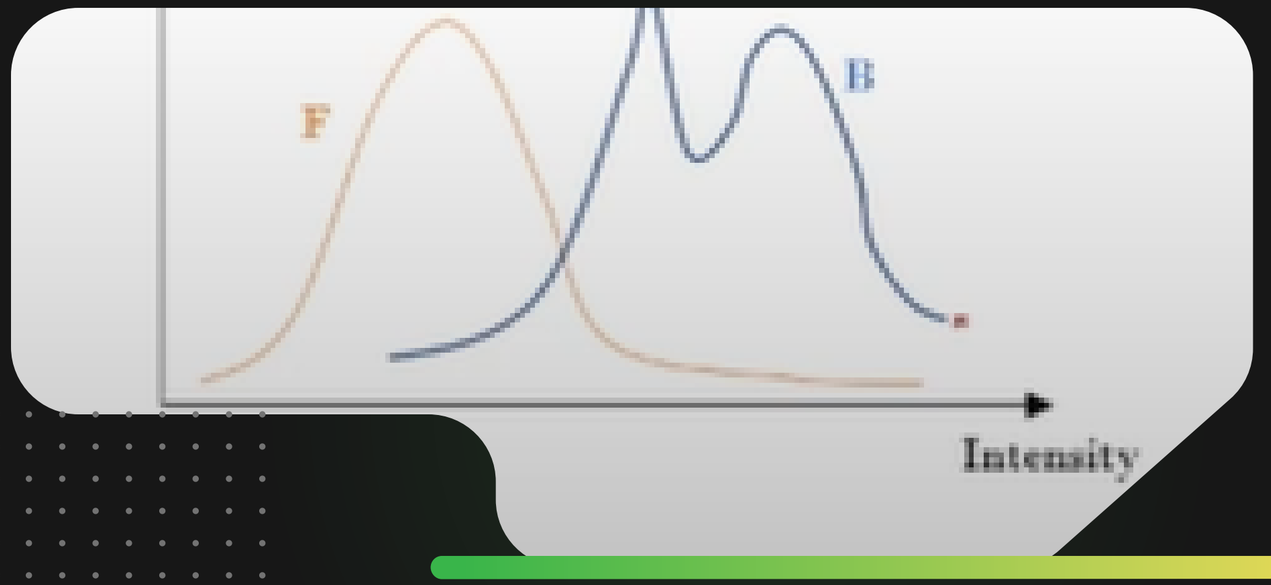
FOR FOREGROUND AND BACKGROUND

- FOREGROUND : WIF = INFINITY AND WIB = 0
- BACKGROUND: WIF = 0 NAD WIB = INFINITY

FOR NON-SCRIBLED

WE WILL USE THE PROBABILTY DISTRIBUTION FOR PF AND PB

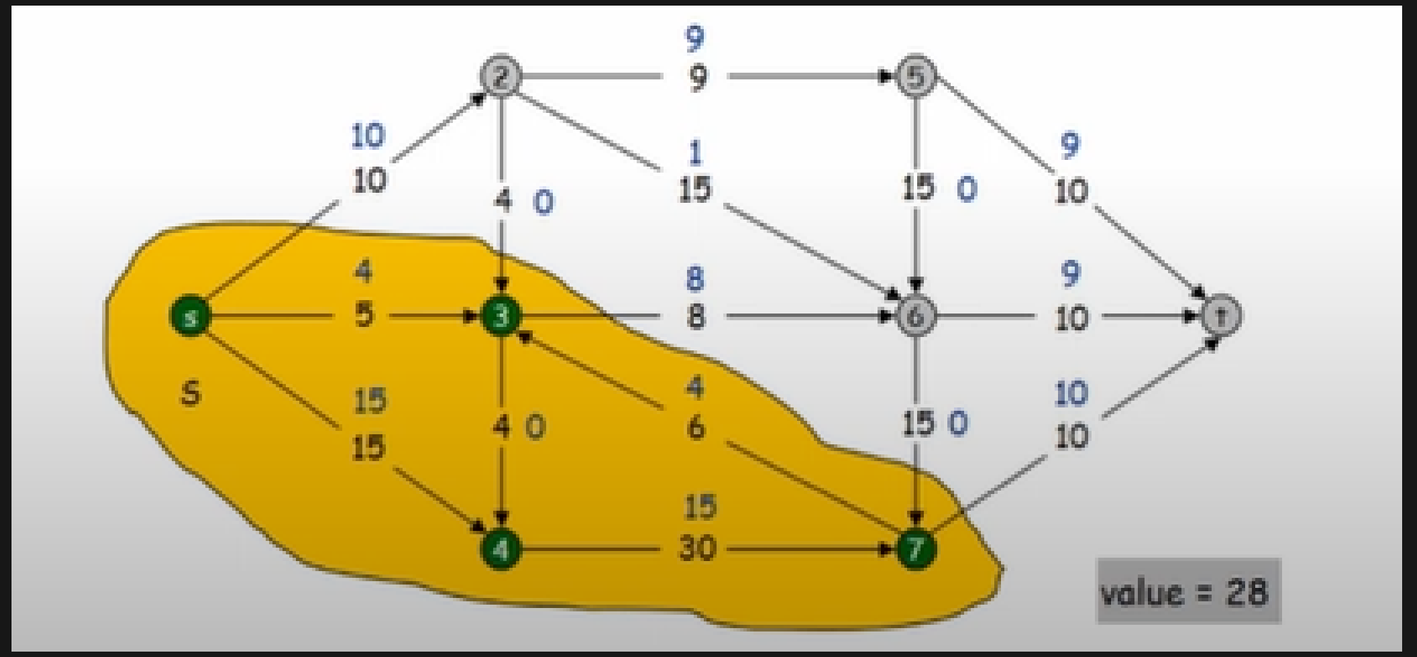
- WIF = -LAMBDA LOG PB(I)
- WIB = -LAMBDA LOG PF(I)



MIN-CUT MAX-FLOW

ADVANTAGES

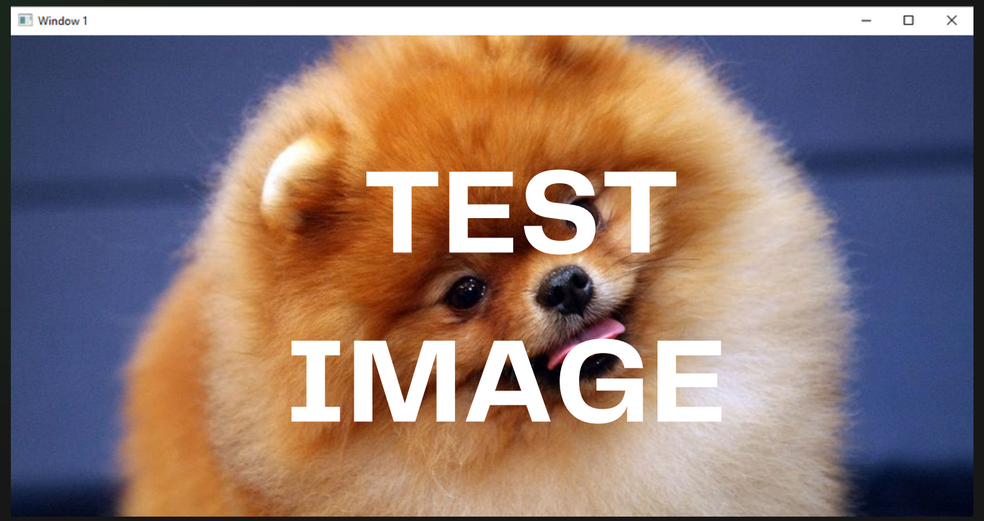
- OVER-SEGMENTATION
- SPEED AND EFFICIENCY
- SIMPLICITY
- EASE OF IMPLEMENTATION



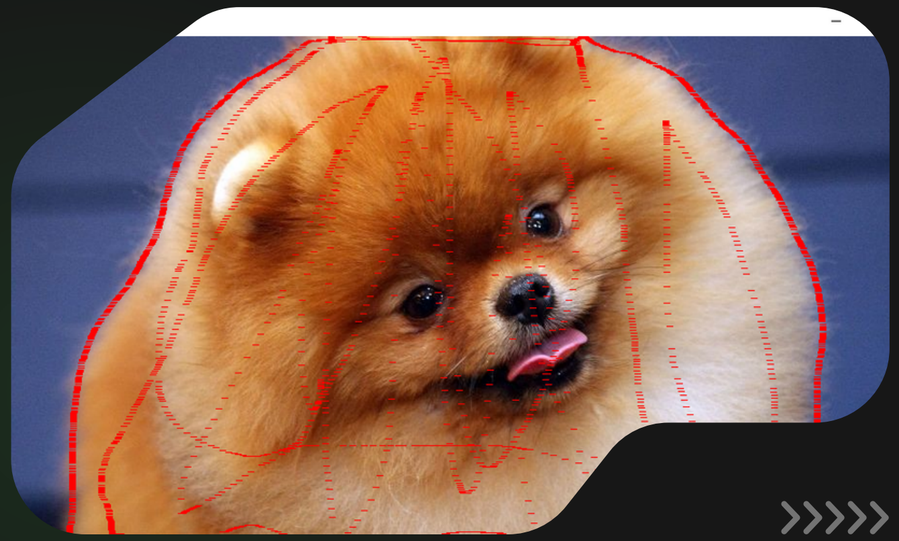
METHODOLOGY

HOW TO START

PRESS 'O' TO START MARKING OBJECTS (IN RED) AND PRESS 'B' TO START MARKING THE BACKGROUND (IN BLUE). PRESS 'ESC' WHEN DONE



STEP 2



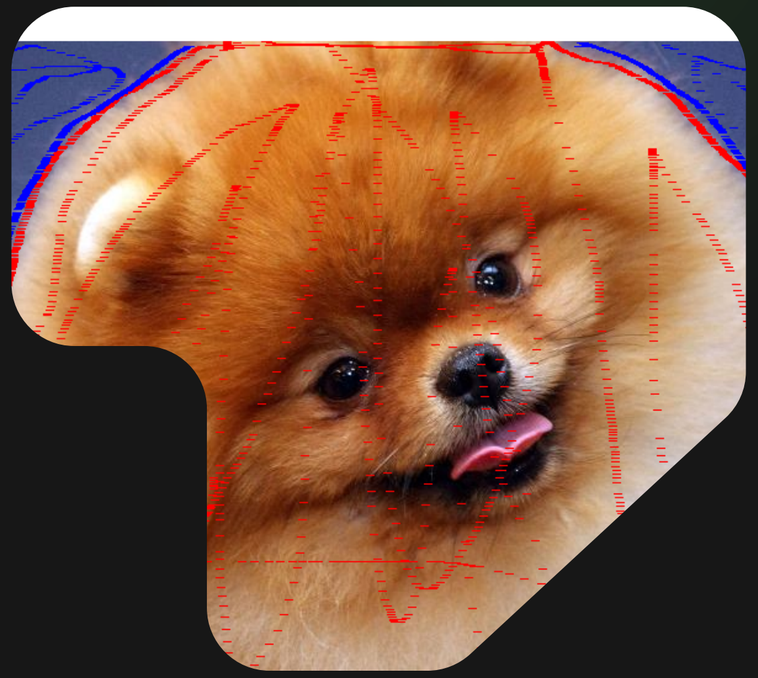
1

THE GROUPS OF PIXELS DEFINED IN A CLASS 'SPNODE' ARE MADE

2

EACH GROUPS PROPERTIES ARE ALSO CALCULATED (THE CENTROID, COLOR ETC)

THE ALGORITHM



1

THERE ARE TWO TREES – SOURCE TREE AND TARGET TREE

2

INITIALLY THE SOURCE TREE CONTAINS ONLY ONE NODE THE SOURCE NODE AND TARGET TREE ALSO CONTAINS ONLY ONE NODE THE TARGET NODE

3

THE ALGORITHMS BASICALLY FINDS SHORTEST PATH FROM SOURCE TREE TO THE TARGET TREE

4

EACH NODE CAN EITHER BE 'ACTIVE ' OR 'PASSIVE'

GROW PHASE

1

IN THIS PHASE ACTIVE NODES ARE ALLOWED TO ACQUIRE NEIGHOUR NODES THAT ARE 'SIMILAR'

2

ACQUIRING MEANS THOSE NODES HAVE BECOME ACTIVE AND NOW BELONG TO THE CURRENT NODE

3

IF THERE ARE NO MORE VALID NEIGHOURS, THE CURRENT NODE BECOMES PASSIVE AND THE GROW PHASE CONTINUES TO THE NEXT ACTIVE NODE

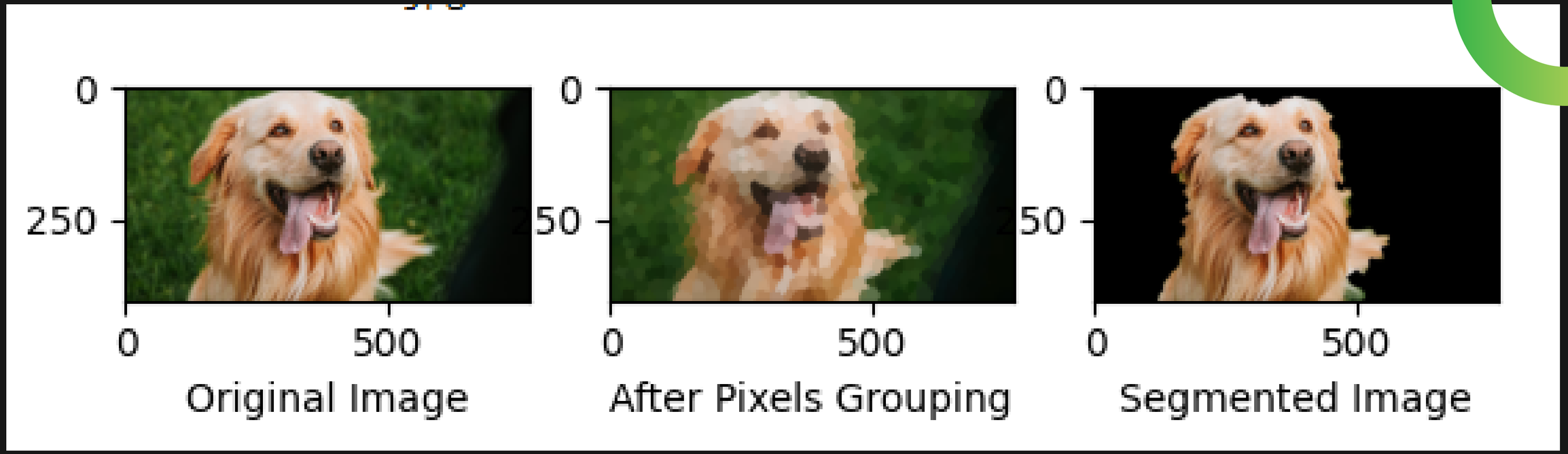
PHASE 2 AUGMENT PHASE

THIS PHASE AUGMENTS THE PATH THAT WAS FOUND IN THE GROW PHASE BY BREAKING DOWN THE TREES

PHASE 3 ADOPTION PHASE

- THE TREES THAT WERE BROKEN DOWN ARE PUT BACK TOGETHER
- BUT WE HAVEJUST CONSIDERED ALL THESE SEPARATED PARTS AS FREE NODES AND PUT THEM BACK INTO THE TREES
- THESE PHASE TERMINATES WHEN THERE ARE NO MORE DISCONNECTED TREES

Results


0
250
0
500
Original Image
0
250
0
500
After Pixels Grouping
0
250
0
500
Segmented Image

Home Page - Select or create a | dsa project - Jupyter Notebook

localhost:8888/notebooks/dsa%20project.ipynb

*DEMO

jupyter dsa project Last Checkpoint: 04/30/2023 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [4]:

```
import cv2
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import math
import Min_Cut

class SPNode():
    def __init__(self):
        self.label = None
        self.pixels = []
        self.centroid = ()
        self.type = 'na'
        self.mean_lab = None
        self.lab_hist = None
        self.real_lab = None

def mark_pixels(event, x, y, flags, param):
    global drawing, mode, marked_bg_pixels, marked_ob_pixels, I_dummy
    h, w, c = I_dummy.shape

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing == True:
            if mode == "ob":
                if x >= 0 and x <= w-1 and y > 0 and y <= h-1:
                    marked_ob_pixels.append((y, x))
                    cv2.line(I_dummy, (x-3, y), (x+3, y), (0, 0, 255))
            else:
                if x >= 0 and x <= w-1 and y > 0 and y <= h-1:
```

APPLICATIONS

1. IMAGE AND VIDEO EDITING
2. OBJECT TRACKING AND RECOGNITION
3. VIDEO COMPRESSION
4. AUGMENTED REALITY
5. SELF-DRIVEN CARS

FUTURE TRENDS

INTEGRATION OF DEEP LEARNING TECHNIQUES

DEEP LEARNING HAS SHOWN GREAT POTENTIAL IN VARIOUS COMPUTER VISION TASKS, INCLUDING IMAGE SEGMENTATION.

HANDLING COMPLEX OBJECT INTERACTIONS

FUTURE RESEARCH CAN FOCUS ON DEVELOPING MORE EFFICIENT MAXIMUM FLOW ALGORITHMS OR EXPLORING PARALLELIZATION TECHNIQUES TO HANDLE LARGER IMAGES OR VIDEOS IN REAL-TIME.

HANDLING MULTI-MODAL DATA

IMAGE SEGMENTATION OFTEN ENCOUNTERS CHALLENGES WHEN DEALING WITH MULTI-MODAL DATA, SUCH AS MEDICAL IMAGING OR REMOTE SENSING DATA.

CONCLUSION

The primary objective of the algorithm is to accurately segment the image into foreground and background regions. The algorithm assigns labels to pixels based on their connectivity and pixel similarity, leading to a clear separation between the two regions. The result is a segmentation mask that accurately identifies the foreground and background areas

THANKYOU