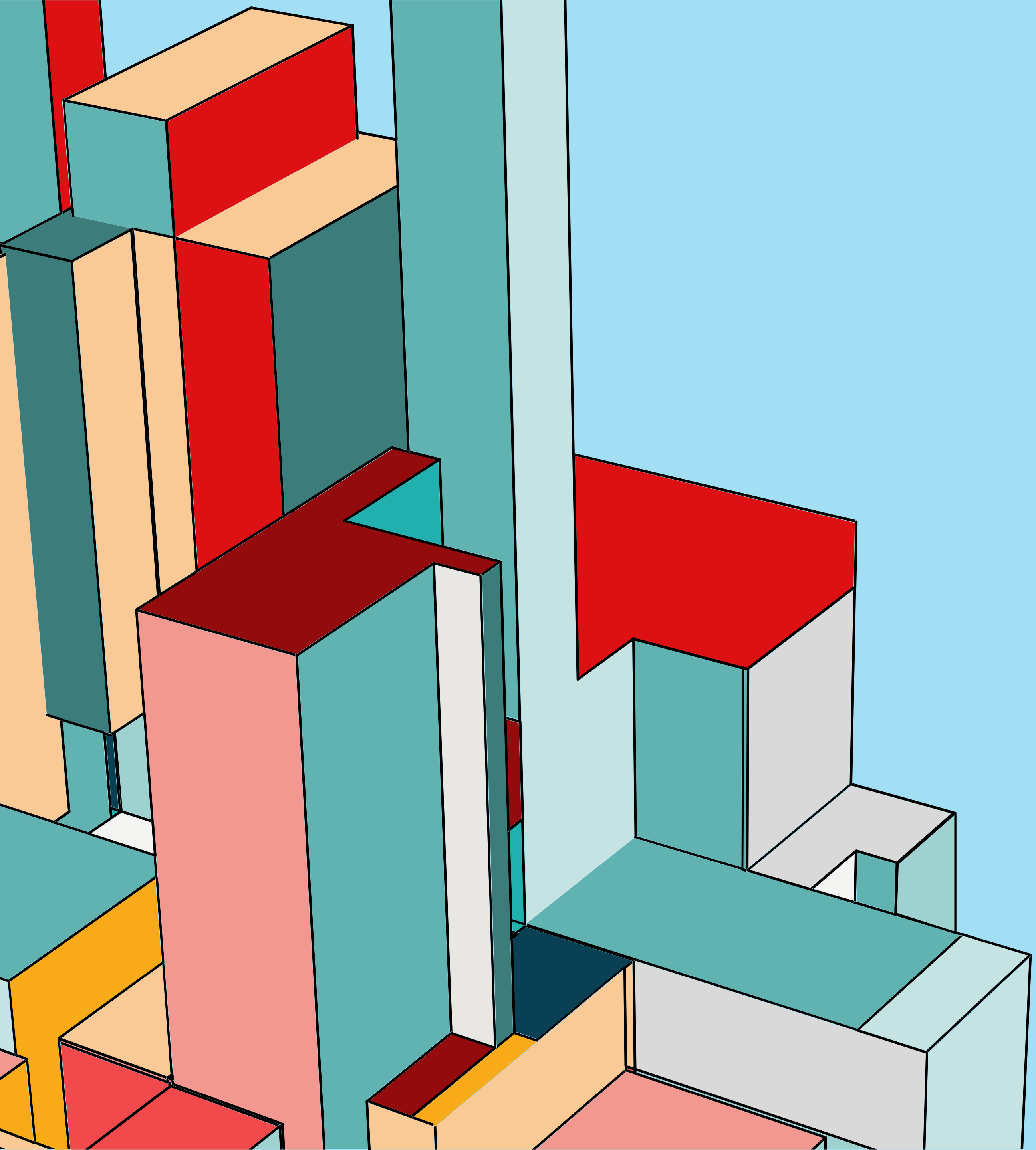
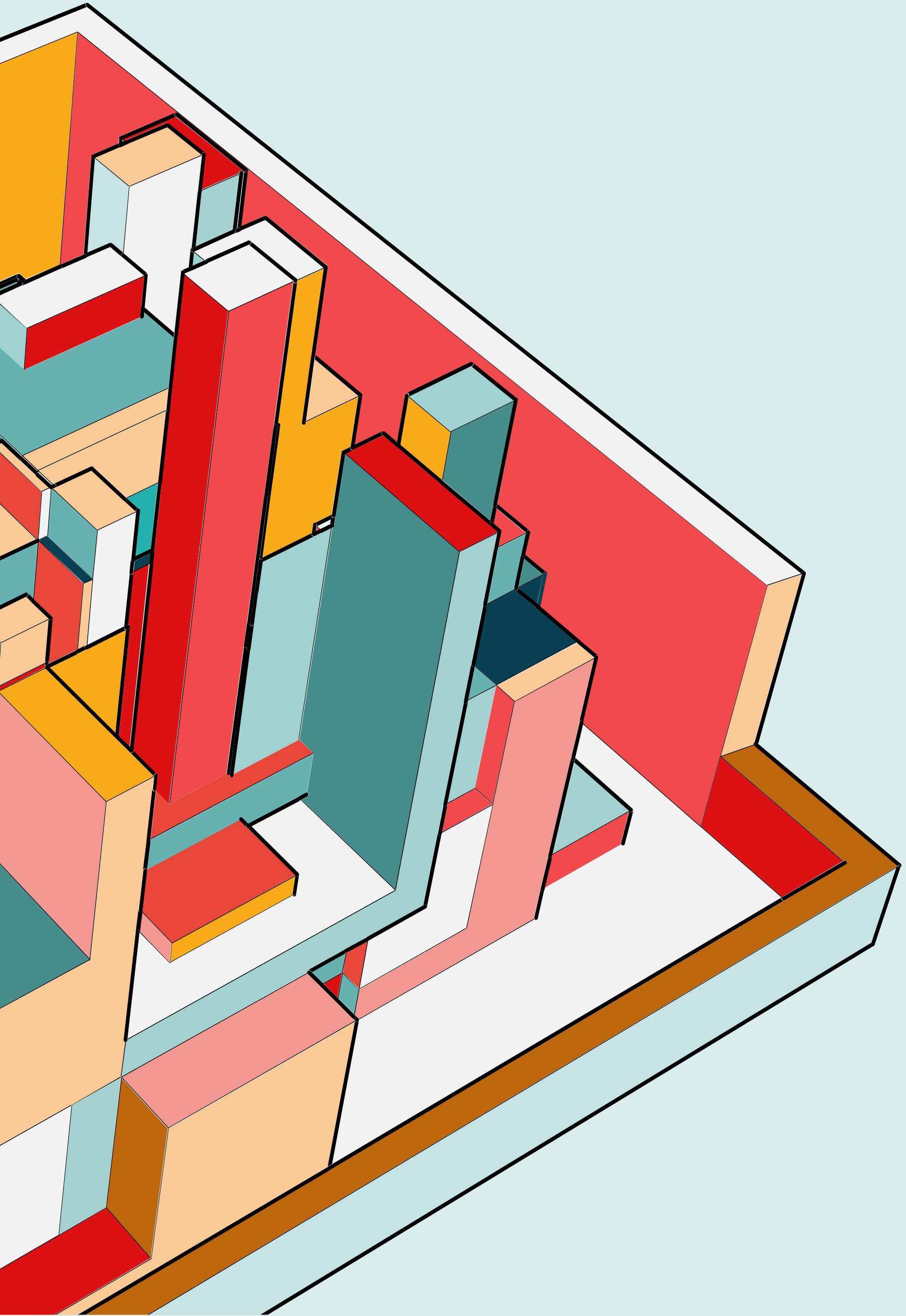


EVERYTHING IS A GRAPH

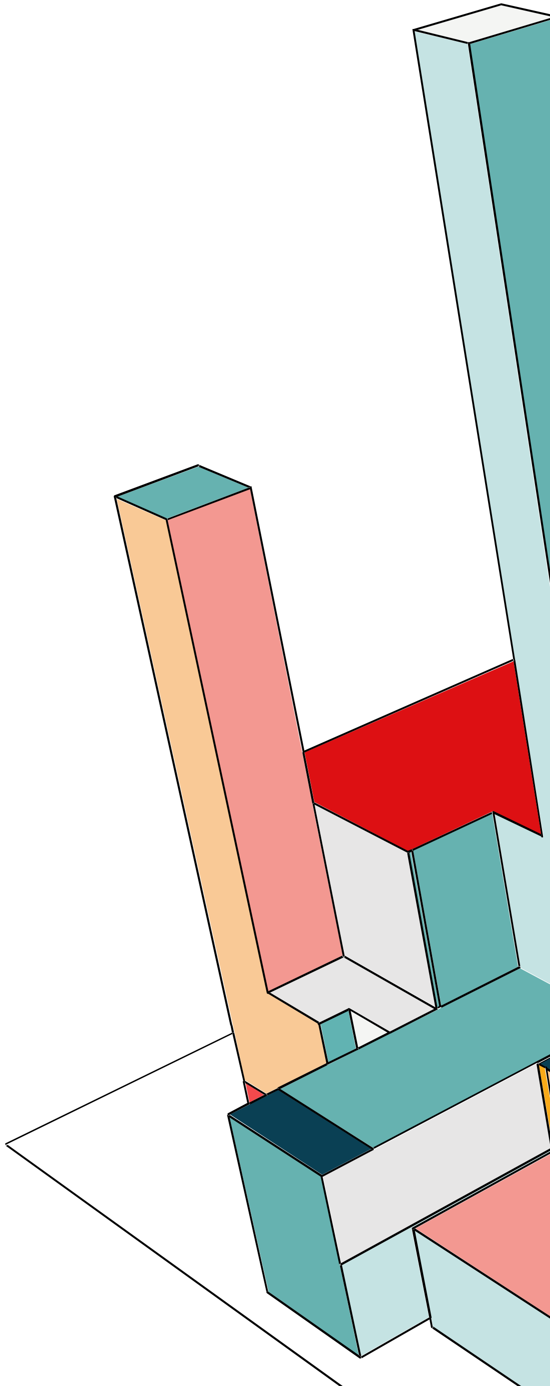
Joshua Piña
CSC 4520: Design & Analysis of Algorithms
Data Science, CS Department, GSU



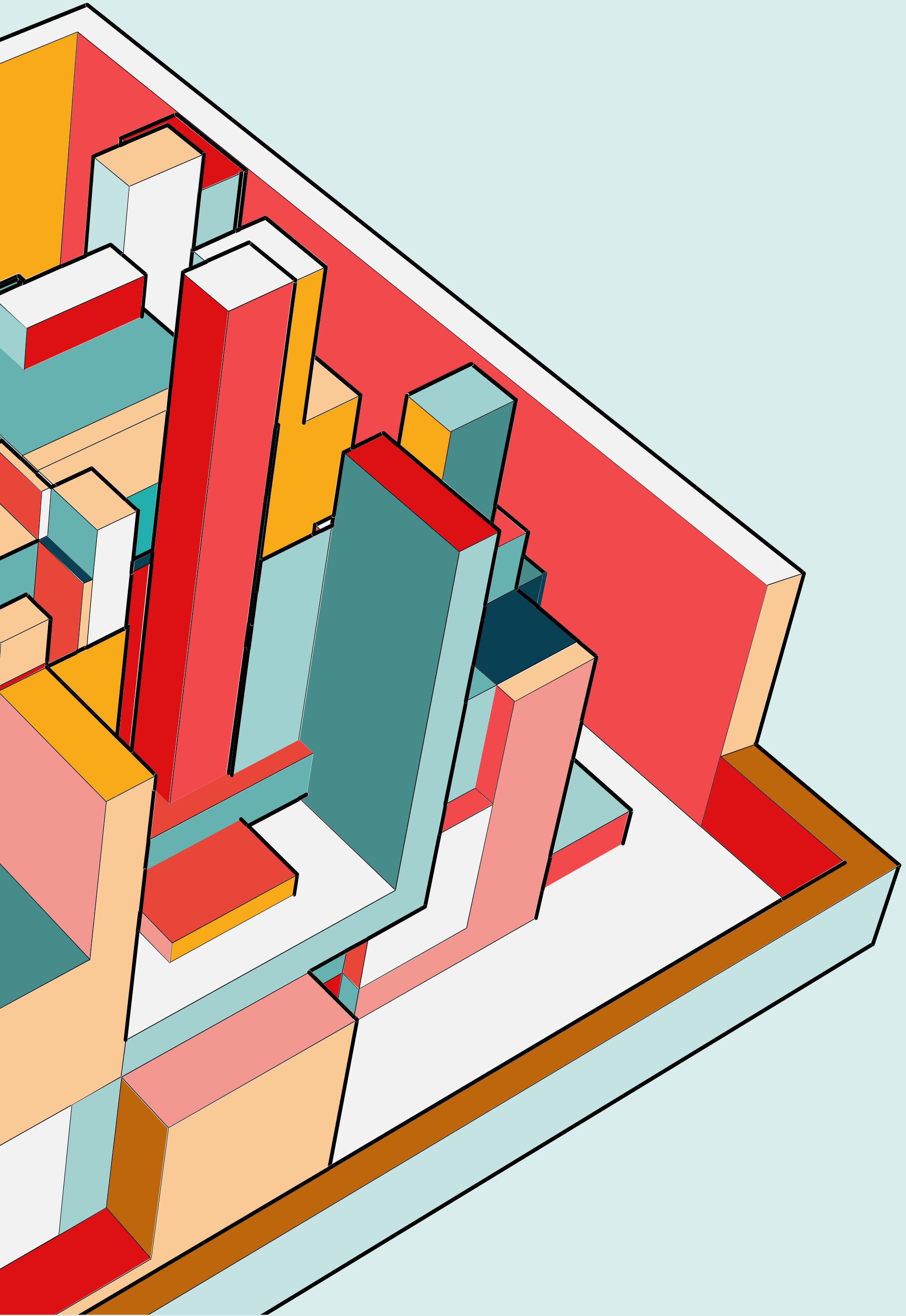
ABSTRACT

ABSTRACT

While modern image segmentation increasingly relies on Deep Learning, classical algorithmic approaches remain fundamental for their interpretability and mathematical robustness. Motivated by the challenges of binary segmentation explored in Digital Image Processing, this project implements a solution using the **Max-Flow Min-Cut theorem**, rather than a neural network. The project models a digital image as a flow network where individual pixels function as nodes in a grid graph. The network is constructed using n-links (neighbor connections) and t-links (terminal connections). Edge capacities between pixel nodes are calculated using an **exponential decay function** based on pixel intensity differences ($w = e^{-|I_u - I_v|/\sigma}$), which penalizes cuts through uniform regions while incentivizing cuts along high-contrast edges. By implementing the **Edmonds-Karp algorithm**, the project computes the maximum flow from a defined Source (foreground seed) to a Sink (background seed). The resulting residual graph reveals the Minimum Cut, effectively isolating the object of interest with minimal energy cost. The final application demonstrates successful segmentation of complex, non-convex shapes from grayscale images, validating the efficacy of classical graph algorithms on unstructured visual data.



AGENDA



INTRODUCTION

INTRODUCTION

Motivation:

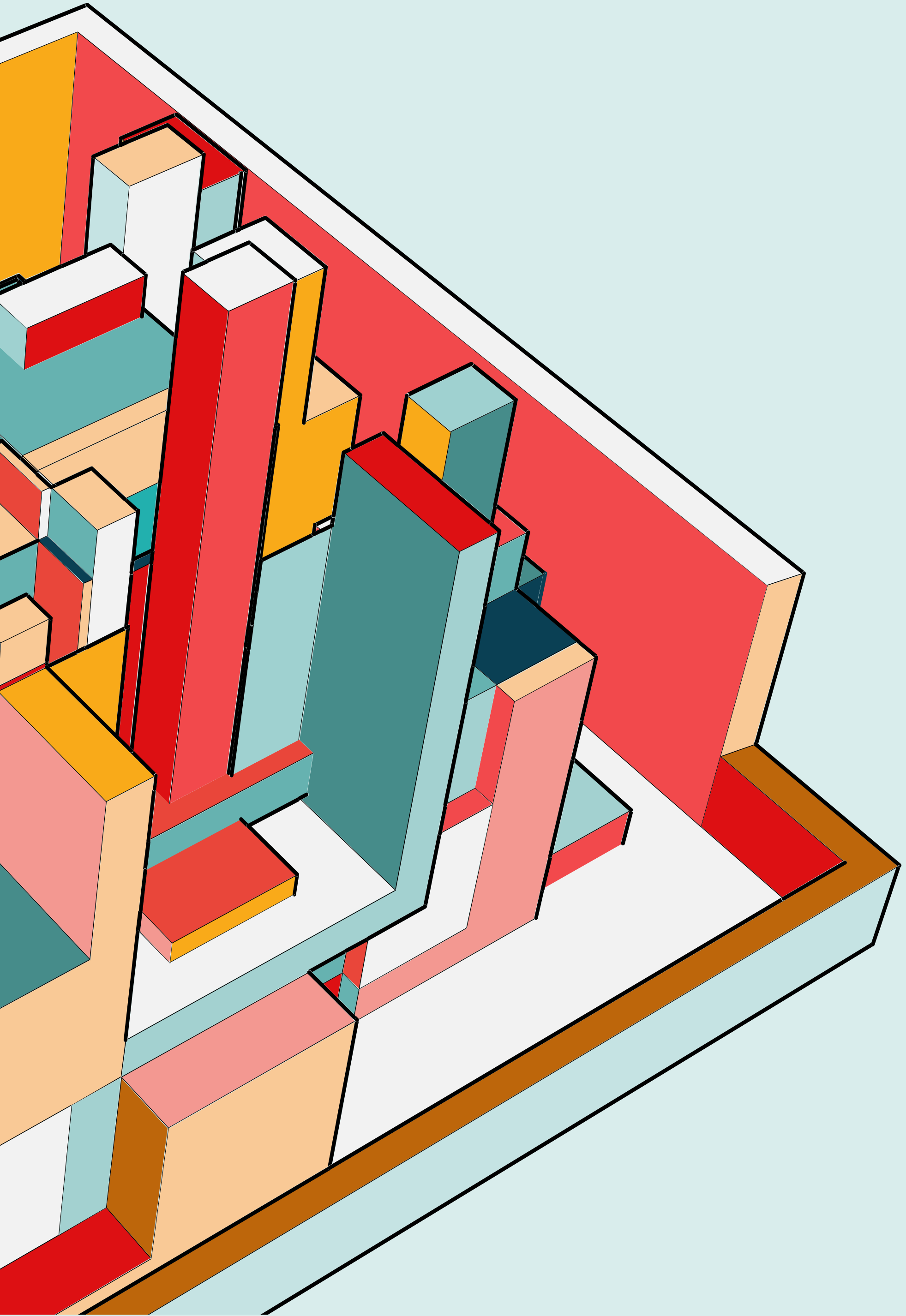
1) Deep Learning dominates modern Computer Vision, but Neural Networks lack intuitive interpretability.
   - *I would like to understand the "why" behind segmentation*
2) Classical algorithms provide provable solutions.
   - *Max-Flow Min-Cut Theorem offers rigorous theoretical foundation.*
3) Eliminates the need for large training datasets required for Deep Learning.
   - *Works immediately for single instances*

Problem Statement:

1) Core Task: Binary segmentation of an image into foreground and background.

2) I need to figure out a way to convert a grid of pixels into a graph flow network.

3) I need to find the cheapest cut to separate the foreground from the background.



KEY TERMS & CONCEPTS

A Quick Refresher

KEY TERMS & CONCEPTS REFRESHER

- Breadth-First Search
- Edmonds-Karp
- Exponential Decay
- Graph Cut
- Max-Flow Min-Cut Theorem
- Network Flow
- Thresholding

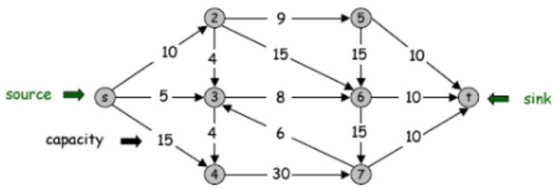
- G = (V, E): Graph w/ Vertices & Edges
- s, t: Source (Object center) & Sink (Background)
- c(u,v): Capacity of the edge b/t nodes u & v
- f(u,v): Current flow through the edge
- Cut(S,T): Partitioning nodes into two disjoint sets

Single-source shortest paths

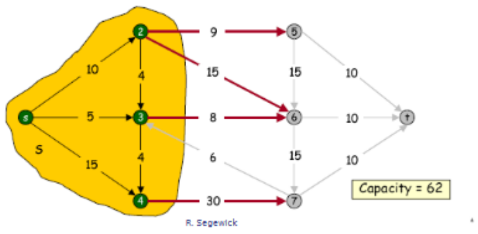
Input: Graph G = (V,E), weights w: E → **R,** vertex s

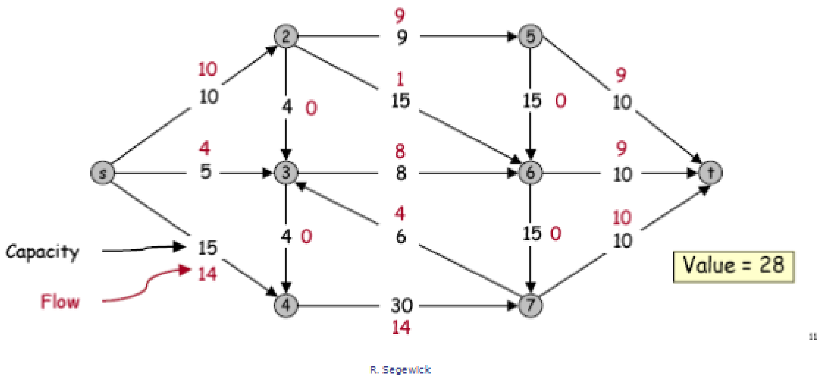
Find: Distances from s to all other vertices

KEY TERMS & CONCEPTS REFRESHER



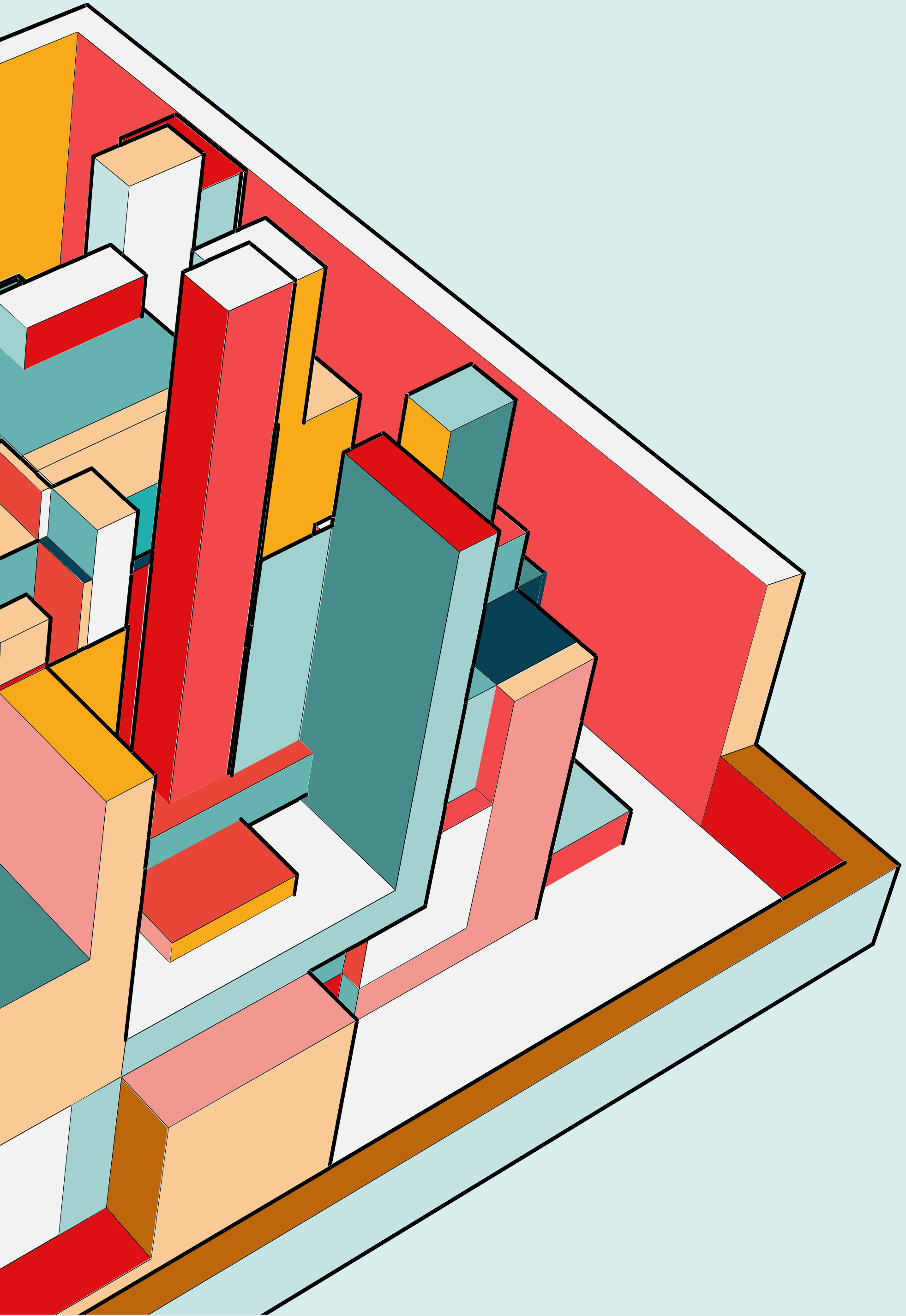
Cut - a partition of vertex set into two subsets (S T) such that $s \in S$ and $t \in T$.





Problem: find maximum flow

Ford-Fulkerson Theorem. For every network value of max-flow is equal to the capacity of min-cut



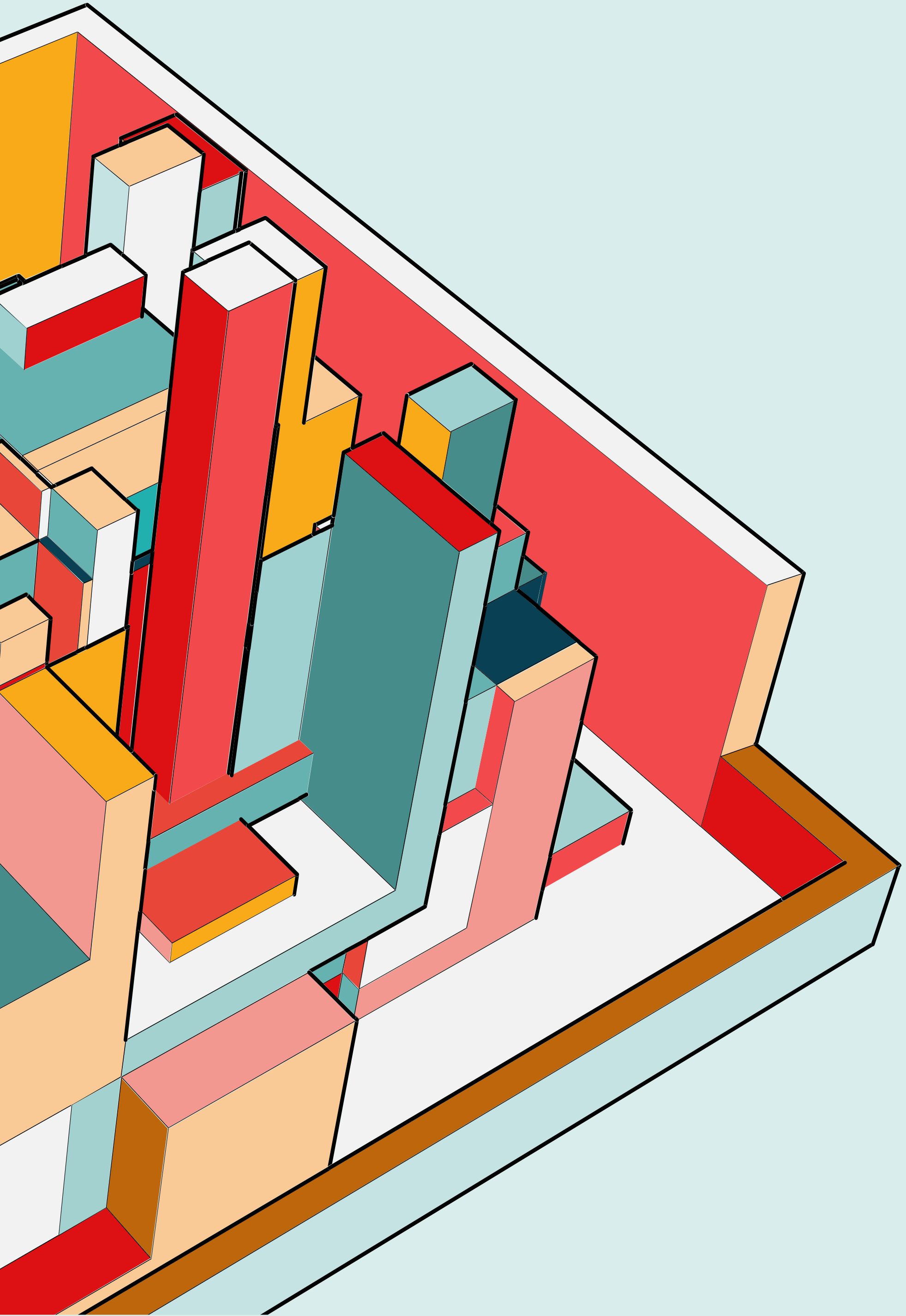
PROBLEM FORMULATION

PROBLEM FORMULATION

- Graph Construction G=(V, E):
  - **Vertices (V):** Every image pixel becomes a node.
  - **Terminals:** Two special nodes added: Source (s) representing the object, Sink (t) representing the background.

- Edge Definitions (E):
  - **T-links (Terminal links):** Connect pixels to s or t. Represent the *likelihood* of a pixel belonging to foreground or background.
  - **N-links (Neighbor links):** Connect adjacent pixels (4-connectivity or 8-connectivity). Represent pixel *similarity*.

- **Goal:** Find the Minimum Cut that separates s from t, partitioning the image nodes.



METHODOLOGY

METHODOLOGY

LANGUAGE

PYTHON

quality | trust me bro

LIBRARIES USED

NUMPY RICH OPENCV

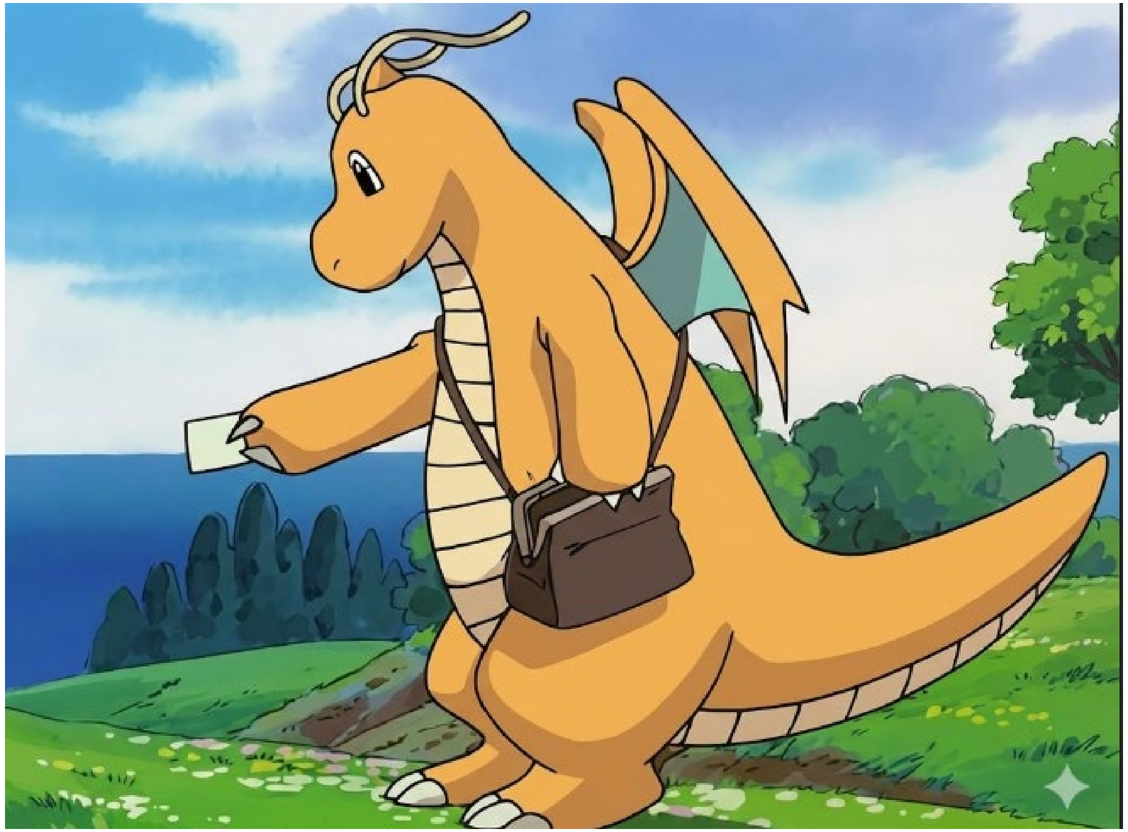
MATPLOTLIB

METHODOLOGY

- **The Challenge:** Defining pixel/node similarity mathematically to create strong boundaries.

- **Initial Approach (Failed):** Using simple intensity differences (e.g., |px_A - px_B|).
    - *Resulted in weak boundaries and "leaky" segmentations.*

- **Next Approach (The Solution?):** Exponential Decay Function.
    - Capacities drop sharply as intensity differences increase.
    - This allows for soft labels that can improve accuracy and reduce noise.



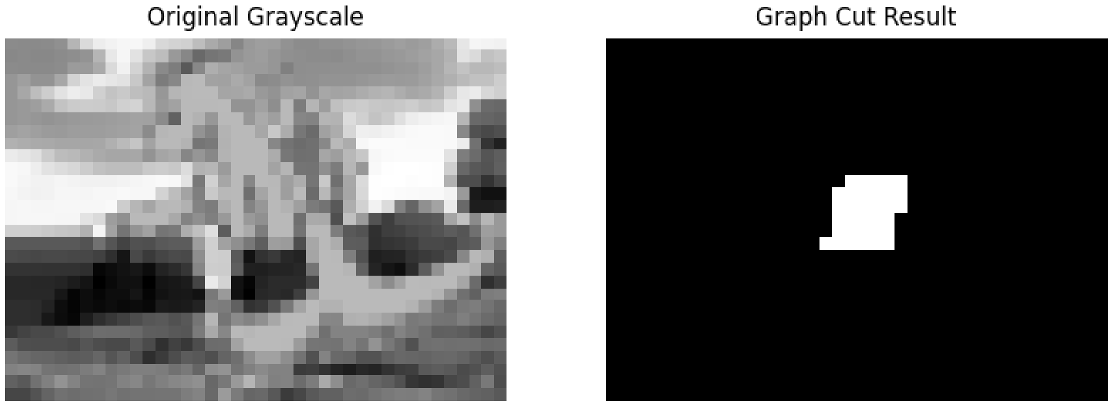
METHODOLOGY

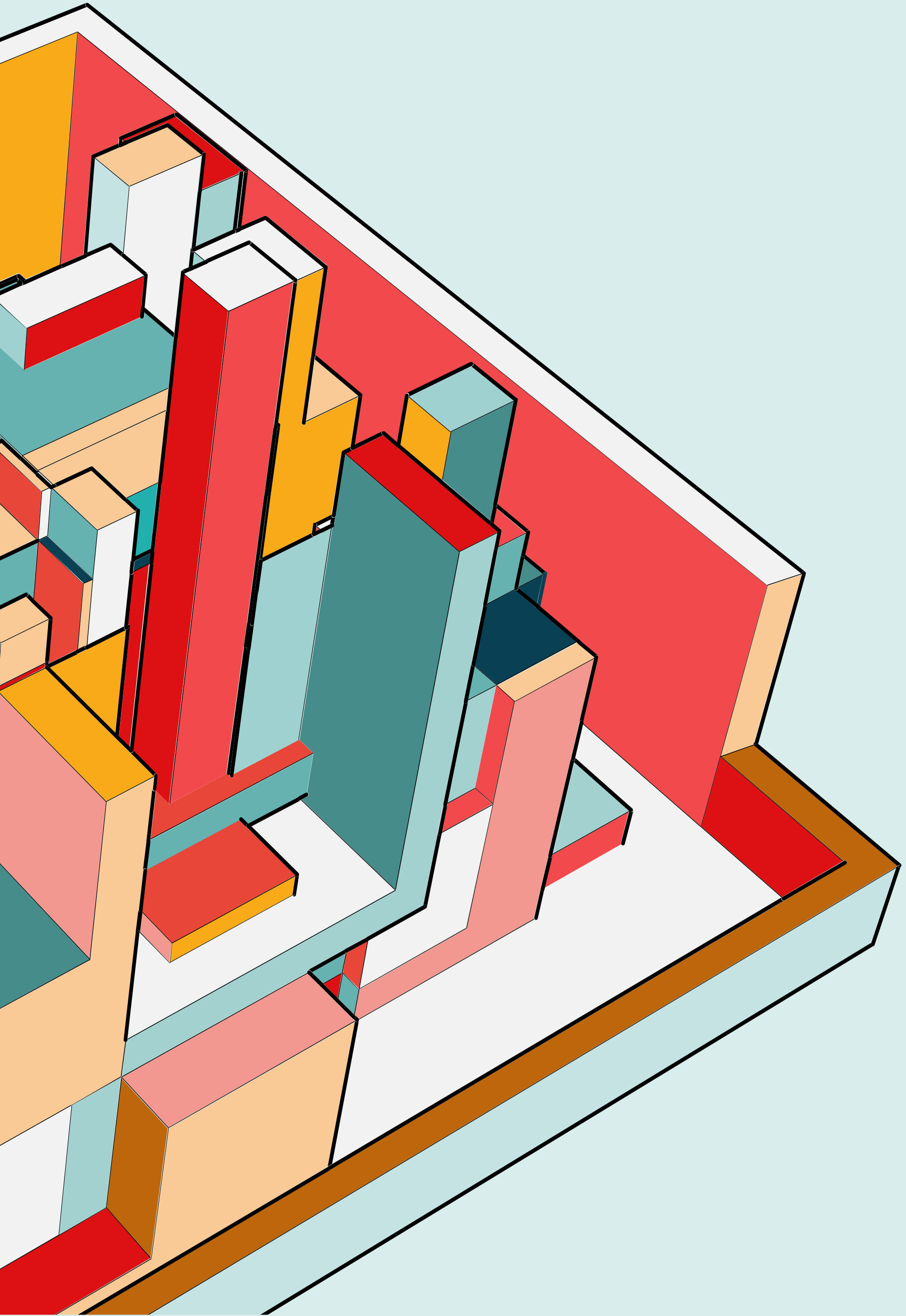
The Edmonds-Karp Algorithm

- **Theoretical Basis:** Max-Flow Min-Cut Theorem.
  - We find the max flow to identify the min cut (the object boundary).

- **Implementation:** Edmonds-Karp.
  - Uses **Breadth-First Search (BFS)** to find the shortest augmenting path in the residual graph at each step.

- **Process:**
  1. Initialize flow to 0.
  2. While there is an augmenting path from s to t (found via BFS):
  3. Push maximum possible flow along that path.
  4. Update residual graph.



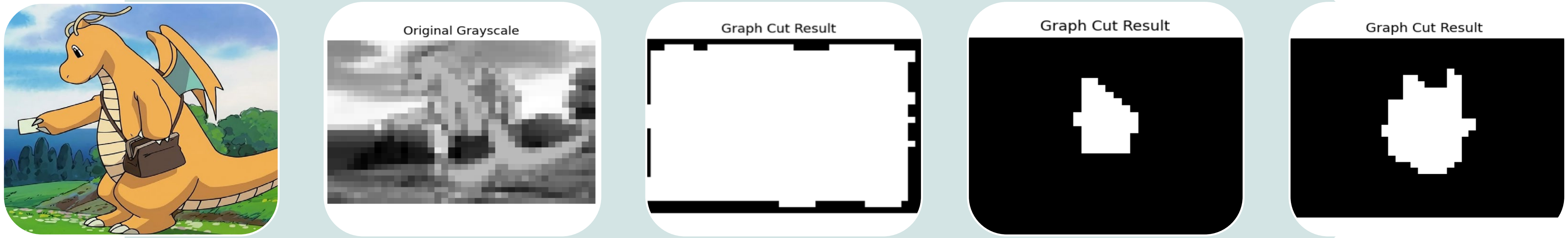


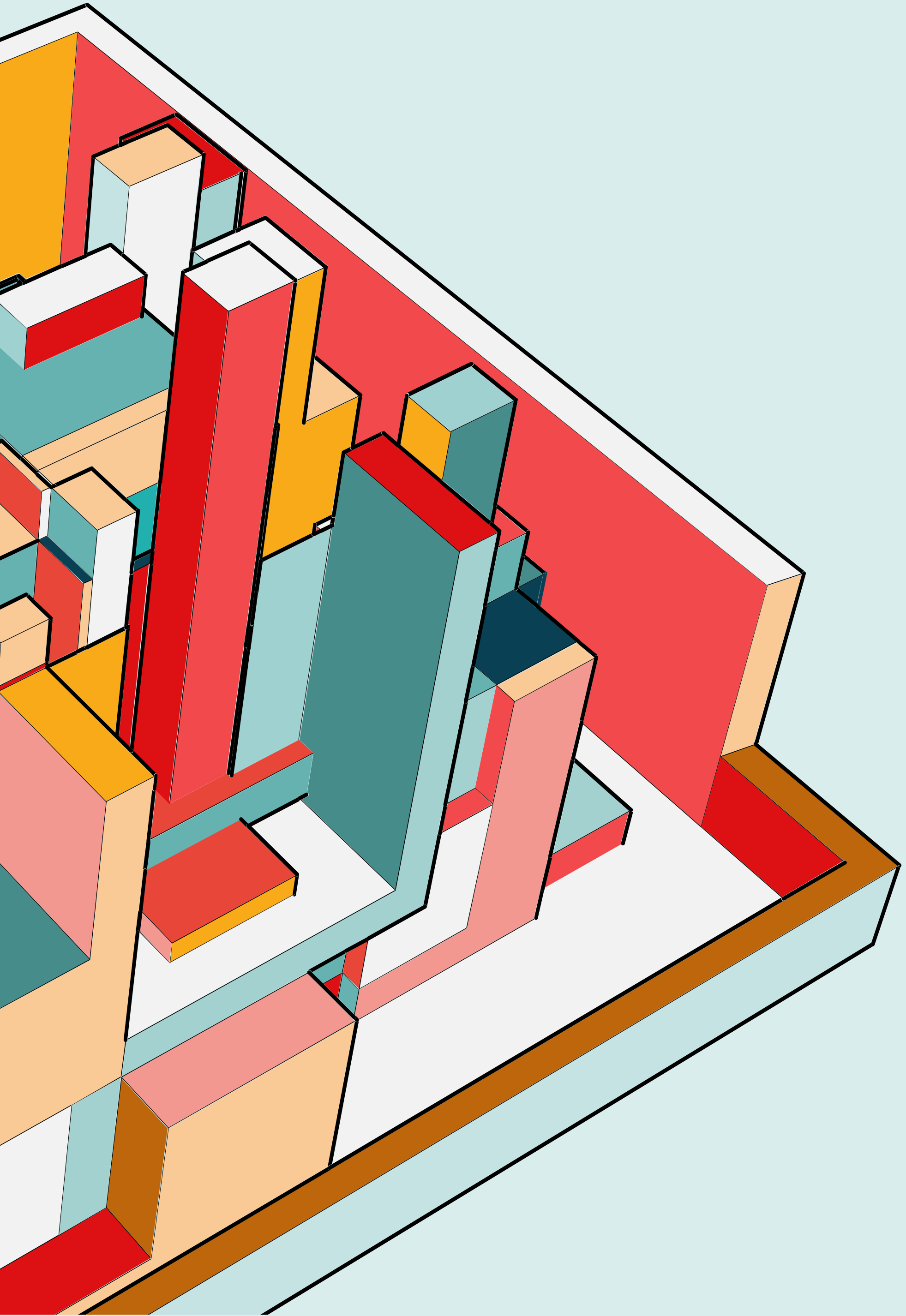
RESULTS

RESULTS

For my initial implementation, I chose a complicated image.

- The first step was to convert the image to grayscale
- Initial cut was aimed at identifying a background
- I then tried different methods of deciding what weights were safe to cut
- Iterations 3 and 4 of the script included the Exponential Decay Function.

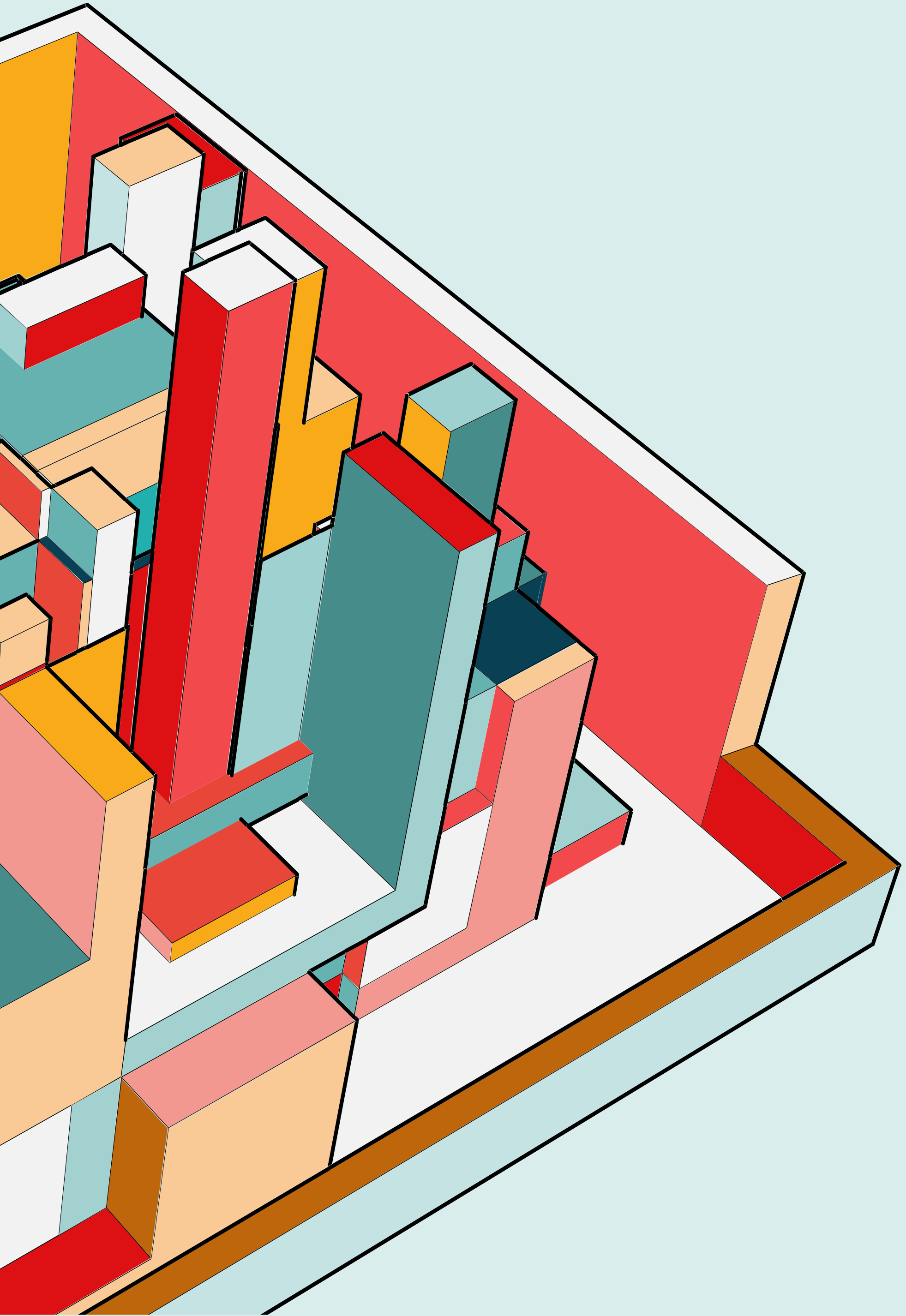




GLOSSARY

GLOSSARY

- **Augmenting Path:** A path from the source (s) to the sink (t) in the residual graph along which additional flow can be pushed.
- **Breadth-First Search (BFS):** A graph traversal algorithm that explores vertices layer by layer. It is used in the Edmonds-Karp algorithm to find the shortest augmenting path.
- **Capacity (c):** The maximum amount of flow that can pass through a specific edge in a network.
- **Edmonds-Karp Algorithm:** A specific implementation of the Ford-Fulkerson method that uses BFS to find augmenting paths, guaranteeing polynomial time complexity.
- **Exponential Decay:** A mathematical function used to define edge capacities based on pixel similarity. It causes capacity to decrease rapidly as the intensity difference between pixels increases, creating natural boundaries.
- **Flow Network:** A directed graph where each edge has a defined capacity and receives a flow that does not exceed that capacity.
- **Graph Cut:** The partitioning of the vertices of a graph into two disjoint sets. In image segmentation, this separates the foreground from the background.
- **Max-Flow:** The maximum amount of flow that can be passed from a source to a sink in a network.
- **Max-Flow Min-Cut Theorem:** A fundamental theorem stating that the maximum flow value from a source to a sink is exactly equal to the capacity of the minimum cut separating them [1].
- **Min-Cut:** The cut that separates the source from the sink with the minimum possible total capacity among all such cuts.
- **Network Flow:** The general concept of moving a substance (like data, water, or electrical current) through a network of interconnected nodes and edges.
- **Residual Graph:** A graph that shows the remaining available capacity on each edge after some flow has been pushed. It is used to find augmenting paths.
- **Segmentation:** The process of partitioning a digital image into multiple segments (sets of pixels), typically to locate objects and define their boundaries.
- **Sink (t):** The terminal node in a flow network where flow is absorbed. In image segmentation, it represents the image background.
- **Source (s):** The terminal node in a flow network where flow is generated. In image segmentation, it represents the center of the target object.
- **Thresholding:** A basic segmentation technique that creates a binary image by setting pixel values to 0 or 1 based on whether they are below or above a specific threshold value.



REFERENCE

REFERENCES

[1] H. S. Esfahan, "Graph Algorithms (Parts 1 and 3)," Design and Analysis of Algorithms, Georgia State University, Atlanta, GA, USA, PowerPoint slides, Oct. 6 and Oct. 27, 2025.

[2] L. R. Ford and D. R. Fulkerson, “A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem,” *Canadian Journal of Mathematics*, vol. 9, pp. 210–218, 1957. doi:10.4153/CJM-1957-024-0

[3] U. Zwick, “The smallest networks on which the Ford-Fulkerson maximum flow procedure may fail to terminate,” *Theoretical Computer Science*, vol. 148, no. 1, pp. 165–170, Aug. 1995, doi:10.1016/0304-3975(95)00022-O.

[4] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vancouver, BC, Canada: IEEE Comput. Soc, 2001, pp. 105–112. doi:10.1109/ICCV.2001.937505.

[5] Y. Y. Boykov and G. Funka-Lea, “Graph Cuts and Efficient N-D Image Segmentation,” *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, Nov. 2006, doi:10.1007/s11263-006-7934-5.

[6] P. Veličković, “Everything is connected: Graph neural networks,” *Current Opinion in Structural Biology*, vol. 79, p. 102538, Apr. 2023, doi:10.1016/j.sbi.2023.102538.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.

THANK YOU

Joshua Piña

CSC 4520: Design & Analysis of Algorithms

Data Science, CS Department, GSU

[Everything is a Graph Repository]