Name: Harsh Vijay Mamania

CS 5330: Week 6 Homework

Question #1

Why is connectedness important in binary image processing? Given an example as to how it might affect something like segmentation/connected components analysis or morphological filtering.

Solution-

- Connectedness defines which pixels belong to the same region/object. Without it, you can't label, count, or analyze objects meaningfully.

- The two choices: 4-connectivity (N/S/E/W) vs 8-connectivity (includes diagonals) directly change your results.

Example #1 (Connected Components):

1 0 0
0 1 0
0 0 1

- 4-connectivity --> 3 separate objects (no shared cardinal neighbors)
- 8-connectivity --> 1 object (diagonal pixels touch)

Same image, completely different segmentation output.

Example #2 (Morphological Filtering):

- A flood fill using 8-connectivity can "leak" through a diagonal gap that 4-connectivity would treat as a sealed boundary, causing hole-filling or region-growing to behave incorrectly.

Question #2

What is the purpose of the morphological operators (A) growing/dilation, and (B) shrinking/erosion?

Solution-

A) Dilation (Growing)

- Expands the foreground (white) regions by adding pixels around the boundary of each object.
- Purpose: fills small holes, bridges narrow gaps between nearby objects, and thickens thin structures.

B) Erosion (Shrinking)

- Contracts the foreground regions by removing pixels from the boundary of each object.
- Purpose: removes small noise/speckles, separates weakly connected objects, and thins structures.

Question #3

What is the purpose of the grassfire transform algorithm? Describe the inputs and outputs.

Solution-

Purpose of the Grassfire Transform

- Computes the distance of each foreground pixel to the nearest background pixel
- Produces a distance map of the binary image

Inputs

- A binary image
    - Foreground pixels represent objects
    - Background pixels represent empty space
- A chosen connectivity rule such as 4-connected or 8-connected

Outputs

- A distance map (same size as input) where each foreground pixel holds its distance to the nearest background pixel.
- Background pixels are assigned 0.
- Foreground pixels farther from the boundary get progressively higher values, like a "hill" rising from the edges inward.

Question #4

When is it more efficient to grow or shrink using the grassfire transform compared to using a standard morphological filter?

Solution-

Grassfire is more efficient than standard morphological filtering when:

1. Large dilation or erosion is needed
    a. Standard morphology requires many repeated operations
    b. Grassfire computes the distance map once
    c. A simple threshold gives any desired amount of growth or shrinkage

2. Multiple radii are required
    a. Standard morphology must be repeated for each size
    b. Grassfire allows multiple thresholds from the same distance map

Standard Morphology Is Better when:

1. Small, single dilation or erosion
2. Simple, one-time operations

Grassfire is more efficient for large or repeated growing and shrinking operations because the distance transform is computed once and reused.

Question #5

The region growing and 2-pass segmentation algorithms each require a data structure. What data structure is required by each algorithm?

Solution-

Region Growing - Queue (or Stack)

- Uses a queue (BFS) or stack (DFS) to track pixels to be visited and expanded from a seed point.

2-Pass Algorithm - Union-Find (Disjoint Set)

- First pass assigns provisional labels and records equivalences; a Union-Find structure efficiently merges equivalent labels.
- Second pass replaces all labels with their canonical root from the Union-Find.

Question #6

How is building an embedding space using PCA different than building one using a deep network like ResNet18? Your answer can discuss process or quality.

Solution-

Process Differences:

- PCA finds a new coordinate system by computing eigenvectors of the data's covariance matrix, projecting pixels/features onto the top-k eigenvectors. It is entirely unsupervised and linear.
- ResNet18 learns features through supervised training on millions of labeled images, using nonlinear transformations (convolutions, ReLU, pooling) stacked across many layers.

Quality Differences:

- PCA captures directions of maximum variance in raw pixel space, which doesn't necessarily correspond to semantically meaningful differences. It struggles with lighting, pose, and appearance variation.
- ResNet18 embeddings capture high-level semantic features (edges, textures, object parts) learned from data, making them far more discriminative and robust to real-world variation.

My final thoughts:

- PCA is fast, interpretable, and requires no labeled data, but produces weaker embeddings.
- ResNet18 embeddings are much more powerful but require a pretrained model and are essentially a black box.
- PCA works best when data variation is mostly linear; deep embeddings handle complex, nonlinear structure in appearance much better.