KENNESAW STATE
UNIVERSITY

CS 7367
MACHINE VISION

ASSIGNMENT 3

MORPHOLOGICAL FILTERS AND SPATIAL FILTERING

<u>INSTRUCTOR</u>

Dr. Sanghoon Lee

Your Name: Adam Kangiser
KSU ID: 000681701

1. ABSTRACT

In this research we present an approach to analyzing images using MATLAB and various image processing techniques. Our focus is on three applications; general morphological operations, fingerprint analysis and identification and counting of cells.

In the part we discuss the use of morphological operators and filters on an image called "morphology.png". We highlight how MATLABs built in functions can be utilized to manipulate and improve the quality of the image although further exploration is needed to delve into the specifics.

Moving forward we delve into fingerprint analysis by employing both median filters to enhance fingerprint images. Our goal is to determine the circumstances under which each filter produces results. The morphological filter employs opening and closing operations to emphasize preserving the details found in fingerprint patterns. On the other hand the median filter effectively reduces "salt and pepper" noise by utilizing a 3x3 neighborhood for filtering resulting in a representation.

Lastly we take an approach, towards counting and analyzing biological cells in a sample image. By using thresholding, morphological operations and connected component analysis our algorithm can identify cells while estimating their pixel area. Additionally it demarcates the boundary of the cell in the output image.

One significant improvement includes setting a size limit to prevent inaccuracies caused by background noise and tiny objects resulting in an accurate count of cells.

2. TEST RESULTS

Figure 1: Original morphology.png

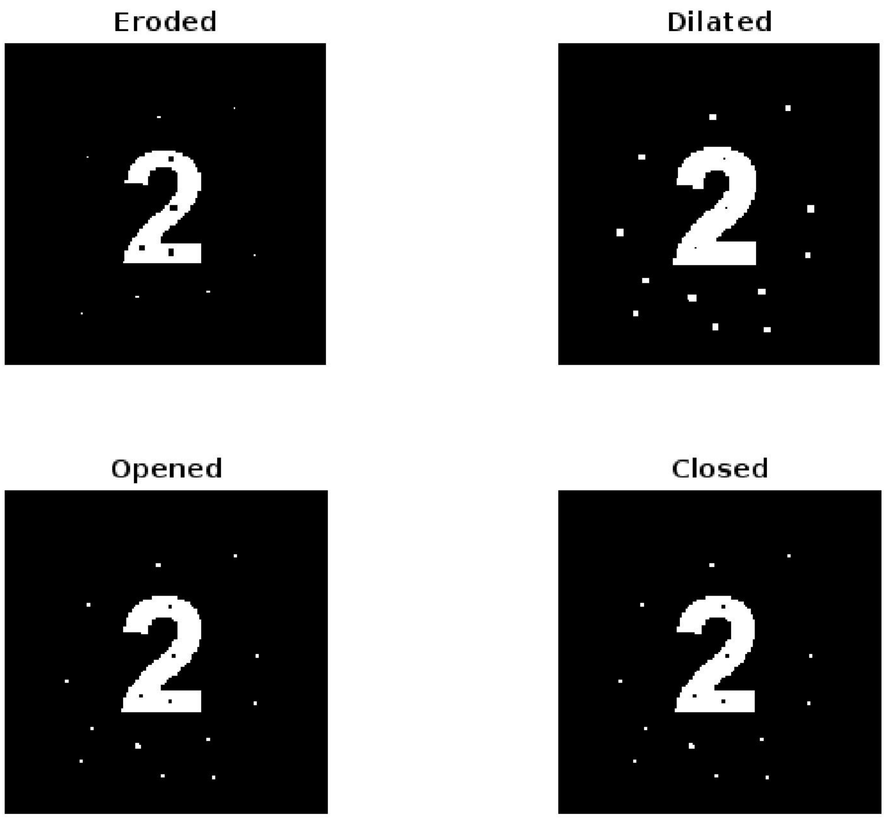

Figure 2: Output images showing each filter.

Figure 3: Fingerprint original image with morphological filter and median filter.

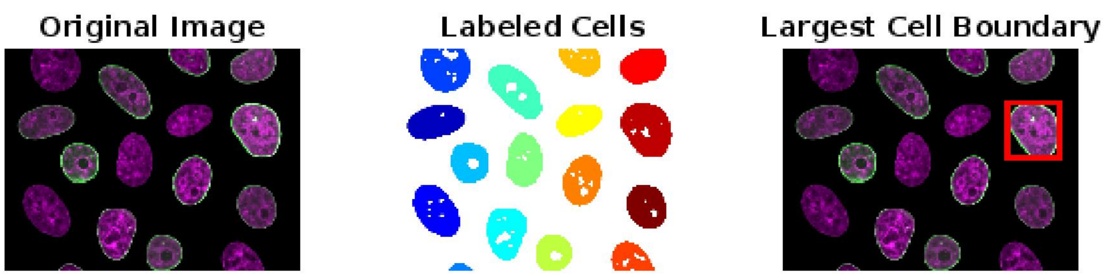

Figure 4: Original cell.jpg image and then the labeled cells image and then the rightmost image showing the largest cell.

Output from cell.jpg:

```
  count_cells
Total number of cells:
16

Size of each cell in pixels:
6684
7511
7055
721
5077
6623
7566
6646
4483
4946
3698
6162
3971
5627
9045
```

3. Discussion

Throughout our research project we delved into the world of image analysis. We used MATLABs collection of built in functions to uncover patterns, in various datasets, including abstract morphology illustrations, fingerprints and biological cells. By working with these sets of data we gained valuable insights into the versatility of morphological filters and their wide range of applications.

In the phase of our project we deepened our understanding of operations establishing a strong foundation for subsequent analyses. The examination of fingerprints underscored the role that finely tuned filtration plays in preserving the details of patterns. This discovery opened up possibilities for improved applications. Additionally our meticulous approach to cell counting demonstrated how precision and technology can come together seamlessly.

One important lesson we learned was the necessity of employing a thresholding strategy to accurately recognize and categorize elements within a heterogeneous image space. It's about finding the balance between sensitivity and specificity. Moreover our project highlighted the importance of optimizing parameters to avoid miscounts and misclassifications—a path that holds potential for future integration, with machine learning techniques.

With time further research could explore the realm of automated parameter tuning. By utilizing AI driven methods we can not. Also learn from and adjust to the distinct qualities of various image types. This paves the way for independent image analysis endeavors. This experience has provided a foundation for investigation nurturing a vision of a future where technology merges seamlessly with precision, in image analysis uncovering hidden patterns and narratives that are easily overlooked.

4. CODES

4.1 Code for morphology.png

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 3
img = imread('morphology.png');
if size(img, 3) == 3
img_gray = rgb2gray(img);
else
img_gray = img;
end
se = strel('square', 3);
img_eroded = imerode(img_gray, se);
img_dilated = imdilate(img_gray, se);
img_opened = imopen(img_gray, se);
img_closed = imclose(img_gray, se);
figure,
subplot(2,2,1), imshow(img_eroded), title('Eroded');
subplot(2,2,2), imshow(img_dilated), title('Dilated');
subplot(2,2,3), imshow(img_opened), title('Opened');
subplot(2,2,4), imshow(img_closed), title('Closed');
imwrite(img_eroded, 'eroded.png');
imwrite(img_dilated, 'dilated.png');
imwrite(img_opened, 'opened.png');
imwrite(img_closed, 'closed.png');
```

•••

4.2 Code for fingerprint_BW.png

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 3
img = imread('fingerprint_BW.png');
if size(img, 3) == 3
img_gray = rgb2gray(img);
else
img_gray = img;
end
se = strel('square', 3);
morph_filtered = imopen(img_gray, se);
morph_filtered = imclose(morph_filtered, se);
med_filtered = medfilt2(img_gray, [3 3]);
figure,
subplot(1,3,1), imshow(img_gray), title('Original');
subplot(1,3,2), imshow(morph_filtered), title('Morphological Filtered');
subplot(1,3,3), imshow(med_filtered), title('Median Filtered');
imwrite(morph_filtered, 'morph_filtered.png');
imwrite(med_filtered, 'med_filtered.png');
```

•••

4.3 Code for cell.jpg

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 3
img = imread('cell.jpg');
img_gray = rgb2gray(img);
bw = imbinarize(img_gray, 'adaptive', 'ForegroundPolarity','dark', 'Sensitivity',
0.4);
se = strel('disk', 2);
bw = imopen(bw, se); % Opening to remove noise
bw = imclose(bw, se); % Closing to fill gaps in the cells
cc = bwconncomp(bw);
stats = regionprops(cc, 'Area', 'BoundingBox');
minCellSize = 100; % Define a minimum cell size (in pixels)
idx = find([stats.Area] > minCellSize);
numCells = numel(idx);
cellSizes = [stats(idx).Area];
[~, idxLargest] = max(cellSizes);
figure,
subplot(1,3,1), imshow(img), title('Original Image');
subplot(1,3,2), imshow(label2rgb(labelmatrix(cc))), title('Labeled Cells');
subplot(1,3,3), imshow(img), title('Largest Cell Boundary');
hold on
rectangle('Position', stats(idxLargest).BoundingBox, 'EdgeColor', 'r', 'LineWidth',
2);
hold off
disp('Total number of cells:'), disp(numCells)
disp('Size of each cell in pixels:'), disp(cellSizes')
```