ECE 6310 – Introduction to Computer Vision – LAB 3 REPORT

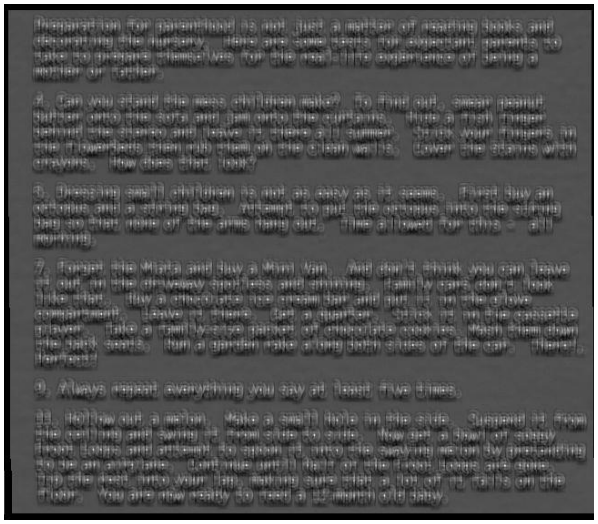
Letters

TASKS

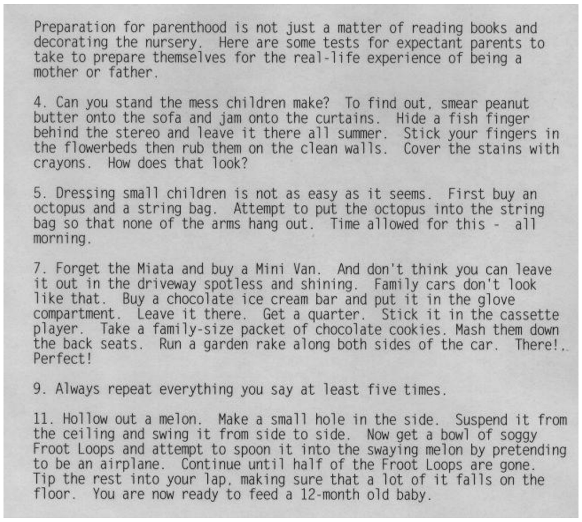
- Use MSF image from Lab 2 to threshold at T and "detect" letters
- If detected, use original image to check 9x15 area around detected pixel location
- Thin the 9x15 image
- Look for branchpoints and endpoints
- If #branchpoints = 1 and #endpoints = 1, mark as detected
- Calculate FP, TP, FN, TN, FPR and TPR
- Generate ROC curve

INPUTS

MSF Image



Input Image



Preparation for parenthood is not just a matter of reading books and decorating the nursery. Here are some tests for expectant parents to take to prepare themselves for the real-life experience of being a mother or father.

4. Can you stand the mess children make? To find out, smear peanut butter onto the sofa and jam onto the curtains. Hide a fish finger behind the stereo and leave it there all summer. Stick your fingers in the flowerbeds then rub them on the clean walls. Cover the stains with crayons. How does that look?

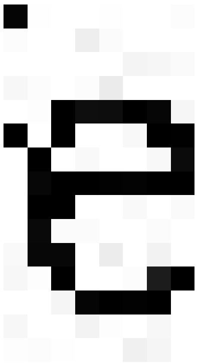
5. Dressing small children is not as easy as it seems. First buy an octopus and a string bag. Attempt to put the octopus into the string bag so that none of the arms hang out. Time allowed for this - all morning.

7. Forget the Miata and buy a Mini Van. And don't think you can leave it out in the driveway spotless and shining. Family cars don't look like that. Buy a chocolate ice cream bar and put it in the glove compartment. Leave it there. Get a quarter. Stick it in the cassette player. Take a family-size packet of chocolate cookies. Mash them down the back seats. Run a garden rake along both sides of the car. There!. Perfect!

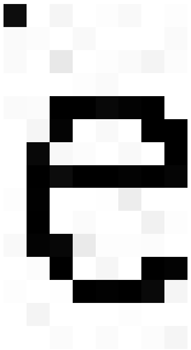
9. Always repeat everything you say at least five times.

11. Hollow out a melon. Make a small hole in the side. Suspend it from the ceiling and swing it from side to side. Now get a bowl of soggy Froot Loops and attempt to spoon it into the swaying melon by pretending to be an airplane. Continue until half of the Froot Loops are gone. Tip the rest into your lap, making sure that a lot of it falls on the floor. You are now ready to feed a 12-month old baby.

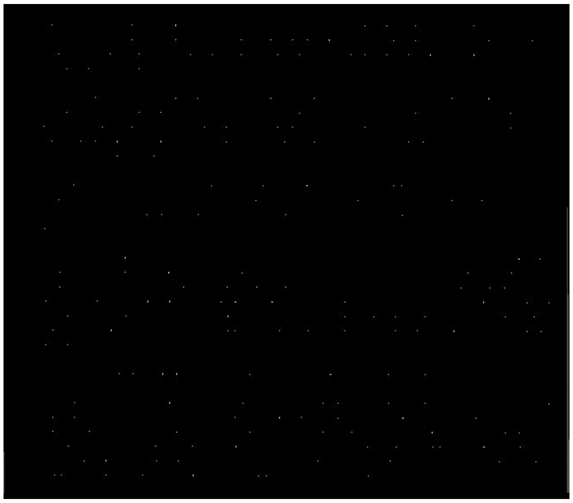
Copy Image



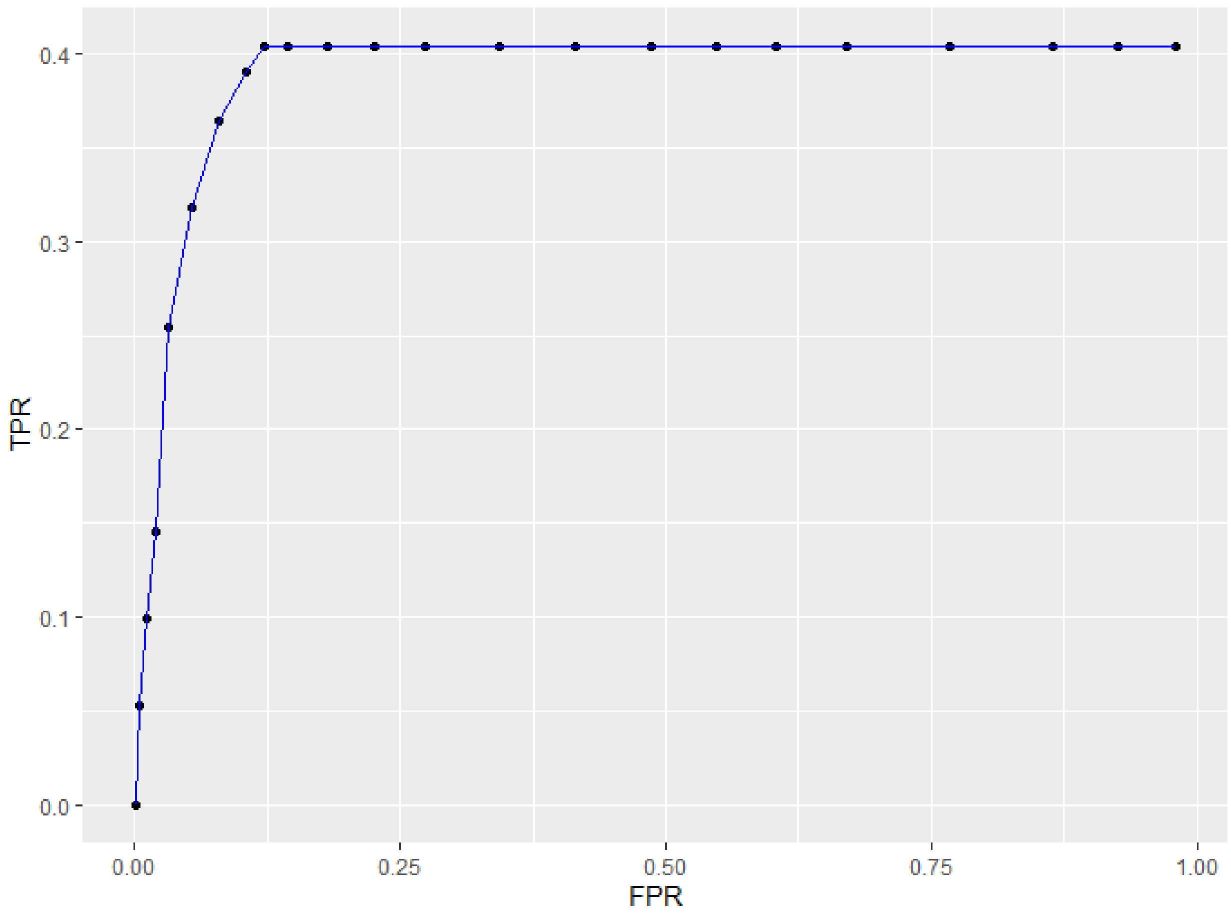
Thin Image



Binary Image (Th = 128)

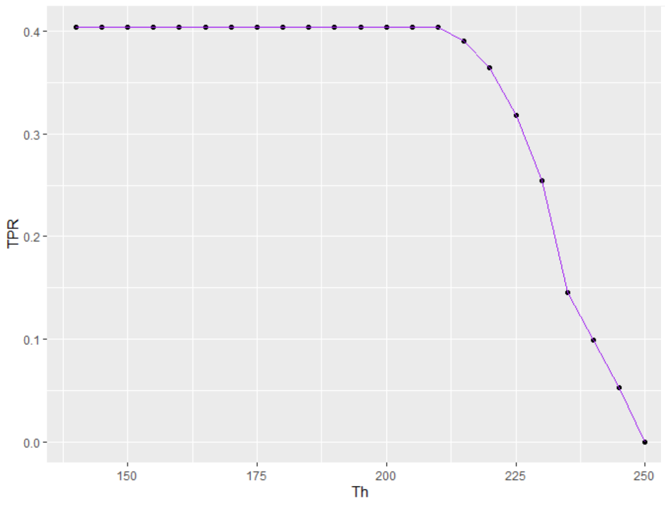


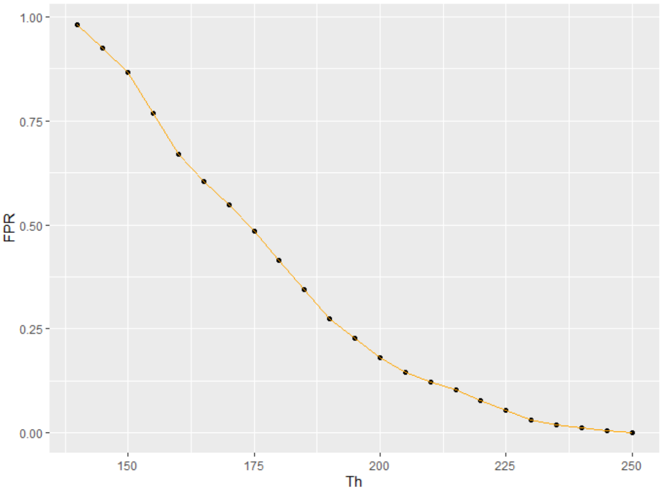
ROC Curve



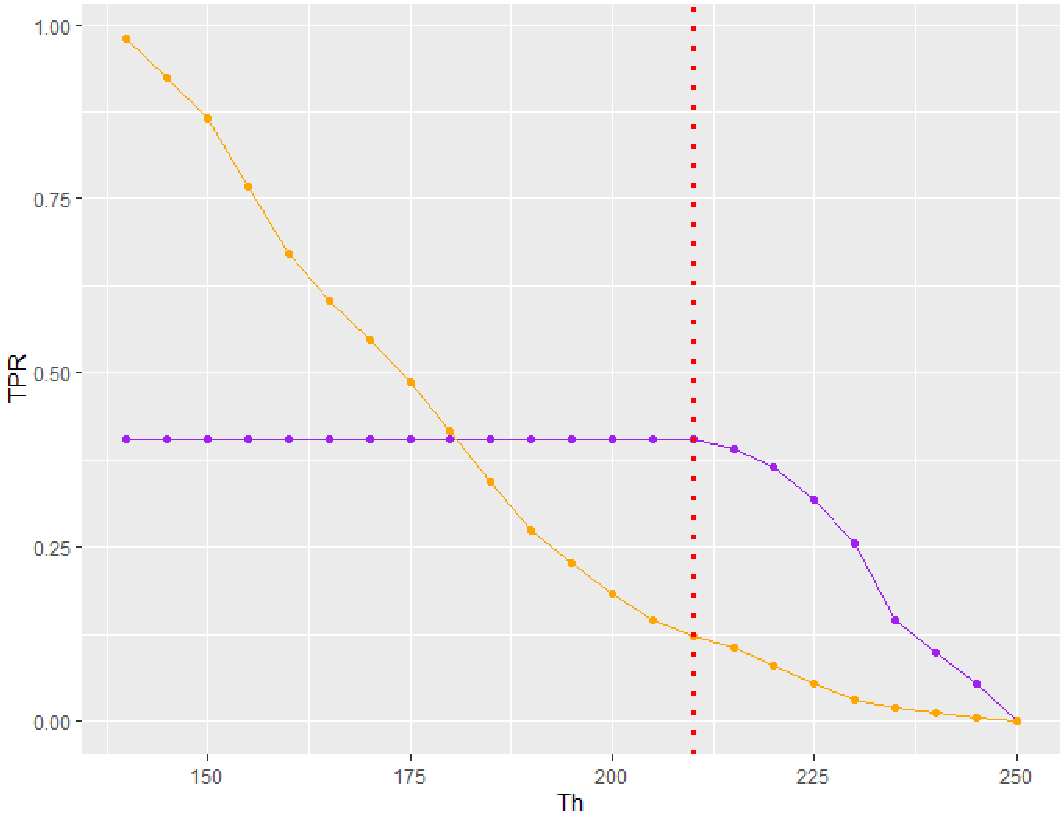
TPR vs Threshold

FPR vs. Threshold





Superimposing the two plots



T = 210

Optimal

Threshold

DISCUSSION AND ANALYSIS

Choosing the optimal threshold value

As seen above, the maximum TPR for this implementation turned out to be 0.404. To obtain the optimal threshold for this particular implementation, we look at the last threshold value at which TPR maintains at 0.404, which is at T = 210. After that, TPR starts to decline.

The following table summarizes the ROC curve

THRESHOLD	BEHAVIOR
T = 140	FPR is almost 1, TPR is at 0.404
T = 145	FPR is decreasing, TPR maintains
T = 210	Highest Threshold value at which TPR remains 0.404 – optimal threshold
T = 215	TPR decreases, FPR continues to decrease
T = 250	TPR = 0, FPR = 0.009, almost zero

Discussion of Implementing the Thinning Algorithm in C

- Code from previous lab was used to build upon.
- Initially, the entire thinning algorithm was implemented inside a while(1) loop, which caused unforeseen errors. These errors were not able to be rectified.

- Then, for each test in the thinning algorithm, a sub-routine was written before main(), providing cleaner code that was easier to debug. Border conditions were handled separately. Segmentation faults were removed.
- The tests, however, did not produce the best thinned image.
- The number of endpoints being counted for each detected letter was causing the implemented code to produce a maximum of 61 detected "e." This difficulty was encountered but not rectified.
- This report, therefore, only shows the "best result" from the maximum possible TPR produced.

Conclusion

The thinning algorithm was used to "erase" pixels, find endpoints and branchpoints, and consequently have better chance at detecting an "e," compared to the previous lab, which simply checked for the threshold value in the MSF image. If this implementation was done correctly, the TPR should have performed better and provide better results compared to the previous lab. However, since it was not fully functioning, the improvement in results was not verified or observed. This lab has helped me gain better understanding of the importance of thinning an image, the implementation of sub-routines, debugging methods, as well as drawing the best results out of what has been implemented.