UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS791 Assignment 2

Image Analysis and Understanding: Multilevel Thresholding

Student:

Arno Jooste - u21457451

Lecturer:

Dr. Thambo Nyathi

October 2024

https://github.com/08Arno30/COS791-A2

1 Introduction

This assignment examined two optimisation strategies, Simulated Annealing (SA) and Variable Neighbourhood Search (VNS), for addressing the multilevel thresholding problem for image segmentation. Multilevel thresholding divides a picture into several zones based on pixel intensity levels, which is an important task in many image processing applications. This assignment prioritised thresholding at levels (k = 2, 3, 4, 5) for a set of five images.

The Otsu method and Kapur's entropy are well-known objective functions for evaluating thresholding quality. These methods was used to identify the best thresholds for segmenting pictures. The purpose was to assess how well the SA and VNS algorithms perform when identifying appropriate thresholds across different levels and images.

In addition, a table is provided to describe the threshold values and key quality metrics collected from both techniques. Additionally, for each image, at least one pair of Otsu and Kapur method findings is visualised to further explain the segmentation results.

2 Experimental Setup

This section discusses the configuration for the experimental setup.

2.1 Parameters

Hyperparameter	Value
Seed	99
Objective Function	Otsu or Kapur
Algorithm	SA or VNS
K (Levels)	2, 3, 4, or 5
Number of iterations	1000
Step size	5
Initial Temperature	100

Table 1: Values for parameters

2.2 Objective Functions

2.2.1 Otsu Method

The Otsu method maximises the variance between classes. The *perform_otsu* method seeks to identify the ideal threshold values that distinguish these classes. The method begins by randomly assigning a set of threshold values. These thresholds define various parts of the image. The cumulative histogram is used to calculate the weight of each region, which represents the probability that pixels will belong to that region. The mean intensity of every region is then determined, and its variance contribution is calculated using its weight and mean. This method then aims to maximise the weighted sum of squared means for each region, which significantly reduces the variance within each class. By minimising the variance within each class, the method tries to make pixel values in various regions as different as possible, resulting in more successful segmentation.

2.2.2 Kapur's Entropy Method

Kapur is an entropy-based thresholding approach. This approach seeks to maximise the entropy of segmented regions. By maximising the entropy, it can determine thresholds that result in regions with the most diverse pixel values. The *perform_kapur* method starts by randomly initialising threshold values to partition the image into regions. The cumulative histogram is then used to determine the pixel distribution for each region. These histograms are normalised such that they accurately represent probability distributions. Finally, the entropy is determined for each region based on the normalised histogram data., where the goal is to maximise the total of entropies in all regions. A higher entropy value suggests greater uncertainty and a broader range of pixel intensities, which is commonly desirable for effective segmentation.

2.3 Algorithms

2.3.1 Simulated Annealing (SA)

The Simulated Annealing (SA) method applies a probabilistic approach to find optimal threshold values for image segmentation. The *perform_sa* function begins with a set of random thresholds and analyses them using either Otsu's variance-based or Kapur's entropy-based methods, depending on the parameters passed in. At each iteration, a new solution (neighbour) is created by varying the present thresholds within the predefined boundaries of image intensity levels (lower bound is 0, while the upper bound is 255). The new solution (neighbour) is accepted if it improves the evaluation (variance or entropy) or is based based on a probability determined by the temperature, which decreases over time to reduce randomness. As the temperature drops, the algorithm focusses more on better solutions and eventually converges on the near-optimal threshold values.

2.3.2 Variable Neighbourhood Search (VNS)

The Variable Neighbourhood Search (VNS) method improves image segmentation by carefully exploring and updating threshold values through local search within different neighbourhoods. Unlike the Simulated Annealing (SA) approach, VNS aims to expand the search by modifying the neighbourhood structure when a local optimum is found. The *perform_vns* function begins with an initial set of thresholds and iteratively perturbs them by adding random variations within a specific step size, ensuring that the new thresholds stay within the given bounds (lower bound is 0 and upper bound is 255 for image intensity levels). This approach improves the thresholds by picking candidate thresholds with better evaluations (by Otsu's or Kapur's method).

2.4 Stopping Criteria

The following stopping criteria were used:

- **Number of iterations:** The algorithm (VNS or SA) can only run for a number of generations to ensure exploration of the search space.
- **Fitness Stagnation:** The program terminates if the the variance or entropy (depending on the objective function) has not improved for a certain number of iterations. In this assignment the value *10* was used as the maximum number of iterations the best variance or entropy can remain unchanged.

3 Results

This section only shows images of the best performance. The other performances are reported in each table.

Image T22 Results:

Image	Level		Otsu		Kapur	
			SA	VNS	SA	VNS
T22	2	th	50, 235	95, 168	3, 213	35, 111
		values	217776364.92	212137118.41	5.06	4.87
		SSIM	61.31%	52.26%	40.52%	62.90%
		PSNR	19.13	17.68	12.97	13.55
	3	th	28, 107, 241	42, 217, 231	4, 85, 239	110, 145, 196
		values	239122109.53	235036409.92	8.84	8.42
		SSIM	72.49%	64.55%	60.52%	46.96%
		PSNR	20.54	20.08	16.22	15.99
	4	th	12, 58, 140, 252	32, 32, 50, 107	4, 68, 197, 241	73, 74, 158, 203
		values	243929487.50	246254827.28	12.18	11.53
		SSIM	74.58%	61.17%	65.81%	59.88%
		PSNR	19.83	11.09	20.05	19.74
	5	th	11, 59, 83, 128, 238	22, 28, 33, 130, 250	5, 71, 115, 175, 252	27, 27, 37, 126, 252
		values	249818122.74	244658079.83	15.71	14.97
		SSIM	79.53%	63.50%	76.51%	64.23%
		PSNR	18.92	13.75	23.74	13.97

Table 2: Best Results for Image T22

Abbreviations: *th = Thresholds, values = Objective Function values (variance for Otsu, and total entropy for Kapur)*

Note: *Refer to the tabel for the threshold values (red vertical lines)*

Discussion: We clearly note that Simulated Annealing (SA) consistently outperforms Variable Neighbourhood Search (VNS) in terms of SSIM, by at least 7%. This shows that SA's ability to explore more of the solution space while avoiding local optima contributes to greater structural detail preservation than VNS, which has a more limited exploration capability. While PSNR values remain stable across thresholds, demonstrating that both approaches maintain image fidelity, SSIM's fluctuation demonstrates SA's advantage in capturing visual quality. Finally, the Otsu method with SA has the best SSIM score, illustrating the effectiveness of integrating statistical thresholding with SA's global search.

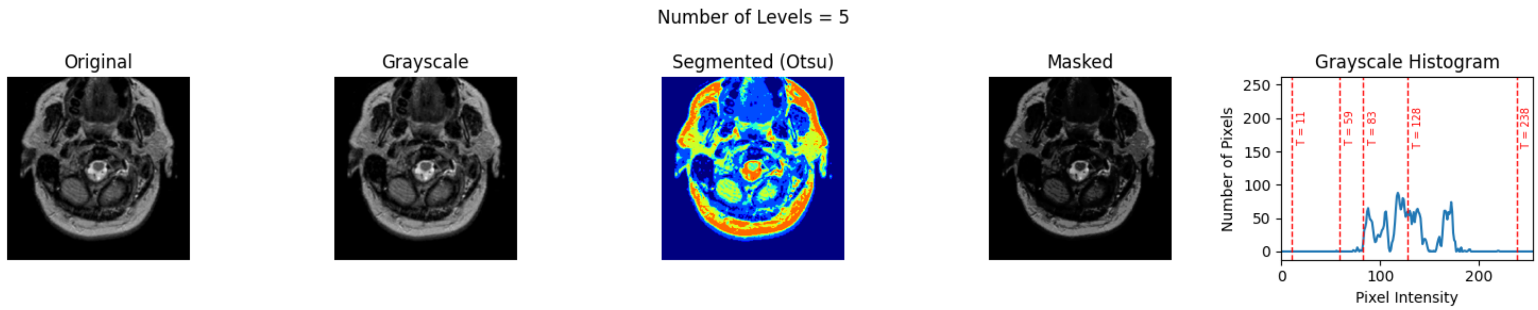

Figure 1: Best result for T22: Otsu method with SA

Image T32 Results:

Discussion: Similar to the first result's discussion, here Simulated Annealing (SA) also consistently outperformed Variable Neighbourhood Search (VNS) in terms of SSIM. But here it was by at least 5%. This yet again shows that SA has the ability to explore more of the solution space while avoiding local optima. This contributes to greater structural detail preservation than VNS, which has a more limited exploration capability. Here the PSNR values remain stable across thresholds, demonstrating that both approaches maintain image fidelity, however SSIM's fluctuation yet again demonstrates its advantage in capturing perceptual quality. To conclude this discussion, we note that the Otsu method with SA has the best SSIM score.

Image	**Level**		**Otsu**		**Kapur**	
			SA	**VNS**	**SA**	**VNS**
T32	2	th	53, 248	182, 240	6, 240	121, 180
		values	254803704.83	254378401.94	5.11	5.03
		SSIM	64.24%	35.90%	45.56%	47.70%
		PSNR	19.60	11.82	14.15	14.93
	3	th	23, 99, 234	0, 115, 133	32, 175, 252	118, 141, 207
		values	272250662.86	259162525.97	8.88	8.75
		SSIM	73.47%	23.85%	66.14%	47.97%
		PSNR	19.62	11.11	19.93	15.01
	4	th	24, 96, 114, 221	58, 79, 127, 252	13, 55, 182, 254	33, 71, 83, 193
		values	275137411.817	272943084.87	12.45	11.28
		SSIM	78.15%	71.06%	71.04%	71.71%
		PSNR	21.07	20.65	20.06	16.18
	5	th	11, 63, 120, 126, 248	79, 81, 140, 164, 223	33, 11, 169, 218, 252	59, 88, 157, , 177, 250
		values	284224115.85	274387747.19	15.23	14.91
		SSIM	79.53%	64.88%	76.15%	70.77%
		PSNR	18.92	20.36	20.88	21.76

Table 3: Best Results for Image T32

Abbreviations: *th = Thresholds, values = Objective Function values (variance for Otsu, and total entropy for Kapur)*

Note: *Refer to the tabel for the threshold values (red vertical lines)*

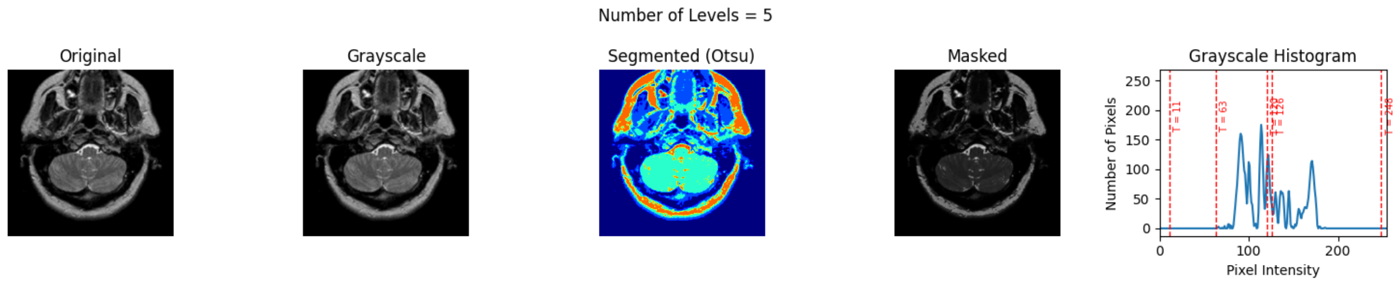

Figure 2: Best result for T32: Otsu method with SA

Image T42 Results:

Discussion: The next table shows different results than the prior discussions. Here Simulated Annealing (SA) and Variable Neighbourhood Search (VNS) shared dominance over one another, in terms of SSIM, in various occurrences. The Otsu method's SA SSIM scores were mostly dominant over its VNS SSIM scores by at least 5%. In contrast to this, the Kapur method's VNS SSIM scores mostly outperformed its SA SSIM scores. This shows that when SA is used with the Otsu method, it has the ability to explore more of the solution space while avoiding local optima, but not with the Kapur method. The Kapur method's VNS shows greater ability to explore the solution space than SA. To conclude this discussion, we note that the Kapur method with SA achieved the best SSIM score (82.75%).

Image	Level		Otsu		Kapur	
			SA	VNS	SA	VNS
T42	2	th	63, 240	127, 195	6, 237	8, 118
		values	307282336.79	273474714.93	5.19	5.19
		SSIM	64.14%	46.39%	44.83%	58.38%
		PSNR	19.22	14.06	14.37	12.63
	3	th	38, 96, 227	113, 210, 231	1, 137, 247	33, 105, 150
		values	328241792.40	322455460	9.13	8.61
		SSIM	73.67%	47.62%	51.67%	75.78%
		PSNR	19.93	14.05	17.22	18.56
	4	th	48, 102, 187, 244	21, 148, 166, 221	0, 108, 174, 228	41, 91, 160, 247
		values	338065542.60	326105687.27	12.60	12.16
		SSIM	75.64%	70.47%	33.23%	77.86%
		PSNR	23.09	19.75	14.72	24.45
	5	th	16, 75, 140, 168, 254	116, 165, 200, 240, 241	16, 91, 140, 168, 254	54, 88, 100, 183, 226
		values	347138282.97	340057924.41	15.57	15.43
		SSIM	79.53%	43.78%	82.75%	72.53%
		PSNR	18.92	13.25	24.73	21.96

Table 4: Best Results for Image T42

Abbreviations: *th = Thresholds, values = Objective Function values (variance for Otsu, and total entropy for Kapur)*

Note: *Refer to the tabel for the threshold values (red vertical lines)*

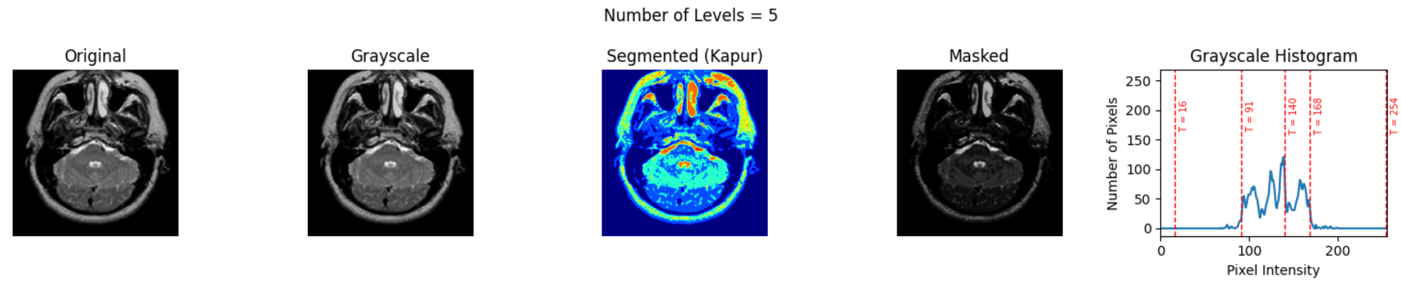

Figure 3: Best result for T42: Kapur method with SA

Image T52 Results:

Discussion: Similar to the first two results' discussion, here Simulated Annealing (SA) also outperformed Variable Neighbourhood Search (VNS) in terms of SSIM, for most cases. Some cases with the use of the Kapur method showed that the SA did not outperform the VNS approach. This shows that SA has the ability to explore more of the solution space while avoiding local optima, but in other cases when used with the Kapur method, VNS has this ability. Hence, it depends on the combination of the methods used to prove which method (SA or VNS) contributes to greater structural detail preservation. Here the PSNR values again remain somewhat stable across thresholds, demonstrating that both approaches maintain image quality. To conclude this discussion, we note that the Otsu method with SA has yet again the best SSIM score (84.20%).

Image	**Level**		**Otsu**		**Kapur**	
			SA	**VNS**	**SA**	**VNS**
T52	2	th	47, 250	33, 142	11, 232	30, 137
		values	328634519.07	328176052.54	5.04	5.03
		SSIM	61.90%	66.11%	48.64%	65.89%
		PSNR	18.82	16.24	15.72	15.65
	3	th	34, 101, 229	10, 187, 227	7, 102, 233	38, 89, 184
		values	47274371.33	349179150.21	8.80	8.52
		SSIM	76.13%	52.95%	69.64%	72.62%
		PSNR	20.18	18.58	18.55	17.99
	4	th	26, 83, 127, 241	23, 34, 41, 149	4, 64, 133, 238	17, 50, 50, 148
		values	355550013.87	349256561.94	12.33	11.86
		SSIM	81.15%	61.36%	72.17%	64.39%
		PSNR	21.67	11.32	19.08	11.80
	5	th	28, 67, 115, 147, 249	3, 35, 44, 52, 160	0, 44, 102, 143, 252	80, 90, 104, 118, 128
		values	359318294.67	357642910.86	15.23	14.58
		SSIM	84.20%	55.71%	48.26%	61.82%
		PSNR	23.62	10.64	15.86	14.74

Table 5: Best Results for Image T52

Abbreviations: *th = Thresholds, values = Objective Function values (variance for Otsu, and total entropy for Kapur)*

Note: *Refer to the tabel for the threshold values (red vertical lines)*

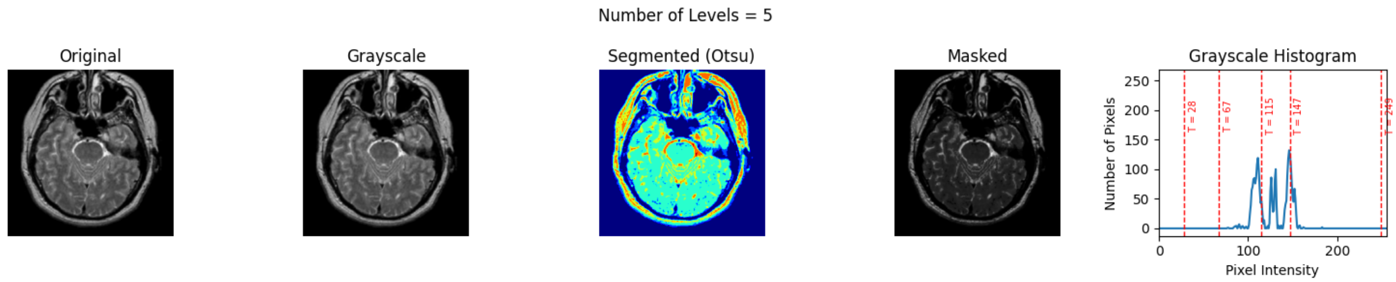

Figure 4: Best result for T52: Otsu method with SA

Image T62 Results:

Discussion: The next table shows a second occurrence where the Kapur method with SA achieved the best SSIM Score. Here Simulated Annealing (SA) and Variable Neighbourhood Search (VNS) shared dominance over one another, in terms of SSIM, in various occurrences. Some combinations showed very poor SSIM scores (<30%) (resulting in poor PSNR scores), which proves those combinations may have got stuck at local optima and might have been very limited do exploring the solution space. The Otsu method's SA SSIM scores were mostly dominant over its VNS SSIM scores by at least 12% (if it outperformed the VNS method). In contrast to this, the Kapur method's VNS SSIM scores mostly outperformed its SA SSIM scores (except for the scenario of using 5 thresholds (levels)). This again shows that when SA is used with the Otsu method, it has the ability to explore more of the solution space while avoiding local optima, but not with the Kapur method. The Kapur method's VNS shows greater ability to explore the solution space than SA. To conclude this discussion, we note that the Kapur method with SA achieved the best SSIM score (82.03%).

Image	**Level**		**Otsu**		**Kapur**	
			SA	**VNS**	**SA**	**VNS**
T62	2	th	62, 251	2, 211	7, 250	46, 77
		values	386573996.83	367951105.21	5.16	5.16
		SSIM	61.55%	41.68%	46.51%	54.97%
		PSNR	18.72	14.66	15.41	8.98
	3	th	31, 97, 244	0, 137, 229	0, 152, 251	108, 188, 250
		values	411036157.58	402221062.41	9.13	8.79
		SSIM	73.72%	29.18%	24.30%	52.07%
		PSNR	19.40	12.66	12.52	13.83
	4	th	37, 93, 127, 237	12, 48, 73, 160	3, 53, 173, 240	21, 92, 243, 244
		values	418235012.23	403686949.74	12.58	12.04
		SSIM	80.20%	68.68%	62.88%	76.15%
		PSNR	22.17	12.84	18.92	21.95
	5	th	15, 39, 128, 135, 253	46, 54, 137, 189, 204	8, 59, 95, 158, 241	30, 84, 159, 199, 218
		values	421538761.43	423999407.35	15.67	14.78
		SSIM	76.10%	76.12%	82.03%	81.07%
		PSNR	20.76	23.54	20.70	23.53

Table 6: Best Results for Image T62

Abbreviations: *th = Thresholds, values = Objective Function values (variance for Otsu, and total entropy for Kapur)*

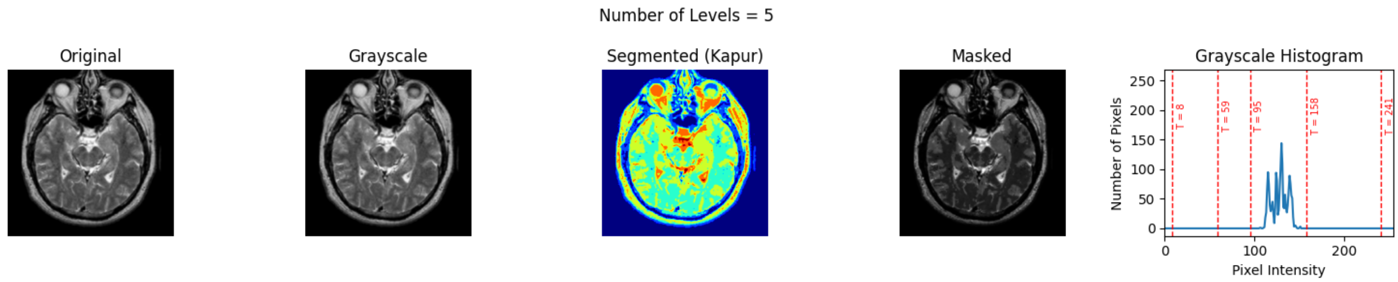

Figure 5: Best result for T62: Kapur method with SA

Note: *Refer to the tabel for the threshold values (red vertical lines)*

4 Conclusion

To conclude, the combination of the Otsu method with Simulated Annealing (SA) consistently dominated the other approaches in terms of structural similarity and overall segmentation quality. The adaptive nature of SA, which allows it to explore a larger solution space while avoiding local optima, contributed to its greater performance when combined with Otsu's variance-based thresholding. Furthermore, increasing the number of thresholds or levels often resulted in higher metric scores since finer segmentation allows for more accurate representation of image characteristics. The multithresholding approach with five thresholds produced the greatest scores, implying that a higher number of thresholds allows for better detection of image regions with intensity changes, resulting in more effective image segmentation and better overall results.