

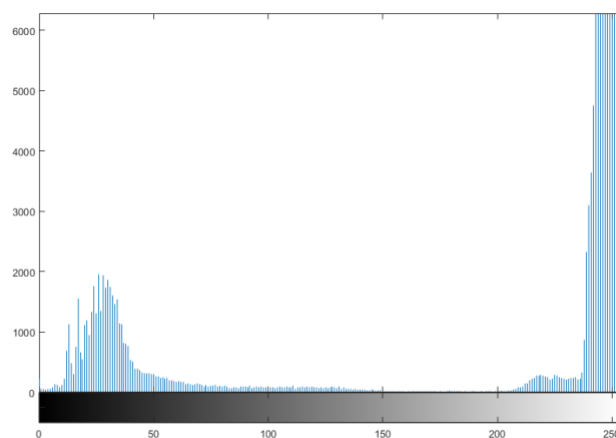
Homework#1 Image Segmentation

1. Threshold-Based Segmentation

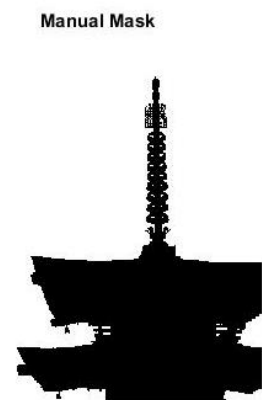
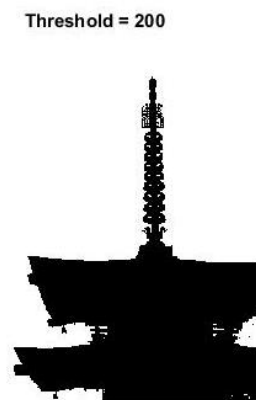
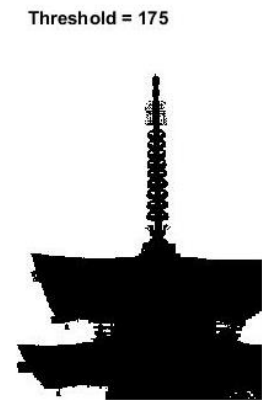
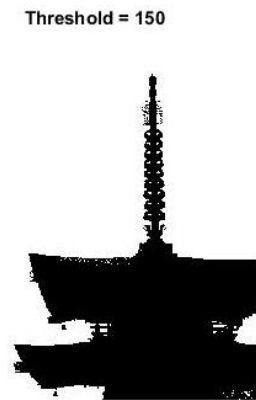
Image:



ต้องการ segmentation หอคอย ออกมา



ซึ่งจากhistogramจะเห็นได้ว่าสามารถแบ่งส่วนมืดและสว่างได้อย่างชัดเจน โดยทำการทดลองตัด Threshold ในช่วง 150,175,200 ตามลำดับได้ผลลัพธ์ดังนี้

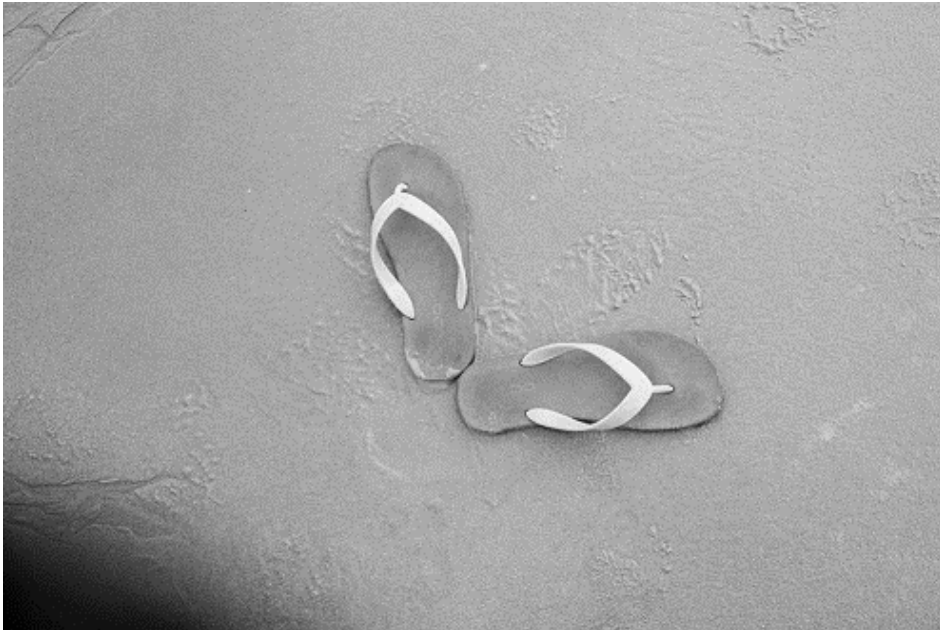


	Threshold = 150			Threshold = 175			Threshold = 200	
	Positive	Negative		Positive	Negative		Positive	Negative
Positive	48532	831	Positive	48977	386	Positive	49316	47
Negative	550	134865	Negative	577	134838	Negative	613	134802
Accuracy	99.25%		Accuracy	99.48%		Accuracy	99.64%	
Error Rate	0.75%		Error Rate	0.52%		Error Rate	0.36%	
Precision	98.88%		Precision	98.84%		Precision	98.77%	
Recall	98.32%		Recall	99.22%		Recall	99.90%	
Specificity	99.59%		Specificity	99.57%		Specificity	99.55%	

เนื่องจากรูปนี้มีความแตกต่างกันของพื้นหลังและวัตถุชัดเจนทำให้สามารถตัด Threshold ได้เลยโดยส่วนพื้นหลังของภาพจะอยู่ในช่วง 200-255 ทำให้สามารถแยกวัตถุได้อย่างชัดเจน

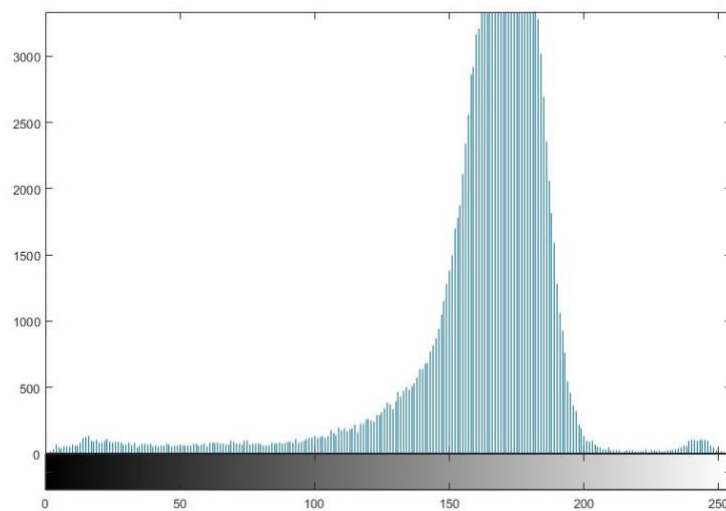
2. Contour-Based Segmentation

รูปที่ใช้คือ

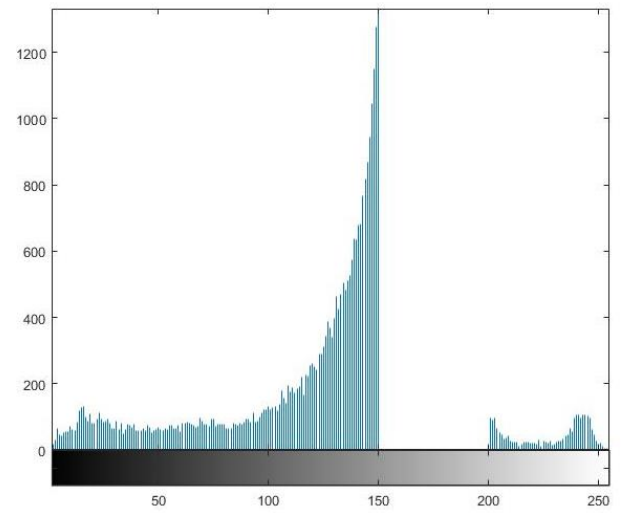
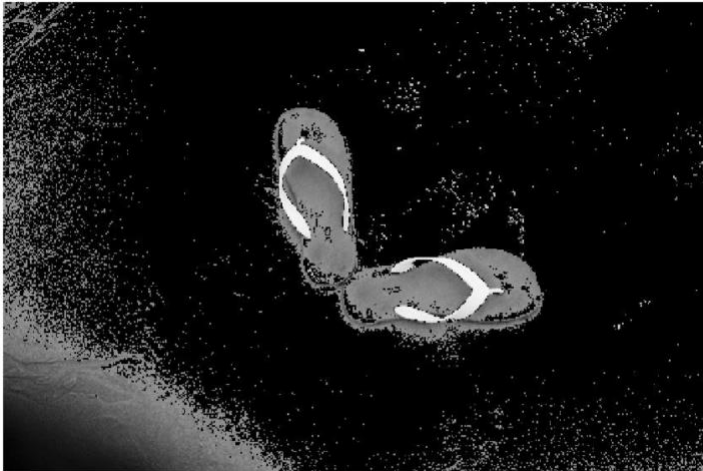


โดยมีจุดประสงค์ต้องการsegmentรองเท้าและที่วางอยู่บนพื้นผิวทราย

โดยมี Histogram ดังนี้



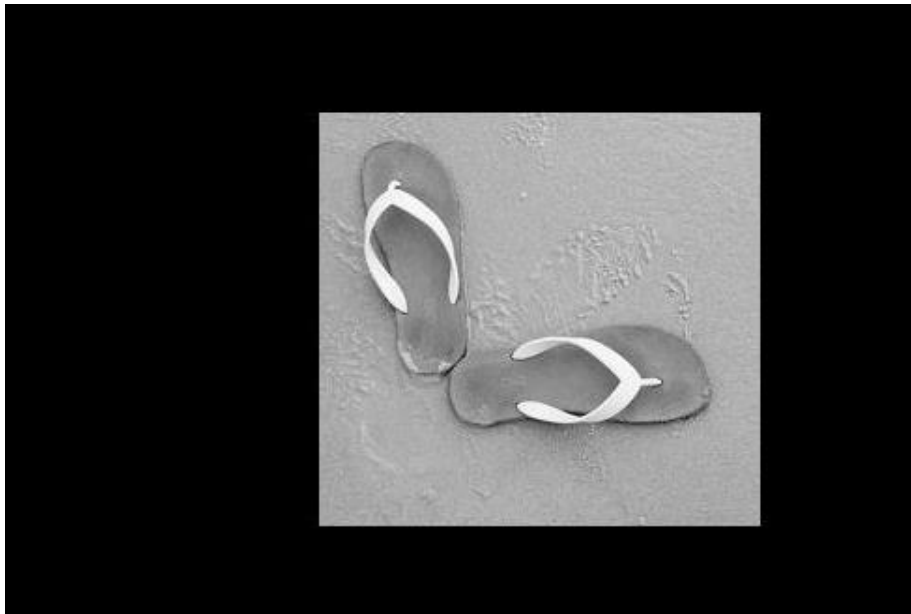
ซึ่งจะเห็นได้ว่าพื้นที่ส่วนใหญ่ซึ่งเป็นทรายจะอยู่ในช่วงประมาณ 150-200 ดังนั้นจึงลองตัด Threshold ในช่วงนี้ออกไปซึ่งได้ผลลัพธ์ดังนี้



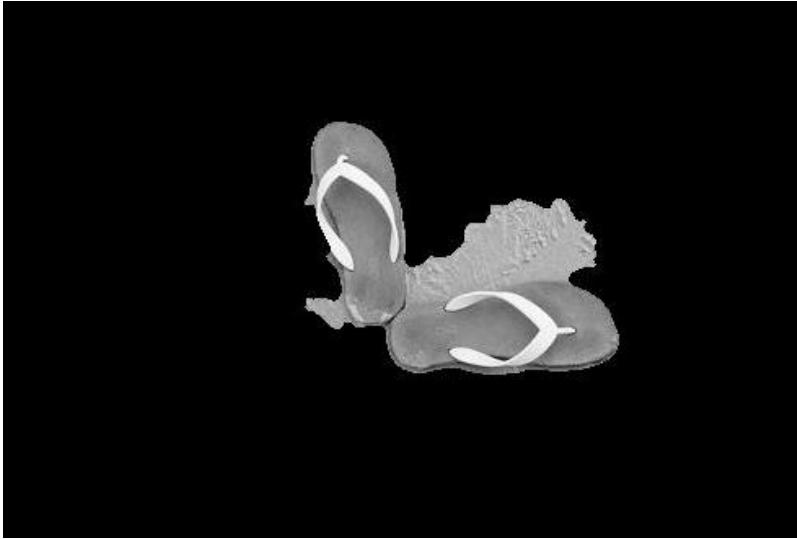
ซึ่งจากรูปจะเห็นได้ว่ายังมีข้อผิดพลาดอยู่หลายจุดได้แก่ ขอบมืดของรูปตรงมุมล่างซ้าย พื้นทรายบางจุดมีสีใกล้เคียงกับรองเท้า

ต่อมาลองใช้ Snake active contour เพื่อที่จะบีบรูปโดยไล่จากพื้นทรายไปยังรองเท้า

เนื่องจากบริเวณมุมล่างขวาของภาพเป็นส่วนมืดแล้วไล่ขึ้นมาจึงกำหนดจุดเริ่มต้นของActive Contour ดังนี้

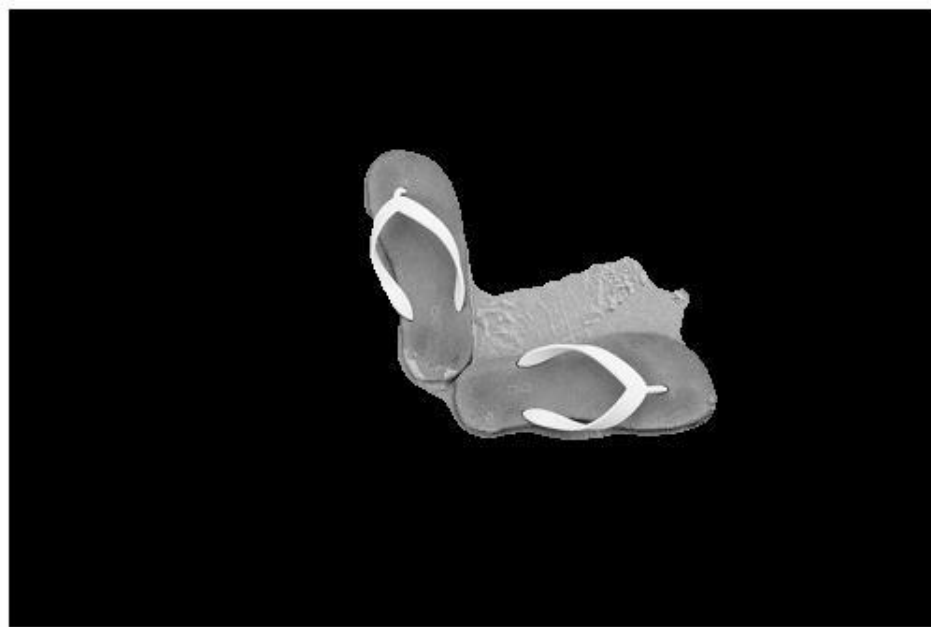


ซึ่งได้ผลลัพธ์จากการรันโปรแกรมใน iterations ที่มากพอได้ดังนี้



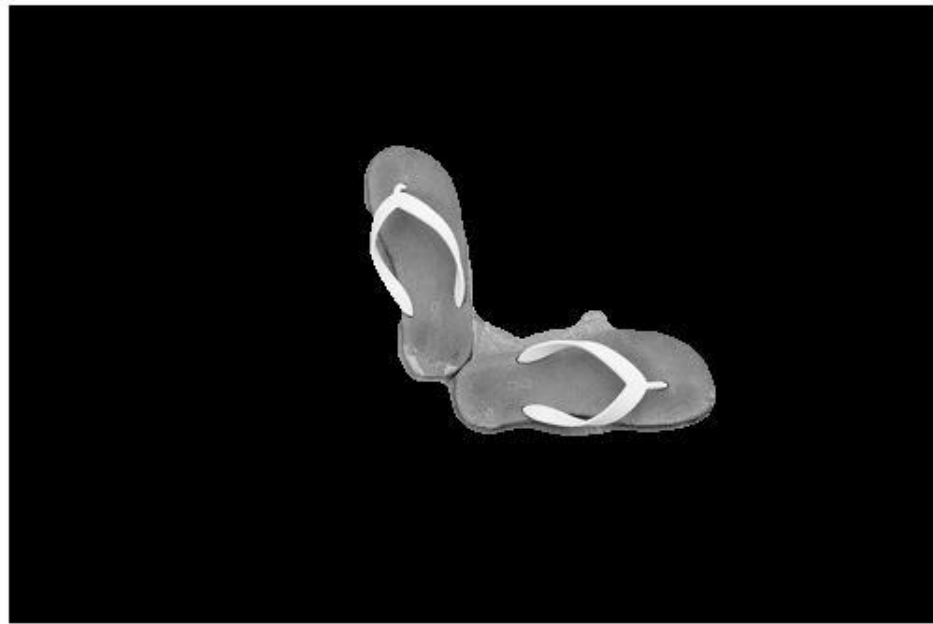
ดังนั้นจึงลองทำการปรับค่า SmoothFactor โดยค่าซึ่งเมื่อเพิ่มจะทำให้อ่อนไหวต่อค่าที่เปลี่ยนไปในการรันมากขึ้น
(default = 1)

SmoothFactor = 2.5

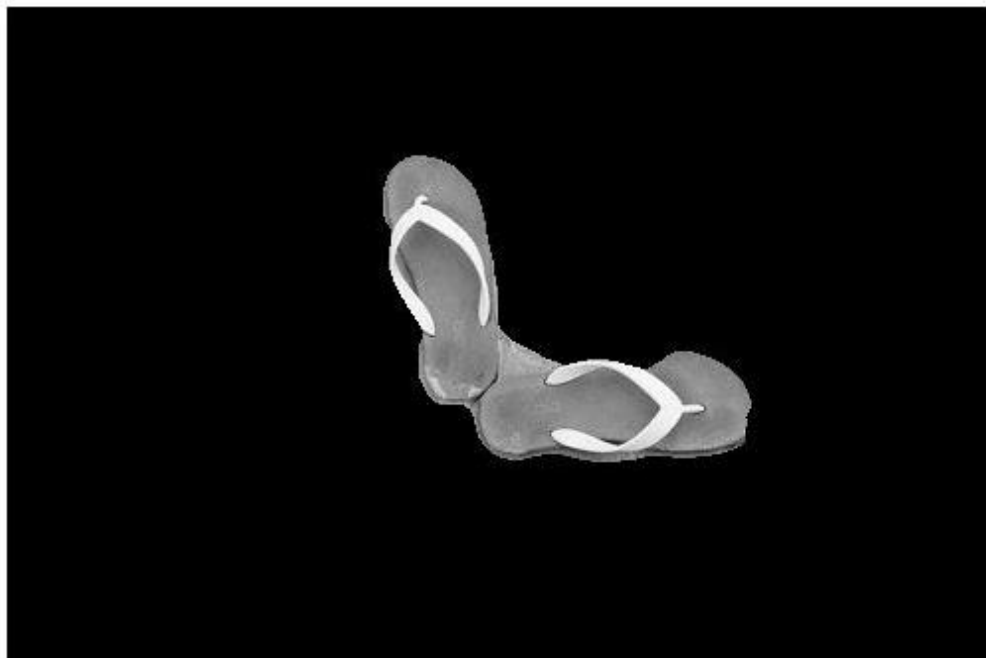


จะเห็นว่าปีบเข้ามามากกว่าเดิม

SmoothFactor = 3

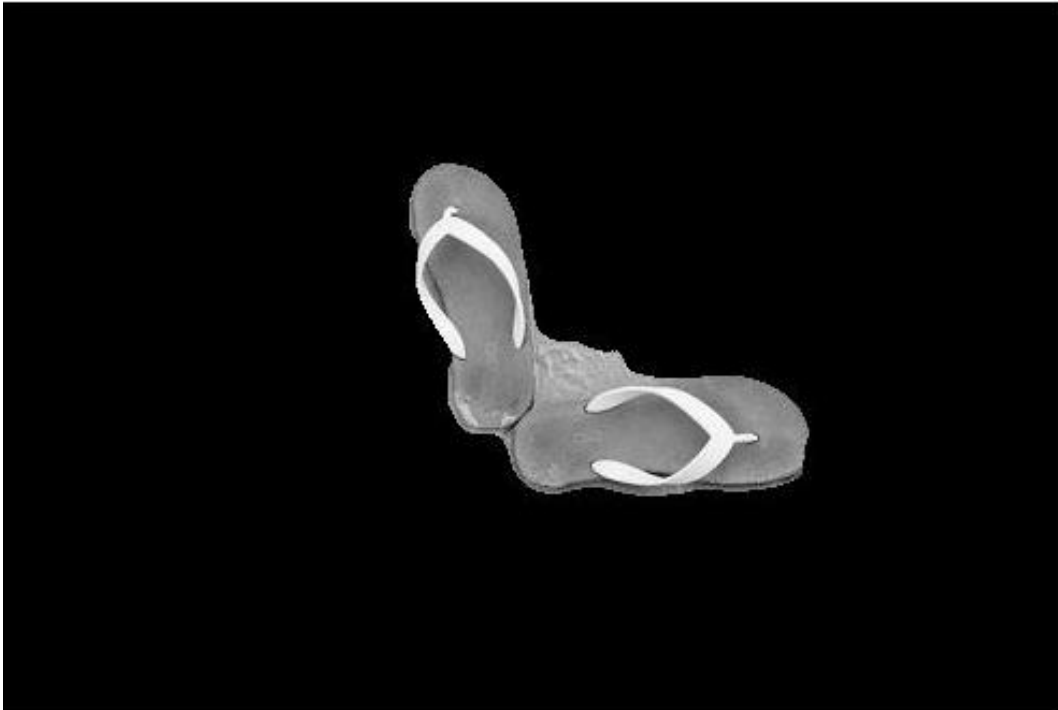


SmoothFactor = 3.5



จะเห็นได้ว่าวงเริ่มกินเข้าไปในรองเท้าบางส่วน

SmoothFactor = 4



เปรียบเทียบ Confusion Matrix ของ SmoothFactor แต่ละค่าได้ดังนี้

SmoothFactor = 3			SmoothFactor = 3.5			SmoothFactor = 4		
	Positive	Negative		Positive	Negative		Positive	Negative
Possitive	11557	442	Possitive	11251	748	Possitive	11449	550
Negative	1023	150328	Negative	747	150604	Negative	1450	149901
Accuracy	99.10%		Accuracy	99.08%		Accuracy	98.78%	
Error Rate	0.90%		Error Rate	0.92%		Error Rate	1.22%	
Precision	91.87%		Precision	93.77%		Precision	88.76%	
Recall	96.32%		Recall	93.77%		Recall	95.42%	
Specificity	99.32%		Specificity	99.51%		Specificity	99.04%	

จะเห็นว่าที่ค่า SmoothFactor = 3 มี Accuracy มากที่สุดแต่มีค่า Precision, Specificity น้อยกว่า SmoothFactor = 3.5

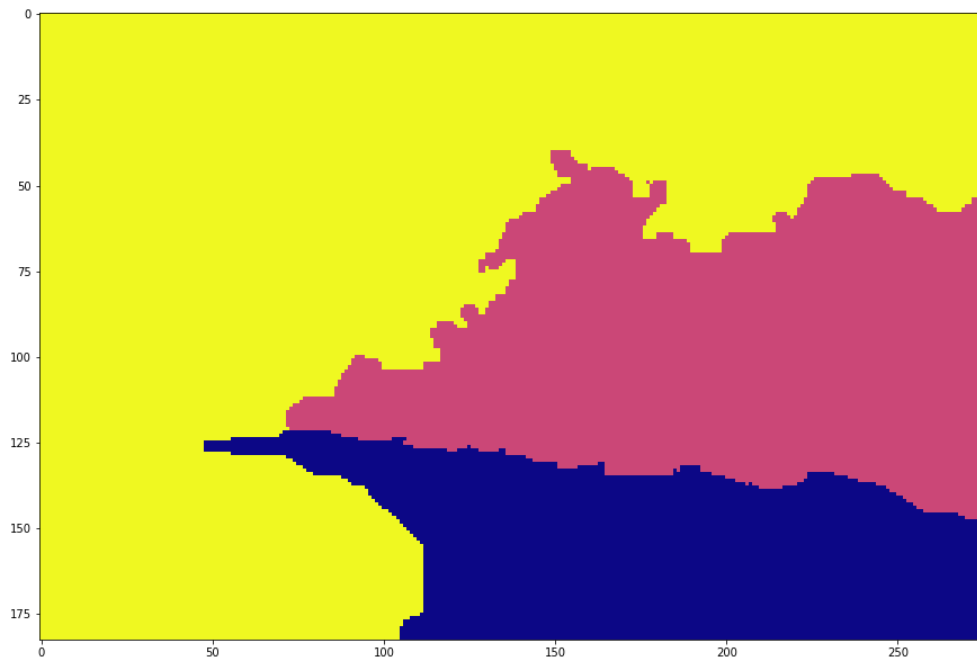
จึงสรุปได้ว่าที่ค่า SmoothFactor นั้นมีความเหมาะสมที่สุด

3.Region-Based Segmentation

ภาพที่เลือกคือ



โดยต้องการแบ่ง 3 ส่วนคือ ต้นไม้,ทราย,ทะเล/ท้องฟ้า โดยมี Ground Truth ดังนี้



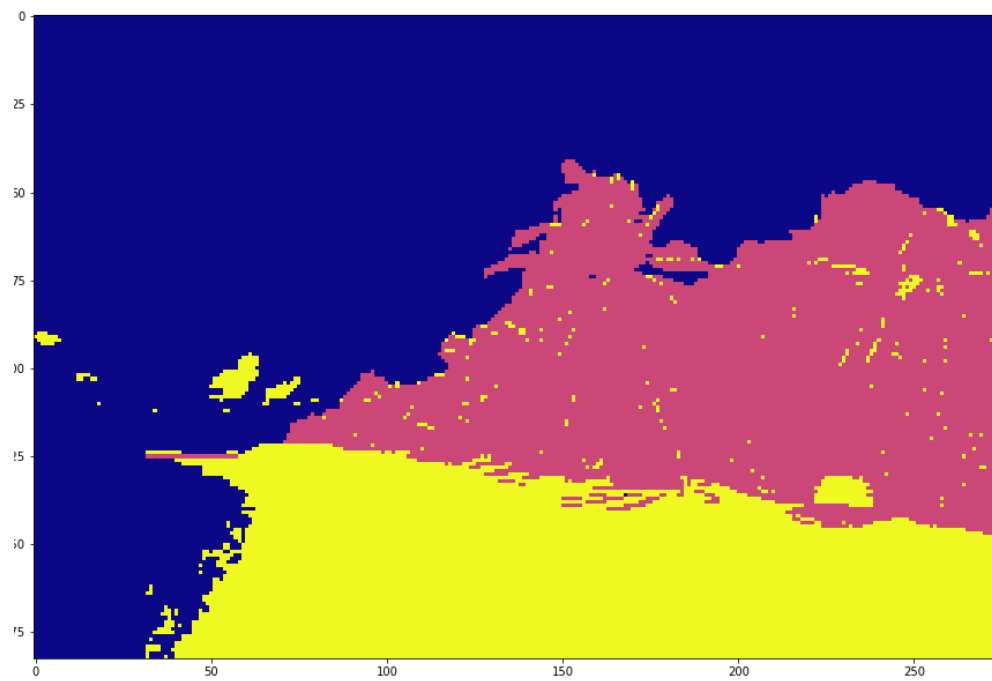
โดยแต่ละสีคือ สีเหลืองคือท้องฟ้า/ทะเล

สีชมพูคือต้นไม้

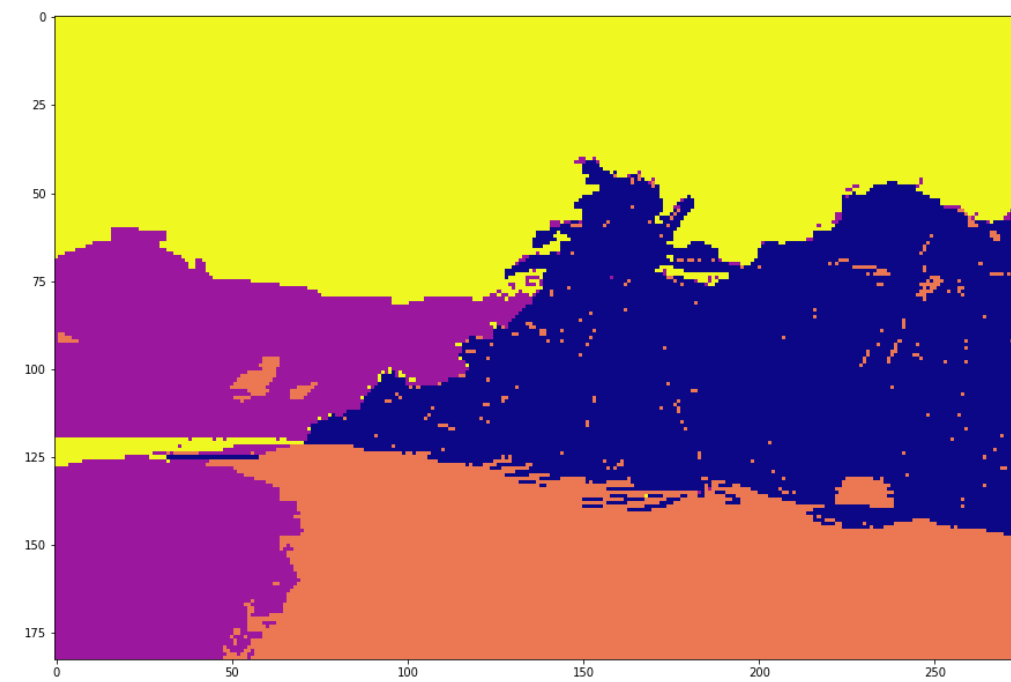
สีน้ำเงินคือทราย

ทำการ Clustering ด้วยวิธี Kmeans โดยใช้ K=3,4,5 ตามลำดับได้ผลลัพธ์ดังนี้

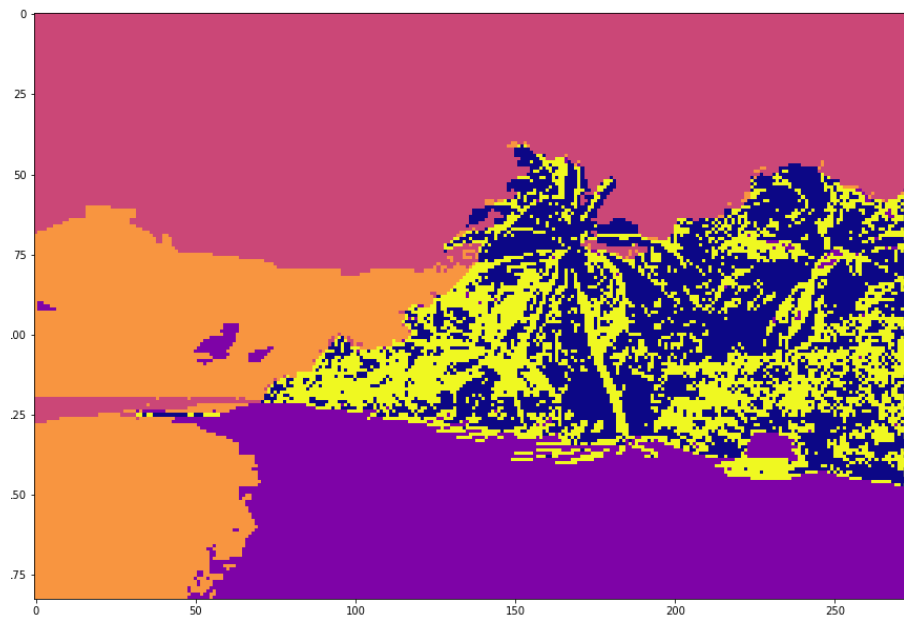
K=3



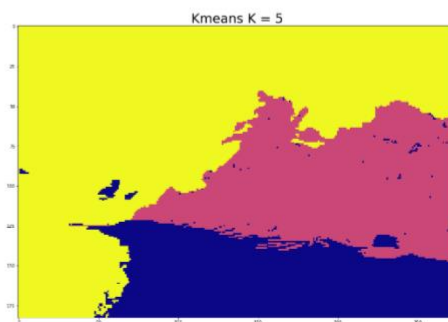
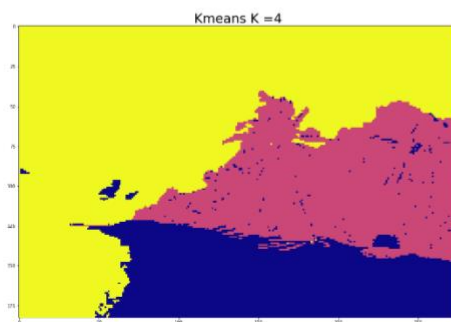
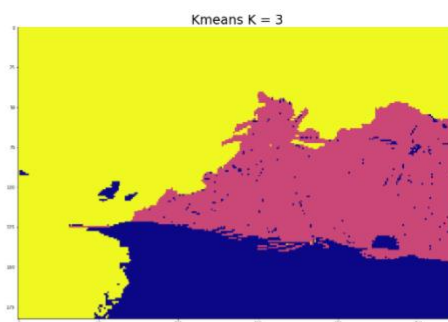
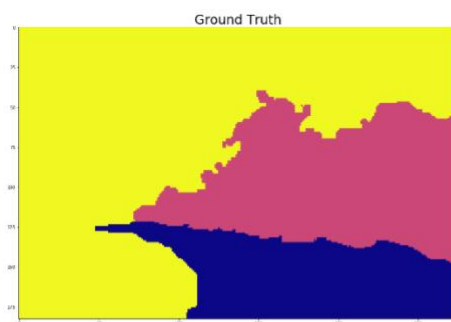
K=4



K=5



ทำการรวมกลุ่มใน K=4,5 และจัดเรียงสีได้ดังนี้



ซึ่งจะเห็นได้ว่า K=3,K=4 แทบไม่เปลี่ยนแปลง แต่ K=5 นั้นเหมือนจะแบ่งกลุ่มต้นไม้นั้นมากขึ้น

Confusion Matrix ของ Kmeans แต่ละค่า K

Kmeans K=3				
	Beach	Tree	Sea/Sky	Recall
Beach	8220	360	6	0.957
Tree	333	12222	395	0.944
Sea/Sky	2303	24	26462	0.919
Precision	0.757	0.970	0.985	
accuracy	0.932			

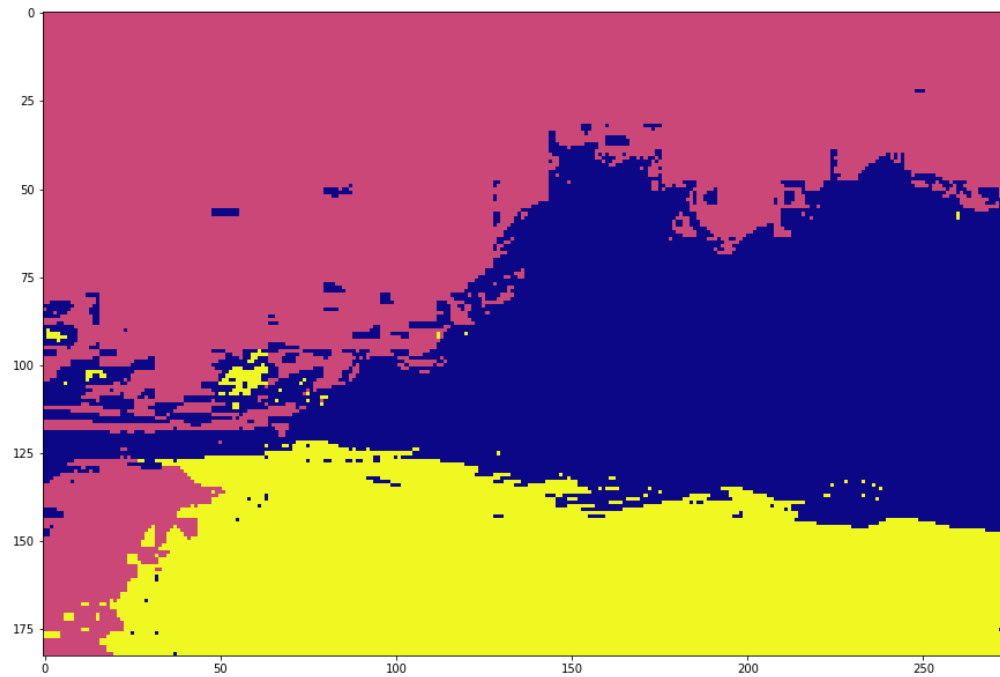
Kmeans K=4				
	Beach	Tree	Sea/Sky	Recall
Beach	8220	360	6	0.957
Tree	333	12208	409	0.943
Sea/Sky	2303	23	26463	0.919
Precision	0.757	0.970	0.985	
accuracy	0.932			

Kmeans K=5				
	Beach	Tree	Sea/Sky	Recall
Beach	8151	431	4	0.949
Tree	140	12416	394	0.959
Sea/Sky	2301	25	26463	0.919
Precision	0.770	0.965	0.985	
accuracy	0.935			

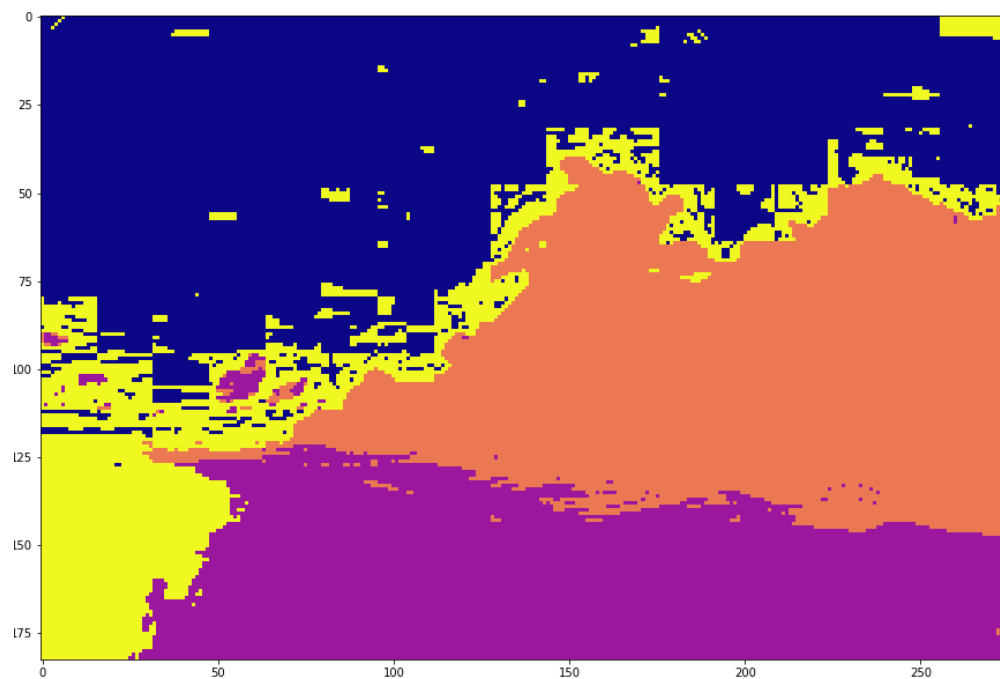
ทำการ Clustering ด้วยวิธี Gaussian Mixture Models(GMM, EM Algorithm)

โดยได้ทดลองทำการแบ่งกลุ่มเป็น N กลุ่มซึ่ง $N = 3, 4, 5$ ได้ผลดังนี้

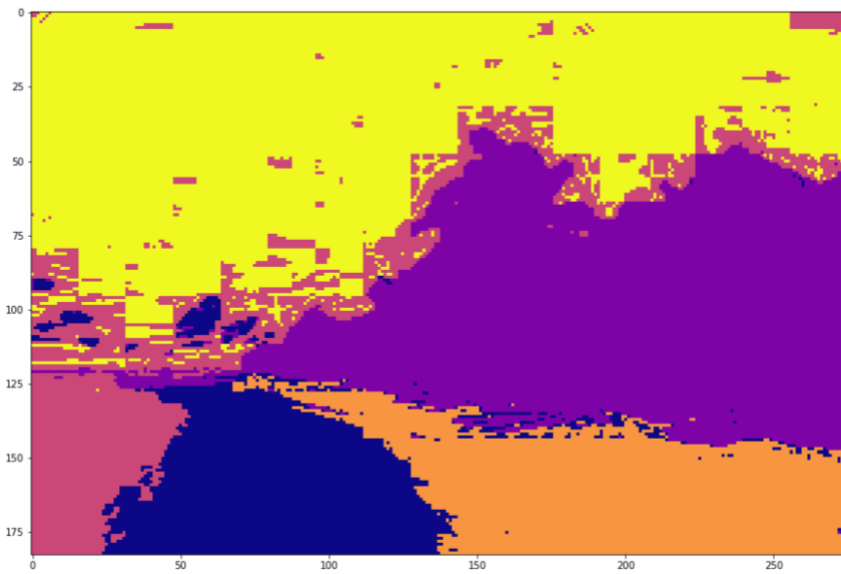
N=3



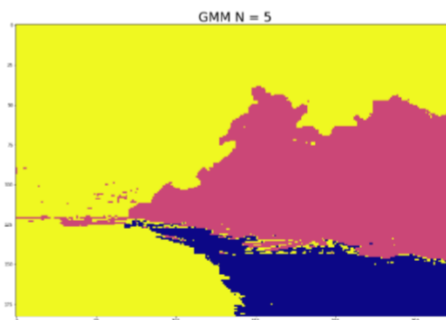
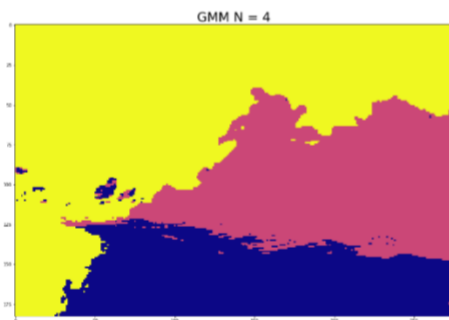
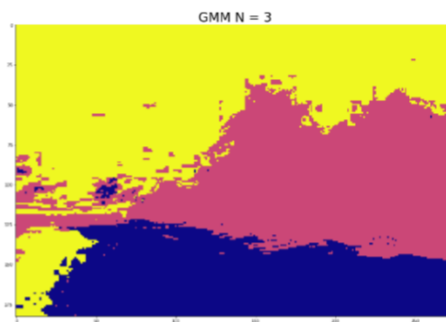
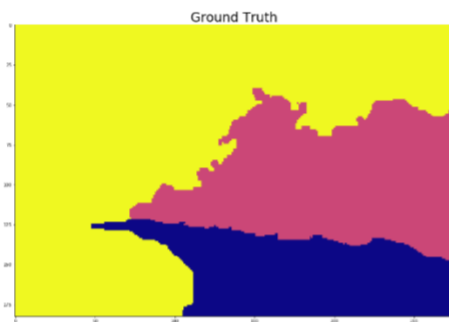
N=4



N=5



ทำการรวมกลุ่มใน N=4,5 และจัดเรียงสีได้ดังนี้



Confusion Matrix ของ GMM แต่ละ N

GMM N=3				
	Beach	Tree	Sea/Sky	Recall
Beach	7755	831	0	0.903
Tree	8	12937	5	0.999
Sea/Sky	4054	2844	21891	0.760
Precision	0.656	0.779	1.000	
accuracy	0.846			

GMM N=4				
	Beach	Tree	Sea/Sky	Recall
Beach	7741	845	0	0.902
Tree	10	12844	96	0.992
Sea/Sky	3444	293	25052	0.870
Precision	0.691	0.919	0.996	
accuracy	0.907			

GMM N=5				
	Beach	Tree	Sea/Sky	Recall
Beach	6258	776	1552	0.729
Tree	0	12888	62	0.995
Sea/Sky	0	453	28336	0.984
Precision	1.000	0.913	0.946	
accuracy	0.944			

จะเห็นว่าเมื่อเทียบกันทั้งวิธี Kmeans ,GMM นั้น สามารถแยกสิ่งที่เราต้องการได้โดยที่ GMM ที่ N=5 นั้นจะมี Accuracy สูงที่สุดรองมา Kmeans K=5 และ Kmeans K=4 ตามลำดับ โดยเมื่อดูจาก GMM ที่ N=3,4 นั้นจะเห็นได้อย่างชัดเจนว่าการแบ่งกลุ่มของน้ำบริเวณชายหาดนั้นยังมีข้อผิดพลาดโดยที่น้ำส่วนที่สีค่อนข้างอ่อนจะไปอยู่พวกเดียวกับทรายแต่ที่ N=5 นั้นได้แยกระหว่างน้ำที่สีอ่อนกับทรายออกมาได้ ในส่วนของ Kmeans นั้น ที่ K=3 ก็สามารถแยกสิ่งที่ต้องการได้อย่างชัดเจนแล้วโดยเมื่อเพิ่มเป็น K=4 จะทำการแบ่งกลุ่มย่อยของน้ำทะเล/ท้องฟ้า เมื่อทำการรวมกลุ่มกันแล้วจึงแทบไม่เห็นการเปลี่ยนแปลงแต่ใน K=5 นั้นจะเพิ่มการแบ่งกลุ่มสีในส่วนของต้นไม้ซึ่งถ้าดูจาก K=3,4 จะเห็นได้ว่าส่วนของต้นไม้ไม่ได้ถูกส่วนของทรายรบกวนดังนั้นที่ K=5 เมื่อรวมกลุ่มแล้วสามารถลดส่วนที่ถูกรบกวนได้จึงทำให้มีประสิทธิภาพเพิ่มขึ้น