Image Segmentation with Metaheuristic algorithms

Gianmarco J. Gallardo
Rochester Institute of Technology
gg4099@rit.edu

Nick Gardner
Rochester Institute of Technology
nag6650@rit.edu

Overview

1. Abstract
2. Introduction
3. Related Work
4. Problem Description
5. Proposed Method
6. Experimental Design / Data
7. Results
8. Discussion / Conclusions

Abstract

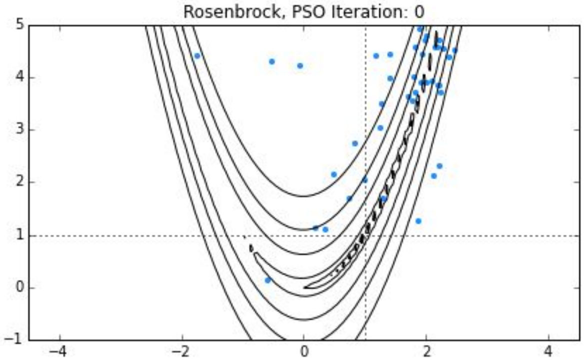
Image Segmentation with Metaheuristics

- Image segmentation can be done by clustering similar pixels into groups.

- In this work, we test several metaheuristic algorithms to solve image segmentation, and compare their performance against the popular k-means clustering method.

- We measure their performance using PSNR, inter-cluster, and intra-cluster distance.

- We found that metaheuristic algorithms are able to perform image segmentation and get comparable results to k-means.



Introduction

Image Segmentation

- Image segmentation is a well-known task in image processing and computer vision, which consists of partitioning an image into regions based on a defined criteria.
- This can be done by clustering similar pixels into groups.
- Semantic segmentation (where groups of pixel have a semantic meaning and not only value similarity) is not covered in this work.

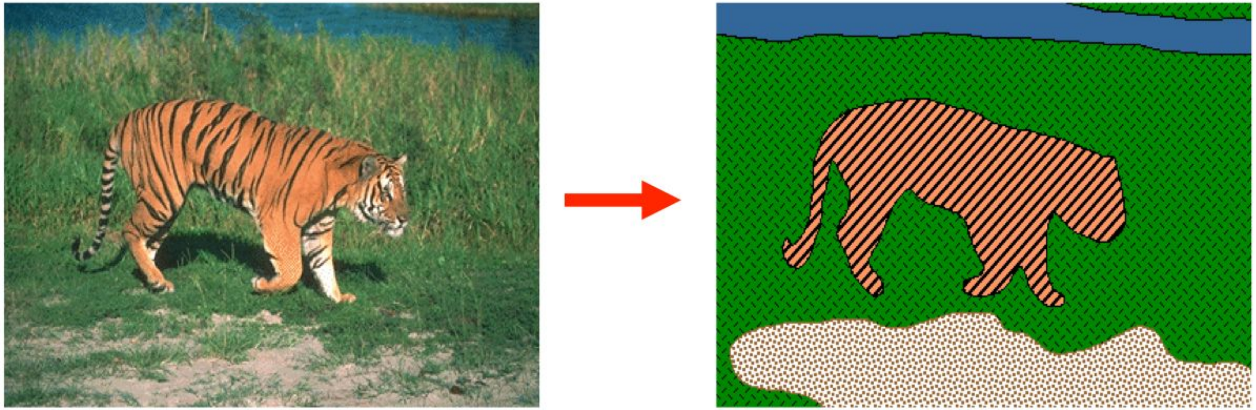
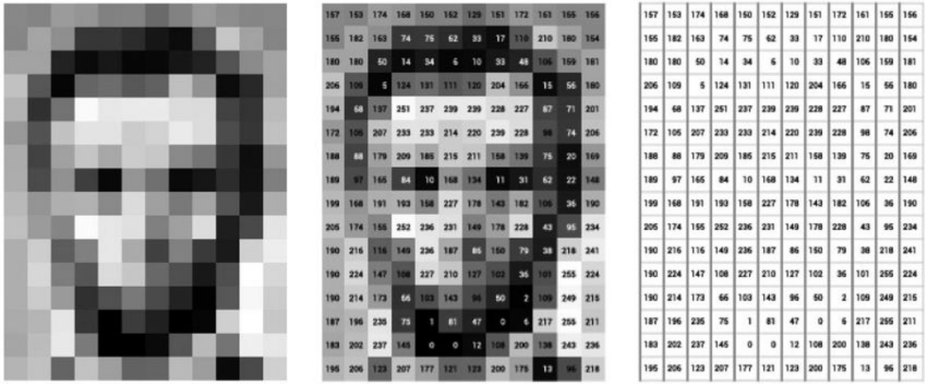
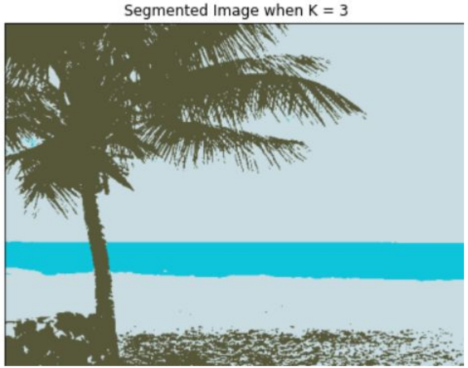


Image Segmentation by clustering pixels

- An image is just a set of numbers (pixels).
- Each pixel is a datapoint for the clustering algorithm.
- Gray scale images have 1D pixels (Intensity). Color images have 3D pixels (RGB)

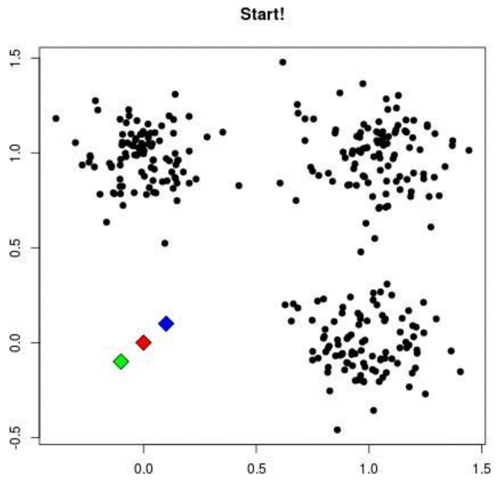


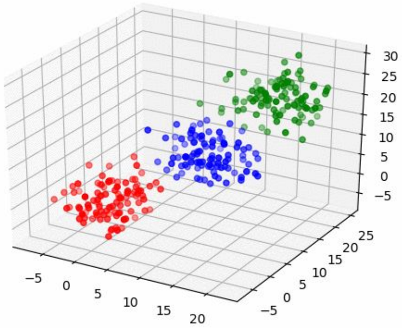


K-means

K-means clustering groups data vectors into a predefined number of clusters, based on the euclidean distance as a similarity measure.

1. Randomly initialize the cluster centroid vectors.
2. For each data vector, assign the vector to the cluster with the closest cluster center, using the euclidean distance between the data vector and the centroid.
3. Re-calculate each cluster’s centroid vector by computing the mean of the data vectors that belong to the cluster.
4. Repeat 2 and 3 until stopping criterion is satisfied (Cluster change is small)





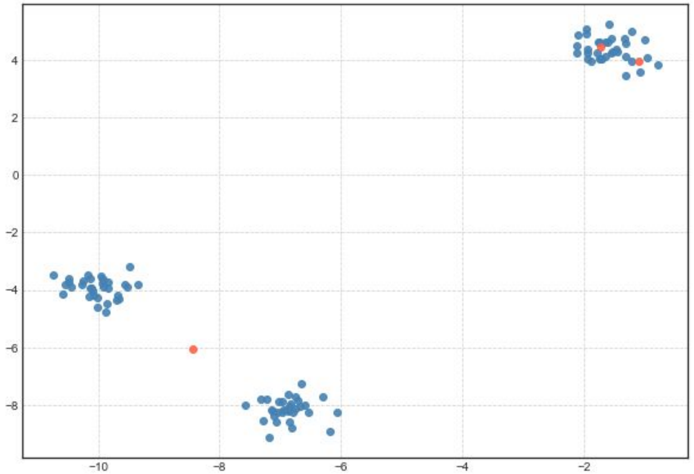
However ...

- K-means performance depends on initial conditions (random initialization), which may cause the algorithm to converge to suboptimal solutions.
- Updating cluster centroids using **metaheuristic algorithms** can make use of their exploitation-exploration capabilities, finding better solutions than K-means.

This work's main contributions are:

➔ *We compare the performance of different metaheuristic algorithms on image segmentation against k-means.*

➔ *We show an easy scheme to transform any metaheuristic algorithm into solving image segmentation by doing clustering*



Related Work

Metaheuristics for clustering

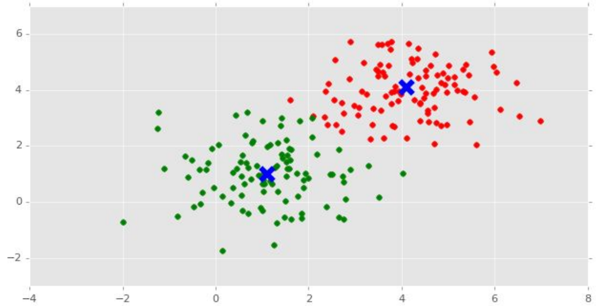
- The use of metaheuristic algorithms for clustering was first introduced in 1991 by Selim in [1].
- Since then, several metaheuristics methods have been proposed for clustering. For example, grouping genetic algorithms, memetic clustering, PSO, CABC, or even combining swarm based algorithms with genetic ones to improve accuracy.
- Exploration has been conducted into combining k-means and metaheuristic optimization such as in [2] Sharma, 2016, where FA was used to optimize k-means initial centroids
- Finally, a similar exploration was performed by [3] Wong, 2011, where particle swarm optimization was used to directly segment an image. This paper was flawed in its selection of test metrics, and we introduce more metaheuristic algorithms to test.

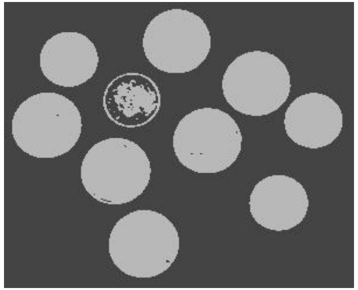
[1] S. Z. Selim and K. Alsultan. A simulated annealing algorithm for the clustering problem. Pattern Recognition, 24(10):1003–1008, 1991. ISSN 0031-3203. Doi: https://doi.org/10.1016/0031-3203(91)90097-O.
[2] A. Sharma and S. Sehgal, "Image segmentation using firefly algorithm," 2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds, 2016, pp. 99-102, doi: 10.1109/INCITE.2016.7857598.
[3] M. T. Wong, X. He and W. Yeh, "Image clustering using Particle Swarm Optimization," 2011 IEEE Congress of Evolutionary Computation (CEC), 2011, pp. 262-268, doi: 10.1109/CEC.2011.5949627.

Problem Description

Image segmentation by pixel clustering

- Having image ***I*** containing ***N*** pixels z_p , we want to find a set of cluster $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ where k is the total number of clusters.
- Each cluster C_j is represented with a centroid m_j. A pixel belongs to the cluster with the closest centroid.
- The task is to find the centroids m_j using a clustering algorithm.
- In this work we explore different alternatives besides k-means clustering, specifically metaheuristic algorithms.





Proposed method

Metaheuristic image segmentation implementation

- A selection of **metaheuristic optimization algorithms**, chosen for their varied nature, are utilized to optimize the cluster centroids for image segmentation
- A single **fitness function** is utilized, **quantization error**, that is selected for its broad nature
- The **population** is constructed as a set of vectors, each vector itself representing a set of **cluster centroids**

Fitness Function

- Quantization error can be described as the **average distance** from each pixel to its cluster centroid
- This metric was selected for the fitness function because it is the most general of implemented metrics
  - Intra-cluster and inter-cluster distance target more specific quantities while quantization error simply details average pixel distance

$$Q_e = \frac{\sum_{j=1}^{N_c} [\sum_{\forall z_p \in C_j} d(z_p, m_j)]/|C_j|}{N_c}$$

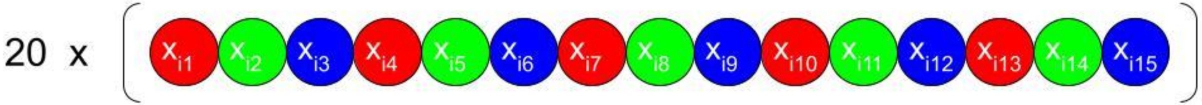
Where:

- N_C = number of clusters
- Z_P = p^{th} pixel
- C_j = j^{th} cluster
- m_j = mean of cluster j
- $|C_j|$ = number of pixels in cluster j
- d(x, y) = Euclidean distance between x and y

Population

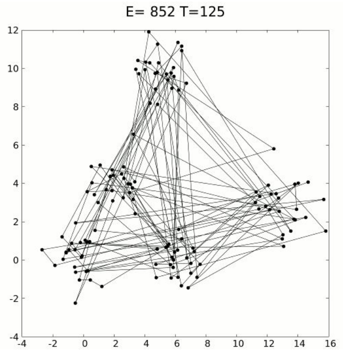
- The population is constructed as a set of k vectors, each of size d
- Each subvector is a flat array containing cluster centroids of shape [number of clusters x number of channels]
- For an RGB image with five clusters and a population size of 20, the population would consist of 20 vectors, each of size 15.

$$X = [x_1, x_2, ..., x_k] \text{ where } x_i = [x_{i1}, x_{i2}, ..., x_{id}]$$

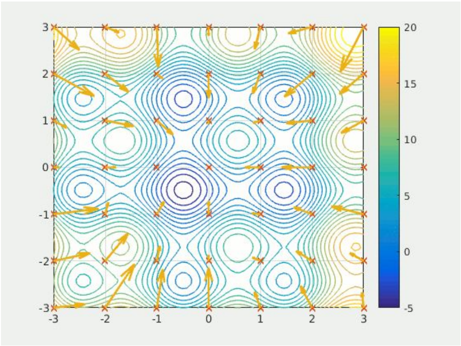


Optimization Algorithms

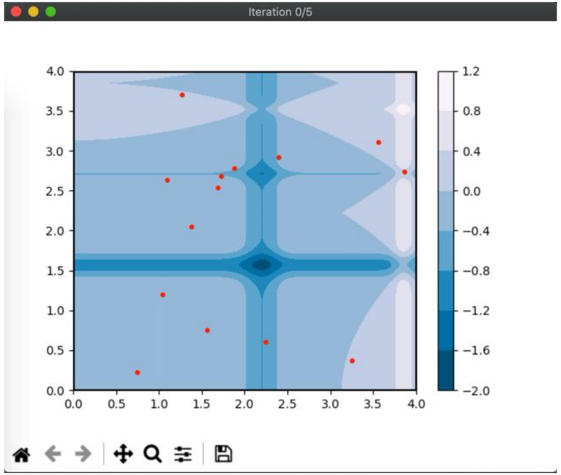
Simulated Annealing (SA)



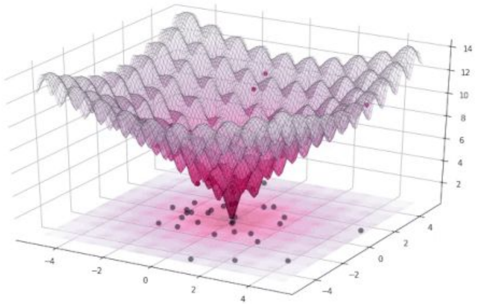
Particle Swarm Optimization (PSO)



Firefly Algorithm (FA)



Differential Evolution (DE)



Bat Algorithm (BA)

Experimental Design / Data

Data

- We perform image segmentation on 4 different images: “lena”, “tiger”, “cameraman”, and “coins”.
- Besides being classic images using by the image processing community, we chose these images because they show different challenges.

lena

tiger

cameraman

coins

Experimental setup

- We set a fixed number of clusters to be found on each image, with 6 for lena, 8 for tiger, 4 for cameraman, and 2 for coins.
- We use the following hyperparameters for each metaheuristic algorithm, after some tuning.

    - For SA we use $T_0 = 10$, $\beta = 0.8$, and 1000 iterations.
    - For DE we use $F = 1$, $C_r = 0.5$, population size of 20, and 50 iterations.
    - For PSO we use $\alpha = 1.5$, $\beta = 0.5$, population size of 20, and 50 iterations.
    - For FA we use $\gamma = 0.0001$, $\alpha = 0.1$, $\beta = 1$, population size of 20, and 50 iterations.
    - For BA we use $\alpha = 0.8$, $\gamma = 0.9$, population size of 20, and 50 iterations.

- It is worth mentioning that we use 1000 iterations for SA to mimic the effective number of iterations of the population based algorithms. (20x50)

Metrics

- **Peak Signal-to-Noise Ratio (PSNR):** It measures the ratio between the maximum possible power of a signal and the power of corrupting noise. Higher PSNR means that the segmentation is of good quality and the original image can be detected.

$$PSNR = 10\log_{10}\frac{1}{\sqrt{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I_{i,j} - S_{i,j})^2}}$$

- **Inter-cluster distance:** It measures the minimum distance between 2 cluster centroids. Higher inter-cluster distance means that each cluster is well separated from the other.

$$d_{min} = min_{\forall i,j,i\neq j}\{d(m_i, m_j)\}$$

- **Intra-cluster distance:** It measures the maximum average distance between a cluster centroid and its pixels. Lower intra-cluster distance means that each cluster is compact.

$$\bar{d}_{max} = max_{j=1,\ldots,N_c}\left\{\frac{\sum_{\forall z_p \in C_j} d(z_p, m_j)}{|C_j|}\right\}$$

Results

Results

- We average results over 5 trials for all the experiments.
- For PSNR, the best method is k-means across all the images. Second best methods are either SA or FA.
- For inter-cluster distance there is no best method across all the images. K-means, SA and PSO perform the best for some images.
- For intra-cluster distance, k-means, SA and FA perform the best on some images.
- We can notice that for “coins”, all the algorithms perform similarly.

Table 1: PSNR results for different clustering algorithms and different images. We show the mean (± standard deviation) over 5 trials. A higher PSNR corresponds to better results. Bold numbers show best values, and underlined numbers show second best values.

METHODS	Lena	Tiger	Cameraman	Coins
K-means	**25.5910** (± 0.0006)	**24.9751** (± 0.0698)	**25.9339** (± 0.0468)	**23.9844** (± 0.0000)
SA	22.5184 (± 2.5776)	22.8917 (± 1.5042)	23.9775 (± 3.4892)	23.9839 (± 0.0009)
DE	20.9154 (± 0.5908)	20.5944 (± 1.0359)	22.5594 (± 1.7642)	23.9590 (± 0.0226)
PSO	23.0332 (± 1.2088)	22.2437 (± 1.0630)	21.8333 (± 1.3805)	23.9676 (± 0.0272)
FA	24.0371 (± 1.1009)	21.4019 (± 1.2603)	25.1963 (± 0.9350)	23.9834 (± 0.0011)
BA	22.0791 (± 1.1043)	20.5917 (± 0.7558)	22.8699 (± 3.2174)	23.9561 (± 0.0092)

Table 2: Inter-cluster distance (d_{min}) results for different clustering algorithms and different images. We show the mean (± standard deviation) over 5 trials. A higher Inter-cluster distance corresponds to better results. Bold numbers show best values, and underlined numbers show second best values.

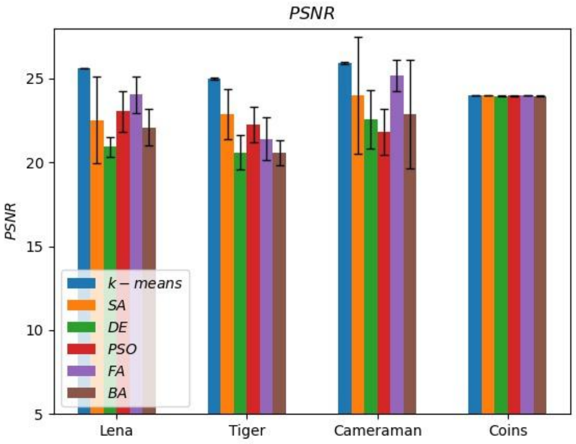
METHODS	Lena	Tiger	Cameraman	Coins
K-means	**0.1761** (± 0.0000)	0.1212 (± 0.0026)	**0.2236** (± 0.0396)	0.7913 (± 0.0002)
SA	0.1209 (± 0.0191)	0.1327 (± 0.0209)	0.1653 (± 0.0091)	**0.7917** (± 0.0003)
DE	0.1115 (± 0.0265)	0.0971 (± 0.0270)	0.2134 (± 0.0721)	0.7906 (± 0.0022)
PSO	0.1158 (± 0.0556)	**0.1450** (± 0.0271)	0.1530 (± 0.0413)	0.7903 (± 0.0017)
FA	0.1451 (± 0.0214)	0.1221 (± 0.0230)	0.2089 (± 0.0544)	0.7914 (± 0.0005)
BA	0.1514 (± 0.0389)	0.1281 (± 0.0404)	0.1938 (± 0.0556)	0.7910 (± 0.0024)

Table 3: Intra-cluster distance ($\bar{d}_{max}$) results for different clustering algorithms and different images. We show the mean (± standard deviation) over 5 trials. A lower Intra-cluster distance corresponds to better results. Bold numbers show best values, and underlined numbers show second best values.

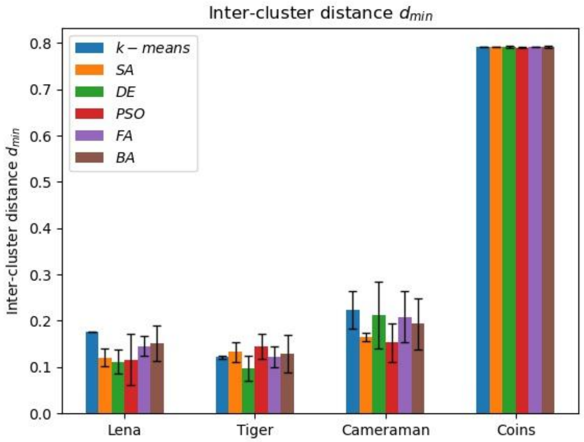
METHODS	Lena	Tiger	Cameraman	Coins
K-means	0.1180 (± 0.0000)	**0.1249** (± 0.0052)	**0.0981** (± 0.0071)	0.1251 (± 0.0002)
SA	0.1491 (± 0.0459)	0.1354 (± 0.0131)	0.1399 (± 0.0950)	**0.1246** (± 0.0003)
DE	0.1606 (± 0.0060)	0.1749 (± 0.0174)	0.1264 (± 0.0112)	0.1258 (± 0.0023)
PSO	0.1345 (± 0.0174)	0.1404 (± 0.0077)	0.1278 (± 0.0158)	0.1261 (± 0.0018)
FA	**0.1165** (± 0.0159)	0.1421 (± 0.0109)	0.1022 (± 0.0102)	0.1250 (± 0.0005)
BA	0.1409 (± 0.0190)	0.1468 (± 0.0132)	0.1497 (± 0.0848)	0.1254 (± 0.0026)

Results

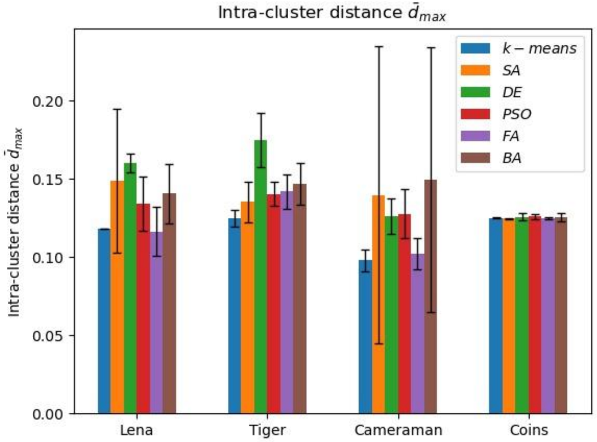
PSNR



Inter-Cluster distance



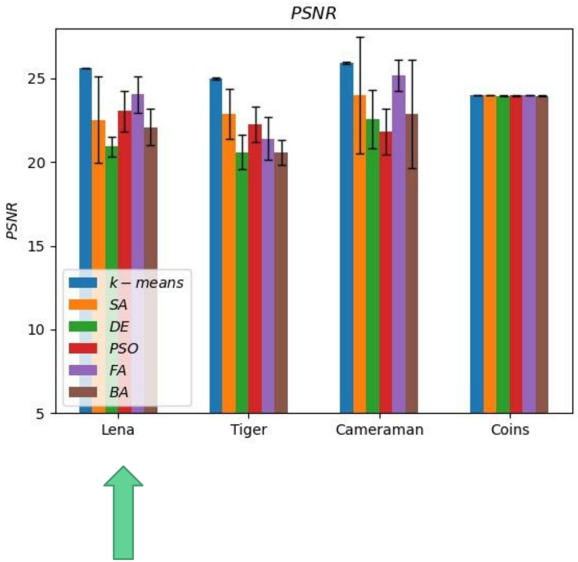
Intra-Cluster distance



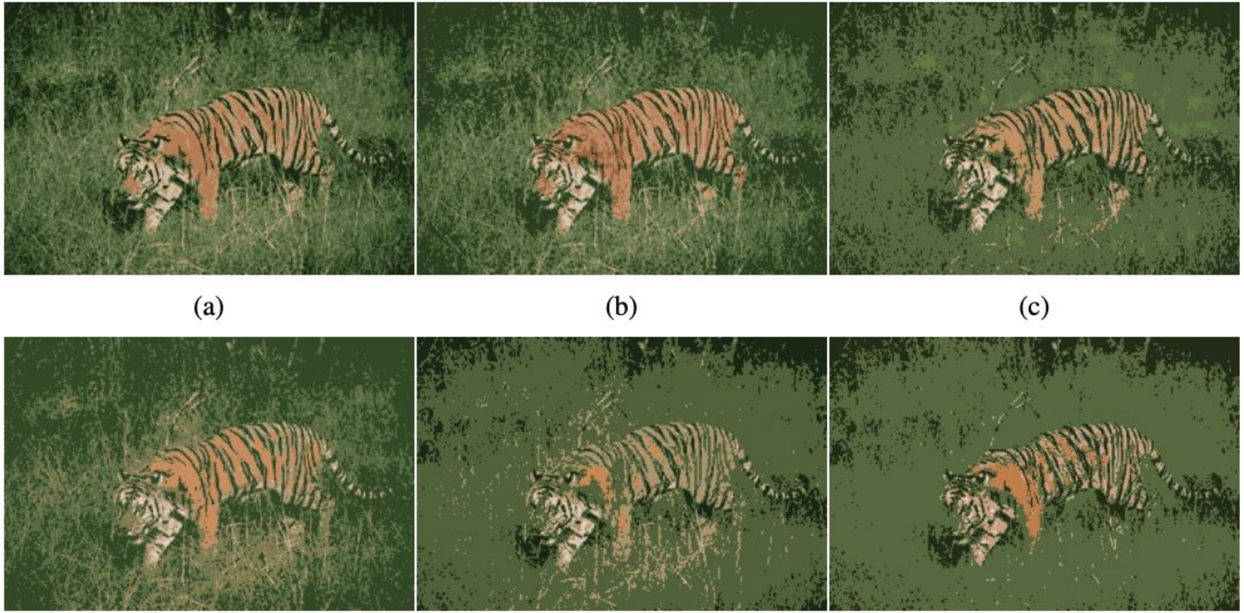
Segmentation - Lena

Figure 2: Segmented images for "lena" using 6 clusters. (a) k-means. (b) Simulated Annealing. (c) Differential Evolution. (d) Particle Swarm Optimization. (e) Firefly Algorithm. (f) Bat Algorithm

PSNR



Segmentation - Tiger

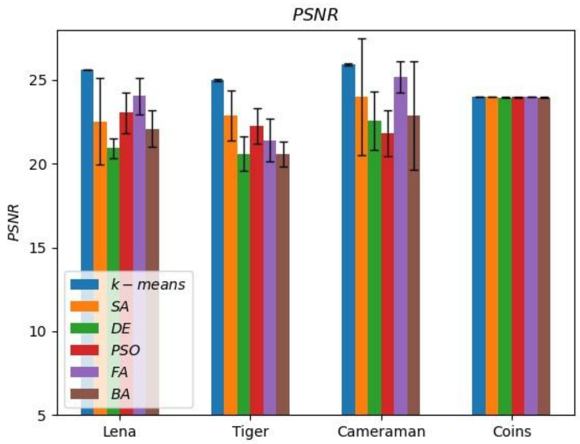


(a) (b) (c)

(d) (e) (f)

Figure 3: Segmented images for "tiger" using 8 clusters. (a) k-means. (b) Simulated Annealing. (c) Differential Evolution. (d) Particle Swarm Optimization. (e) Firefly Algorithm. (f) Bat Algorithm

PSNR



Segmentation - Cameraman

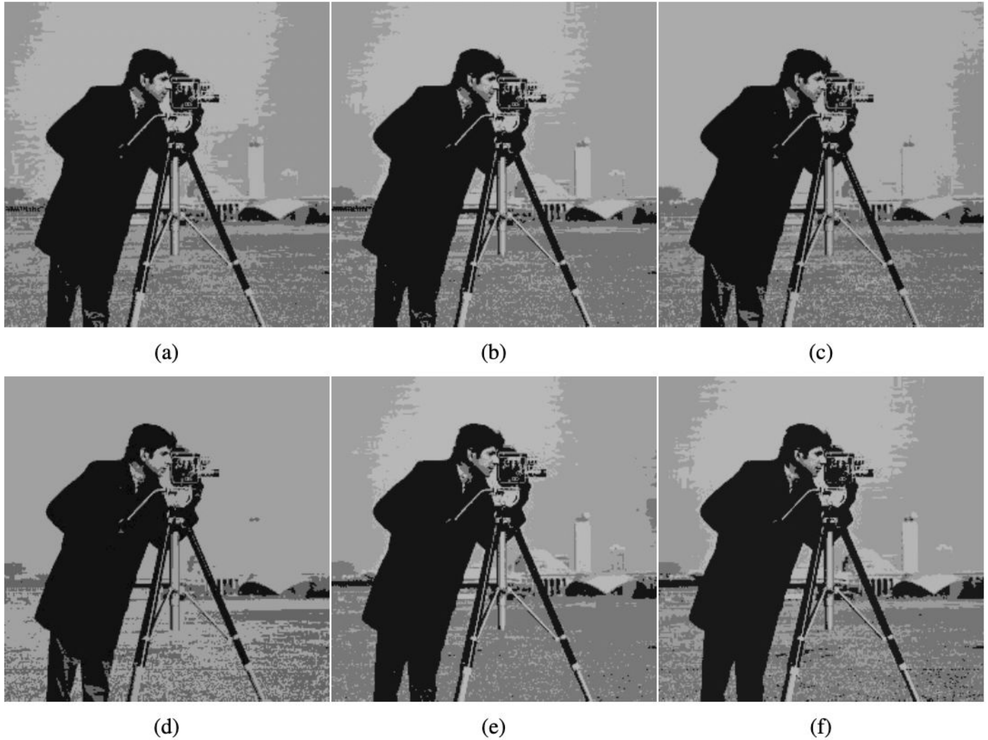
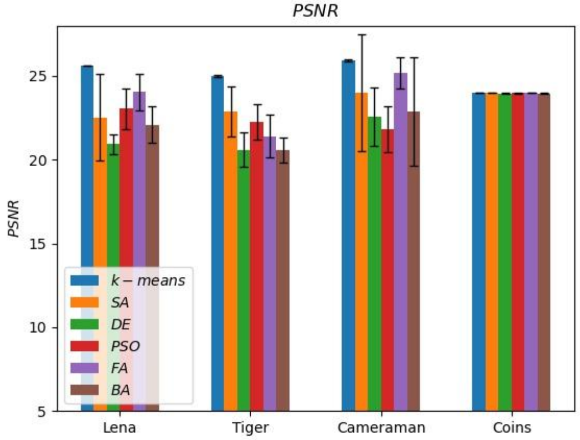


Figure 4: Segmented images for "cameraman" using 4 clusters. (a) k-means. (b) Simulated Annealing. (c) Differential Evolution. (d) Particle Swarm Optimization. (e) Firefly Algorithm. (f) Bat Algorithm

PSNR



Segmentation - Coins

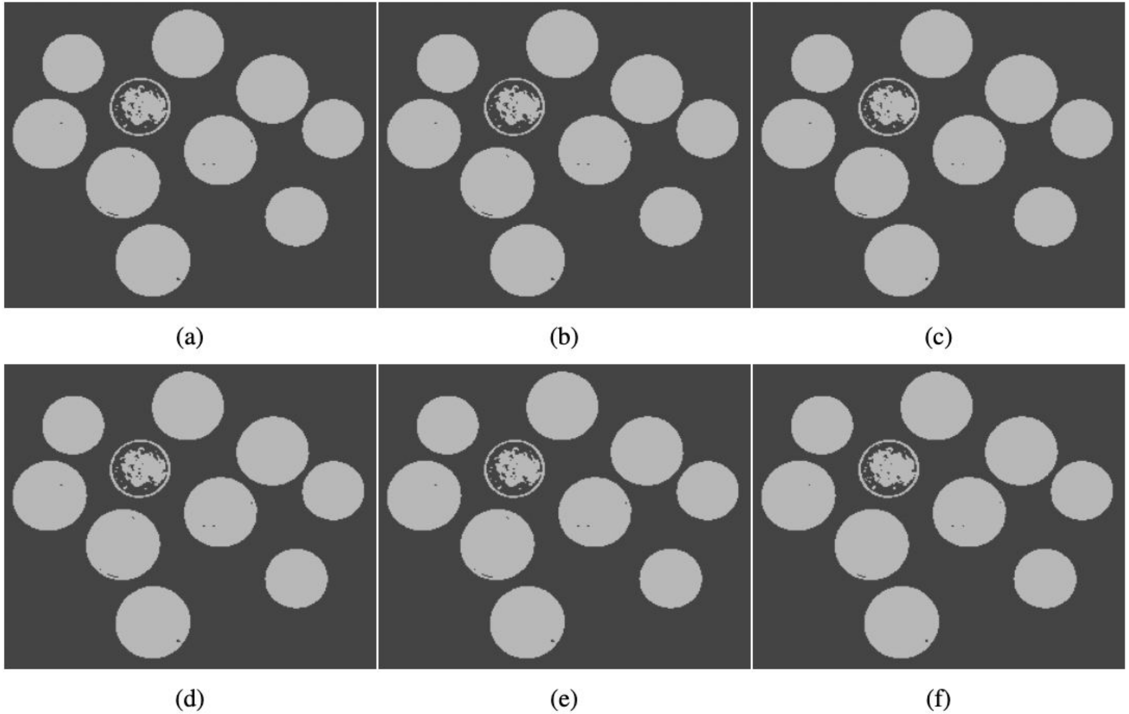
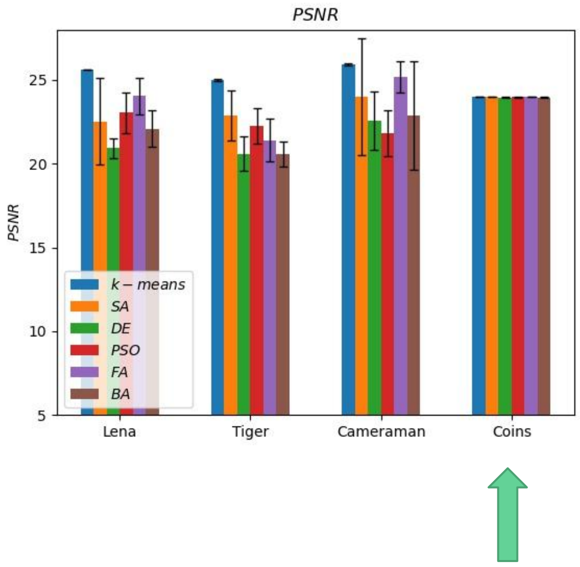


Figure 5: Segmented images for "coins" using 2 clusters. (a) k-means. (b) Simulated Annealing. (c) Differential Evolution. (d) Particle Swarm Optimization. (e) Firefly Algorithm. (f) Bat Algorithm

PSNR



Discussion / Conclusion

Discussion

- Different results than similar work (Wong 2011)
    - Indicative of issues discussed with their experimental design
- Higher standard deviation for metaheuristic results than K-means
    - Perhaps solved with more iterations, so that metaheuristic algorithms more consistently converge
- Unexpected results for the tiger image that included consistently above-average performance from SA
    - Unique solution space and example of the No Free Lunch theorem
- Every algorithm performed almost identically on the coins image
    - Might have been better to exclude this image or use it solely as an initialization
- Validity of testing metrics for describing good segmentation
    - Additional metrics might improve ability to describe segmentation results

Conclusion

- Metaheuristic methods, under fair testing conditions, can have comparable success to k-means for image segmentation
- No matter how generally effective the algorithm is, there will be situations where a typically less effective method is more effective.

In the future:

➔ Expand pixel representation to include spatial information
➔ Introduce semantic pixel labeling to perform semantic segmentation
➔ More challenging images
➔ Explore fitness function and test metric options

Thank you