Setup & References

Python PyCharm was used to run the code. This was done in Python 3.x with Anaconda. ImageIO needs to be installed via pip. Crl+Shift+E can be used in PyCharm to run the code in segments by selecting the code you want to run. There are two files, one for each problem.

Pandas pivot table documentation
https://pandas.pydata.org/pandas-docs/stable/generated/pandas.pivot_table.html

Piazza – HW7 Useful Tips
https://piazza.com/class/jchzguhsowz6n9?cid=1121

Piazza – HW7 Images
https://piazza.com/class/jchzguhsowz6n9?cid=1192

Numpy argmax for word and pixel look up
https://stackoverflow.com/questions/5469286/how-to-get-the-index-of-a-maximum-element-in-a-numpy-array-along-one-axis

ImageIO documentation
https://imageio.github.io/

DAF book – March 28th

Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Problem 1 – Topic Modeling.

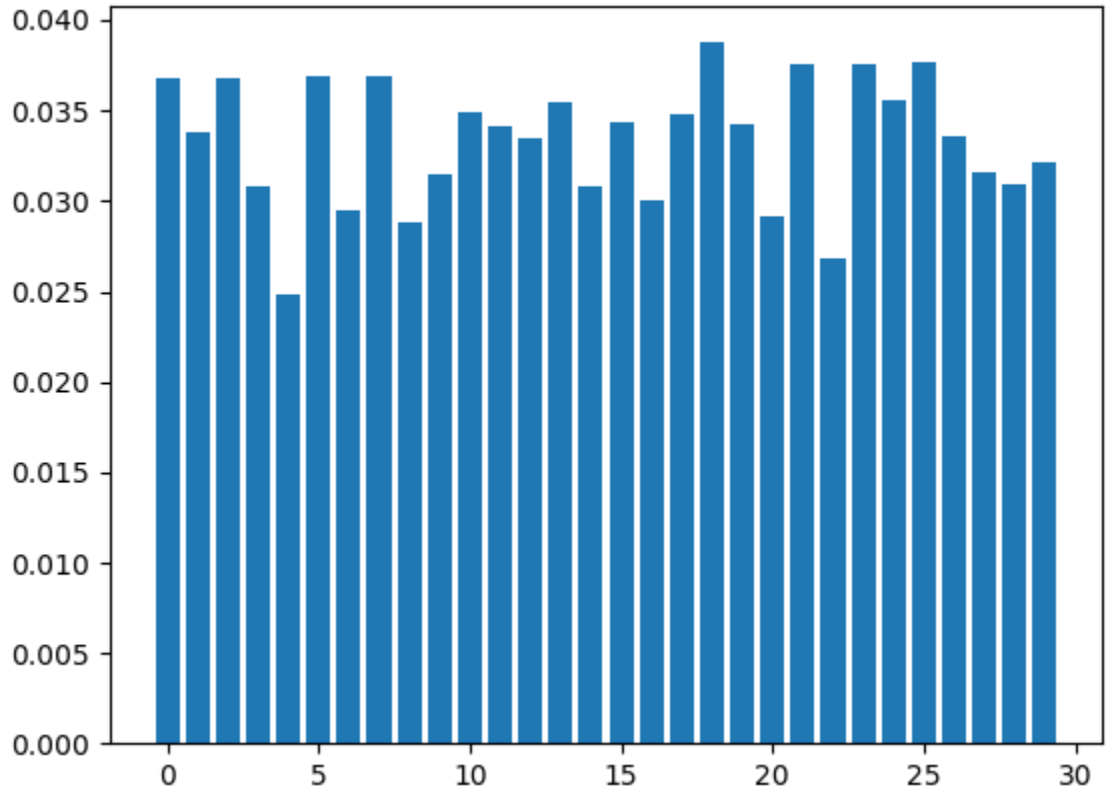
The word counts and topics were clustered to 30 clusters using the topic model. Detailed comments can be found in the code. Nevertheless, the model was done in log space to preserve precision in the probabilities since I am working with very small numbers. Additionally, the prior probability was initialized randomly across the documents (kmeans was NOT used as per Piazza post). And the probability distribution of topics was initialized evenly across the 30 topics.

A smoothing constant was applied to the words with zero probability across topics to ensure an entire topic does not obtain zero probability due to a single word missing in the distribution.

Tracking changed in pi was used as the convergence criteria to stop EM because tracking changes in w showed an oscillating pattern which could indicate that EM has found the minimum and is going back and forth between minimum values. This oscillating pattern was avoided when tracking pi.

Lastly, calculations were done in matrix operations. Specifically, the loops over words topics and documents were replaced with summations, matrix multiplication and element wise multiplication depending on the formula.

The bar chart bellows shows the probability distribution of topics i.e. pi values. It is interesting because these values were all set to 1/30 and so evenly distributed. Now they are more closely representative of the data.

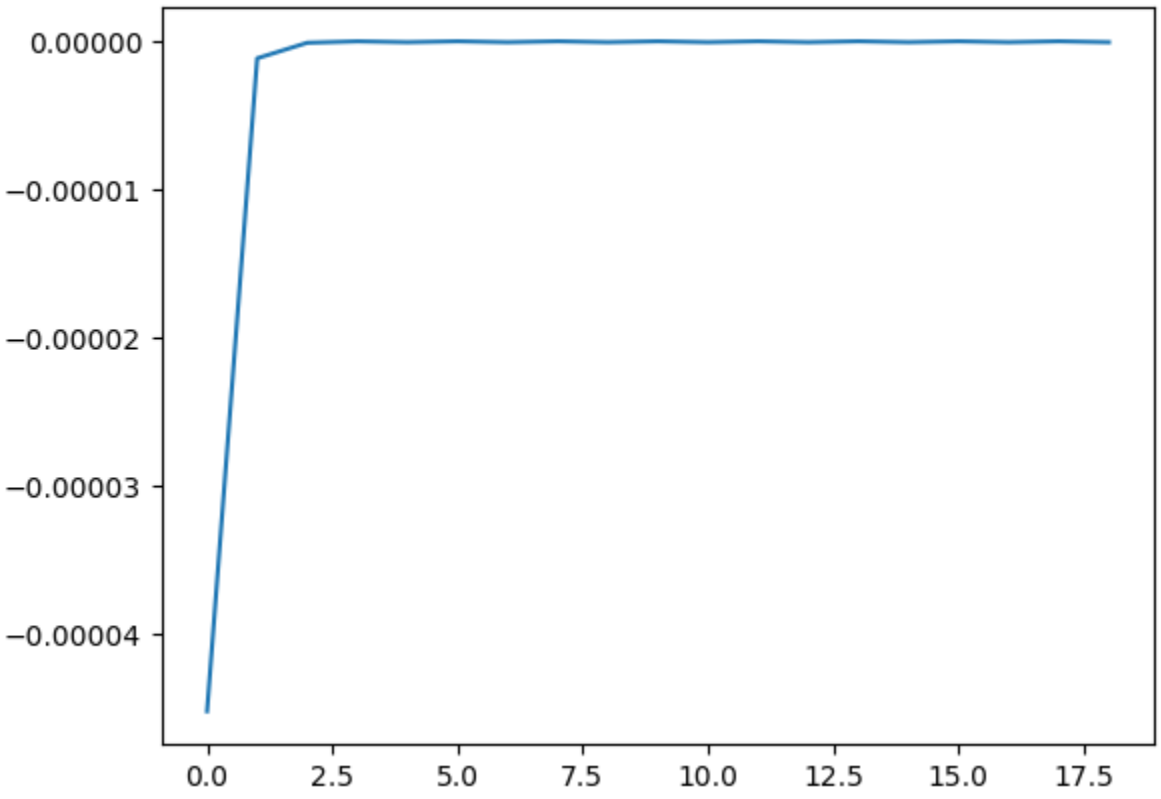


Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

The table below shows the top ten words across all the clusters. Number 1 is the top word. It is interesting that the top word selected is also the most frequent word in the corpus. Additionally, the documents seem to be discussing machine learning topic and neural networks since all the clusters agree on the most probably words.

1	2	3	4	5	6	7	8	9	10	cluster
network	model	function	learning	input	system	set	algorithm	error	neural	0
network	model	data	function	learning	system	input	algorithm	set	method	1
model	network	learning	input	function	algorithm	data	output	set	neural	2
network	model	learning	input	set	unit	neural	training	system	data	3
network	learning	function	input	neural	set	system	model	data	error	4
network	learning	function	neural	input	model	algorithm	data	system	output	5
network	learning	model	input	neural	function	algorithm	set	output	training	6
network	model	neural	learning	input	function	algorithm	system	set	unit	7
network	model	learning	data	set	function	neural	algorithm	system	training	8
network	model	input	system	function	data	learning	neural	problem	set	9
network	model	learning	function	algorithm	set	data	input	neural	training	10
network	model	learning	function	set	neural	algorithm	input	data	weight	11
network	model	input	learning	function	set	unit	algorithm	weight	data	12
network	learning	input	function	model	neural	algorithm	system	unit	neuron	13
network	input	function	system	model	neural	learning	unit	output	algorithm	14
network	model	function	input	learning	data	neural	set	algorithm	system	15
network	model	neural	input	set	data	algorithm	system	function	training	16
network	learning	model	algorithm	set	function	neural	training	input	unit	17
network	learning	model	set	training	algorithm	data	function	input	neural	18
network	learning	function	model	error	unit	training	neural	set	input	19
network	model	learning	set	function	neural	training	input	system	output	20
network	model	input	learning	neural	function	unit	set	neuron	system	21
network	model	neural	function	input	set	data	system	training	neuron	22
model	network	function	input	neuron	learning	system	neural	algorithm	data	23
network	model	function	input	neural	learning	system	unit	neuron	set	24
network	model	learning	system	input	neural	function	unit	set	output	25
network	model	learning	function	system	unit	neural	training	input	set	26
network	model	algorithm	function	input	learning	unit	output	data	set	27
network	learning	model	algorithm	weight	input	unit	function	set	neural	28
network	model	learning	function	neural	set	problem	data	algorithm	error	29

Lastly, the line chart below shows the convergence of the change in pi values. A single pi value was selected for illustration. As you can see as the model quickly reaches an optimal point after only a few iterations.



Problem 2 – Normal Distribution

Below you will see the reconstructed images of the three pictures as well as the sunset picture created with five different initializations. Implementation details can be found in the comments in the code.

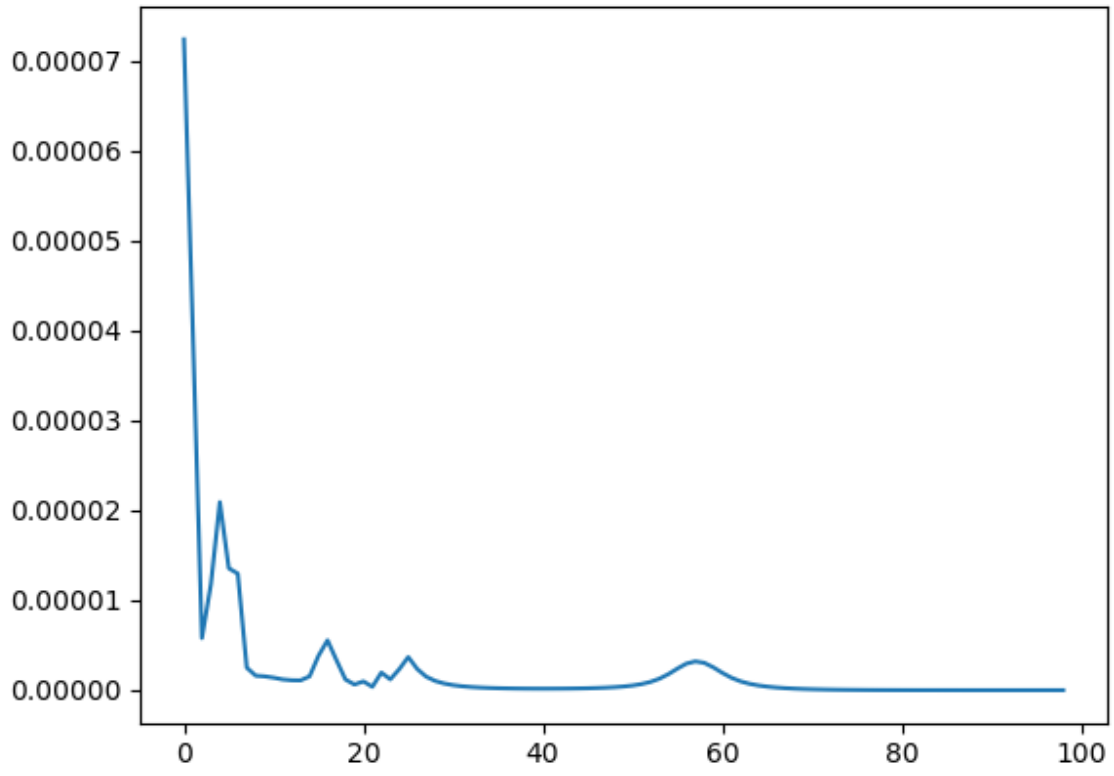
Nevertheless, KMeans was used to initialize the EM algorithm's prior probabilities and pixel distributions and different seeds were used to test different initializations.

The calculations in this model where not done in log space because the squared distance approach on page 249 in DAF's book was used to mange the small probabilities.

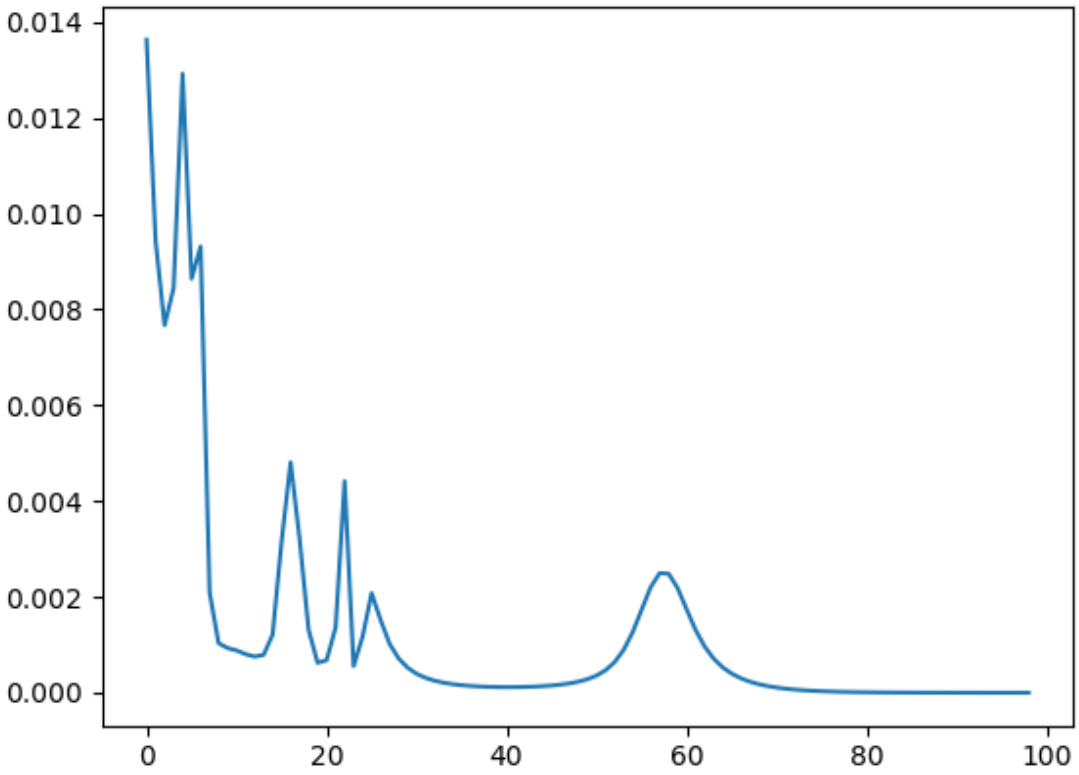
The calculations were done using matrix operations. Specifically, iterations over pixels were done via matrix operations while iterations over clusters were handled using for loops. This affected performance but the model still runs relatively quickly.

Convergence was measured by tracking pi (pixel probabilities) and u (cluster center pixel values) changes over iterations. The two charts below show these tracking. As you can see for this specific sample the image converged around 70 – 80 iterations by tracking both Pi and u values.

Pi Tracking (sunset image 10 clusters)

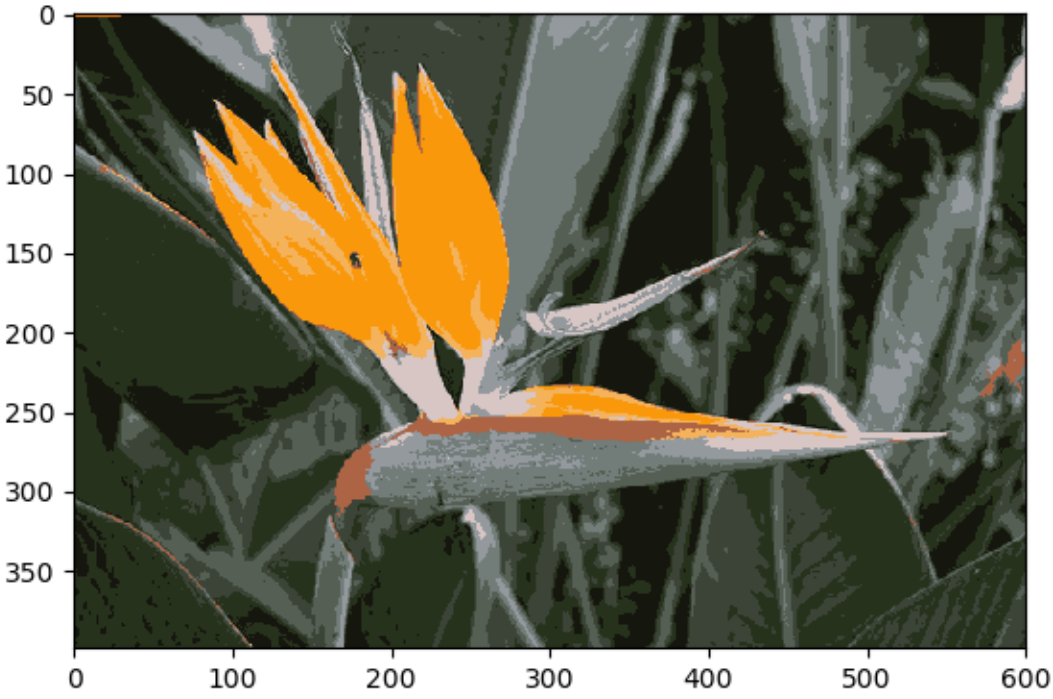


U Tracking (sunset image 10 clusters)



Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Flower 10 Clusters

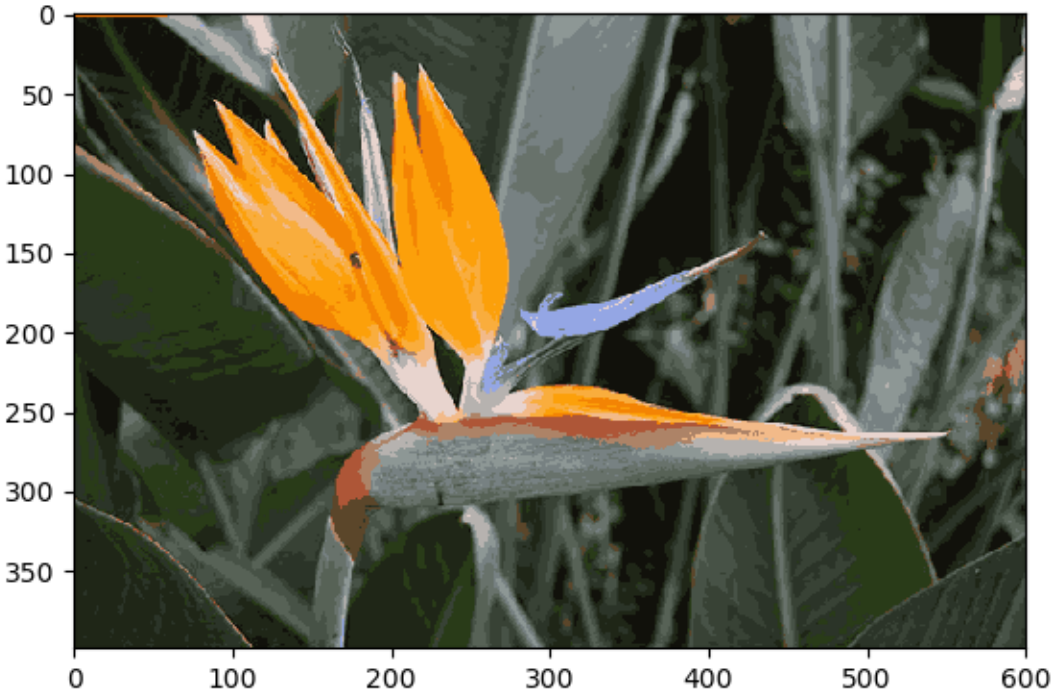


Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Flower 12 Clusters

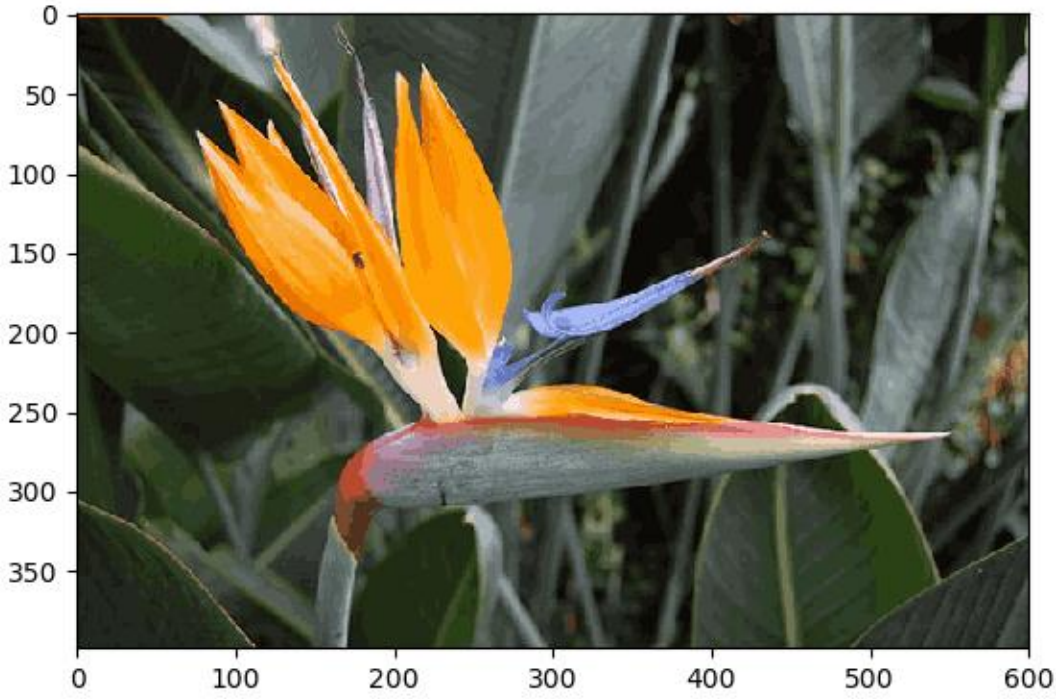
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Flower 20 Clusters



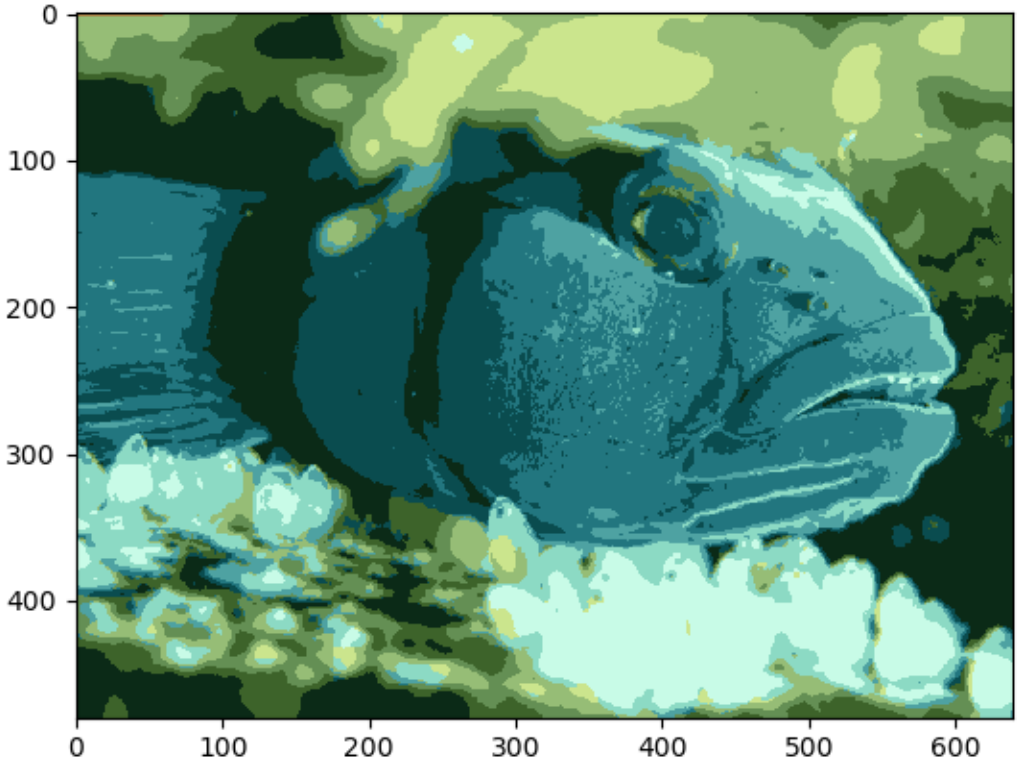
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Flower 50 Clusters



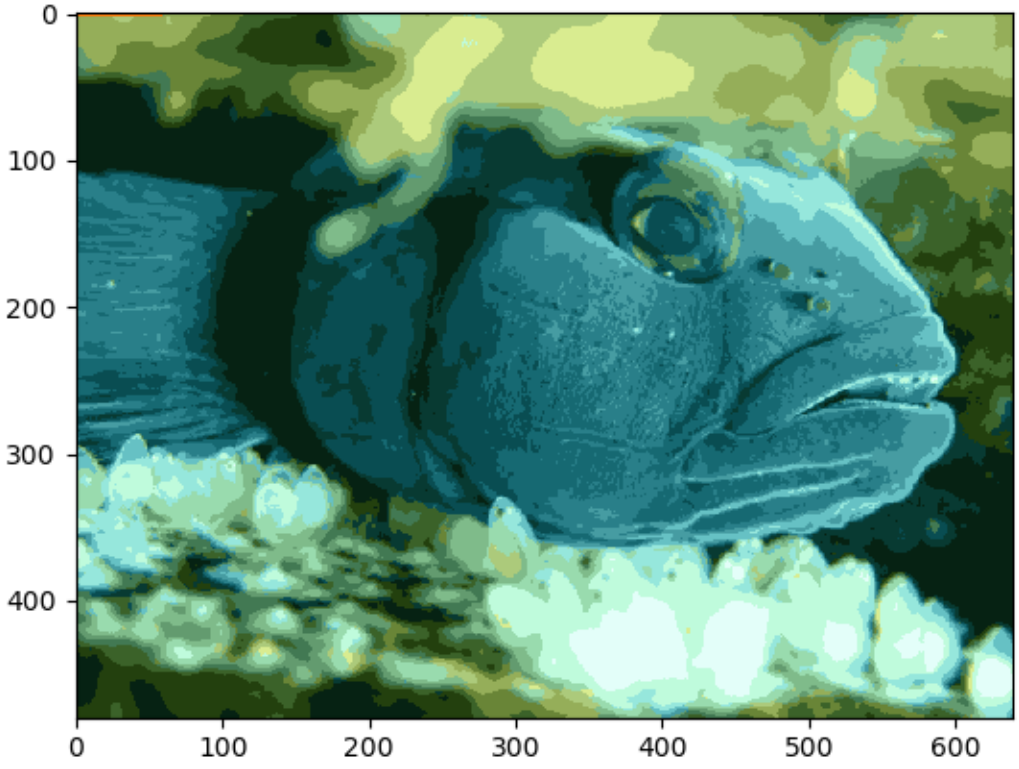
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Fish 10 Clusters



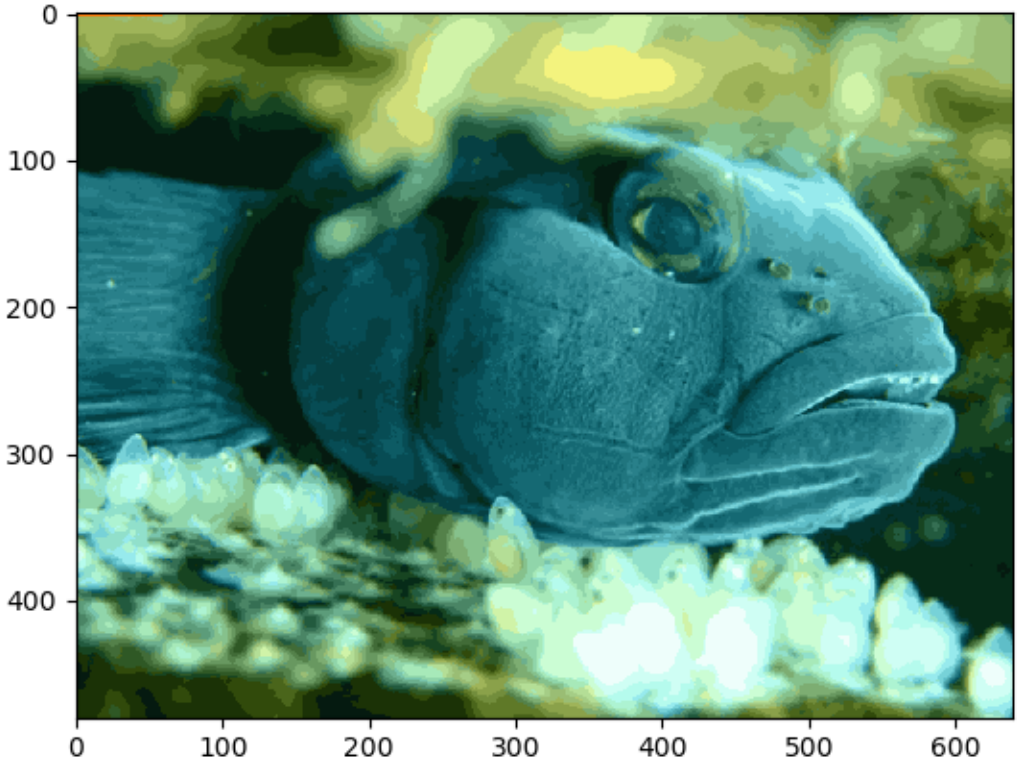
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Fish 20 Clusters



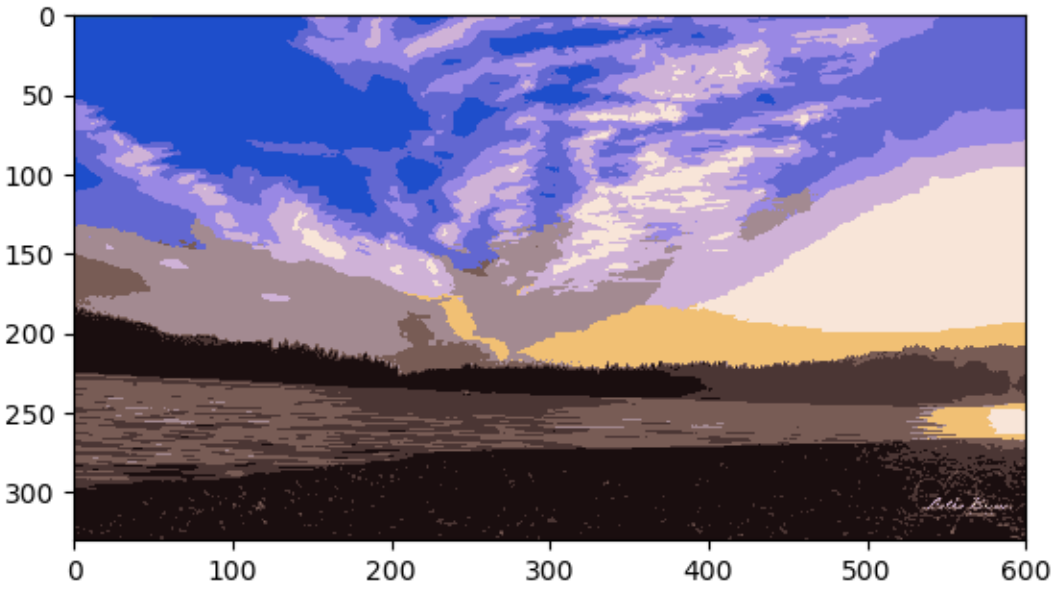
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Fish 50 Clusters



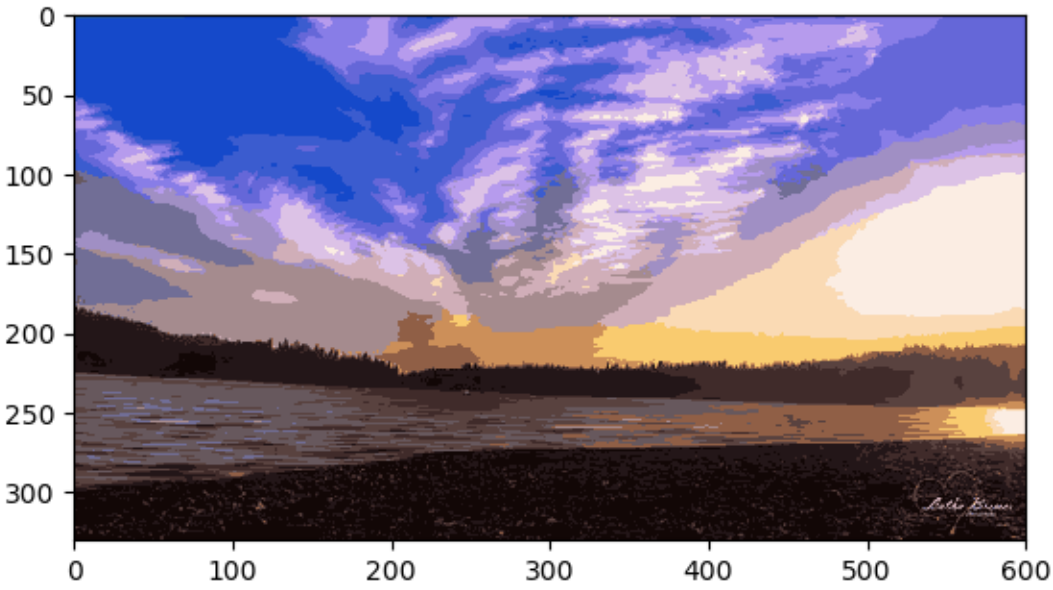
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sunset 10 Clusters



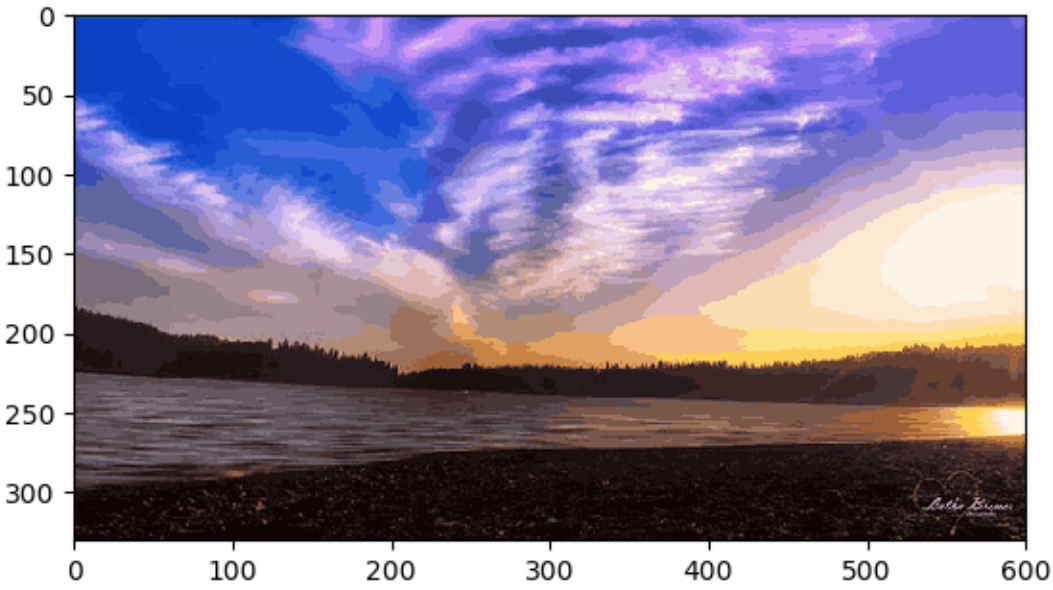
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sun set 20 Clusters



Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sunset 50 Clusters

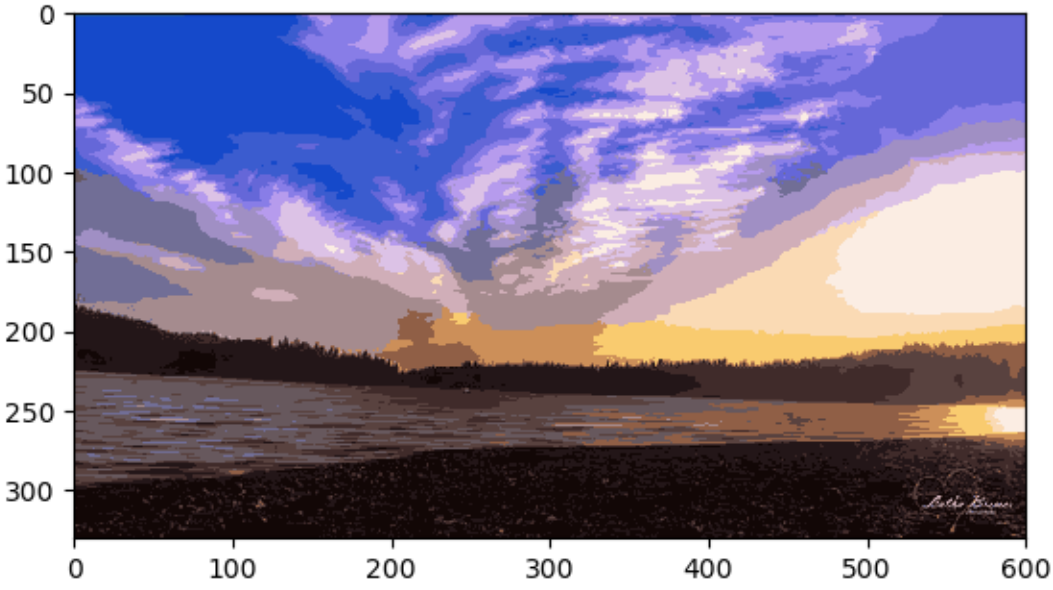


Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Problem 2 – Part 2

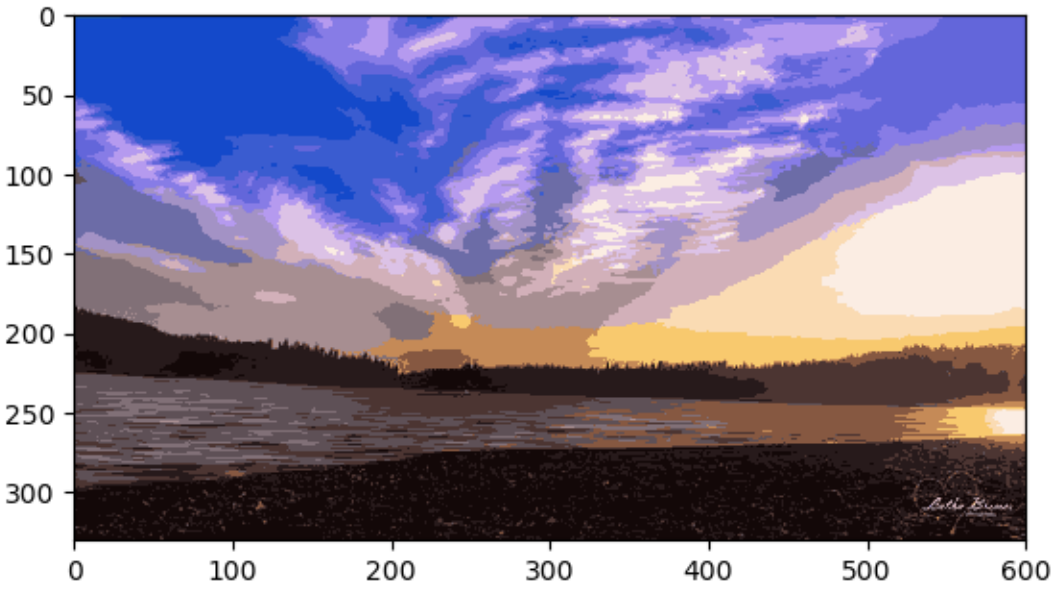
As you can see when using different starting seeds in KMeans we get slightly different images after the EM algorithm. Specifically, the base of the sunset has different clusters as well as many of the bordering pixels in the entire image are slightly shifted.

Sunset 20 Clusters Seed 1



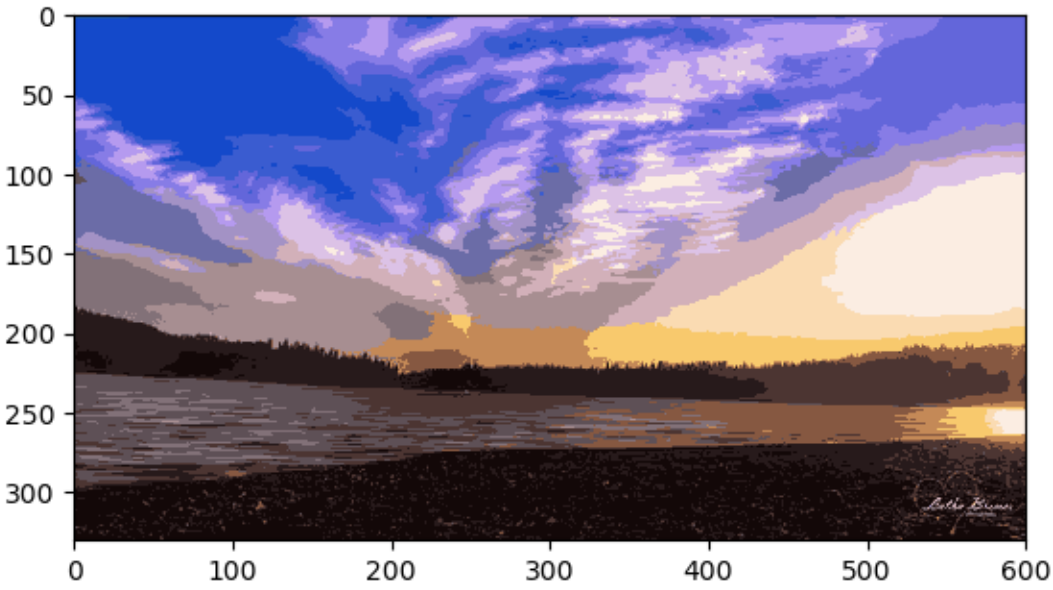
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sunset 20 Clusters Seed 2



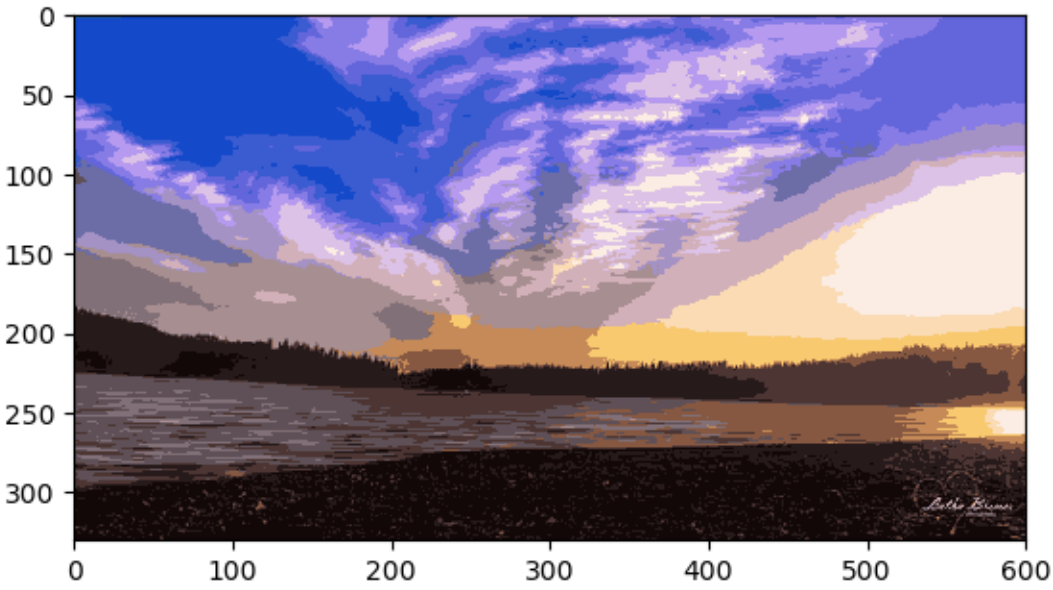
Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sunset 20 Clusters Seed 3



Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sunset 20 Clusters Seed 4



Matthew Bitter – MCS-DS
HW7 – April 8th – CS498

Sunset 20 Clusters Seed 5