Homework #6

Yongjin Shin
20090488
Industrial Management Engineering, POSTECH
dreimer@postech.ac.kr

September 3, 2018

Problem 1. Describe k-means and implement it. Apply it to image segmentation, evaluating your own code for at least three different values of k (e.g., k = 2, 3, 5).

Solution 1. Description

For the problem of data clustering, suppose our data set is composed of N data, $x_1, ..., x_N$, such that each data is in $\mathbb{R}^D$. The goal of clustering is to divide data set into K number of groups, where K is given. It is easy to consider that we need to make a group of several data points which are similar with each others. Also, we may seek the mean values μ_k of each data groups. Therefore, our objective is to find the mean values μ_k such that it makes distances between each groups' elements should be the smallest. Thus, since data is in Euclidean space so that it can be formulated with Euclidean distance form as follows:

$$\mathcal{J} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{n,k}||x_n - \mu_k||^2 \tag{1}$$

where, $r_{n,k}$ is about membership of the $k-th$ group, which means that if data X_n is in the group of k, then it will be assigned as 1, otherwise it should be 0.

First to think about is $r_{n,k}$. If we have n fixed, we only need to consider kth changes. Then we can see that x can make $\mathcal{J}$ smaller when it is assigned into the closest k cluster containing μ_k. From this result, we can decide $r_{n,k}$ as follows:

$$r_{n,k} = 1 \text{ if } k = argmin_j||x_n - \mu_j||^2 \qquad r_{n,k} = 0 \text{ otherwise.} \tag{2}$$

From our objective function, we should figure out two kinds of values, one is $r_{n,k}$ and the other is μ_k. Suppose $r_{n,k}$ has fixed. Then the objective function $\mathcal{J}$ is a quadratic function of μ_k so that it can be minimized by setting its derivative with respect to μ_k to zero:

$$\frac{\partial \mathcal{J}}{\partial \mu_k} = 2\sum_{n=1}^{N} r_{n,k}(x_n - \mu_k) = 0 \tag{3}$$

Hence, μ_k is to give

$$\mu_k = \frac{\sum_n r_{n,k}x_n}{\sum_n r_{n,k}} \tag{4}$$

Thus with given two results, we can say that there are two phase to optimize the objective function. Since two phases reduces the value of $\mathcal{J}$, convergence is assured. Therefore, the following algorithm can be derived:

Algorithm 1: K-means Clustering

input : Dataset $\mathcal{D} = \{x_1, ..., x_N\}$
output: $\mu_k, r_{n,k}$ s.t $k = \{1, ..., K\}$, $n = \{1, ..., N\}$
Initialize μ_k with a random subset of K data points
while *convergence or iteration* $<$ *max_iteration* **do**
  $r_{n,k} = 1$ if $k = argmin_j ||x_n - \mu_j||^2$
  $\mu_k = \frac{\sum_n r_{n,k} x_n}{\sum_n r_{n,k}}$
end
Return μ_k and $r_{n,k}$

2. Method

As k-means algorithm is unsupervised method, we need to look up the results of implementation and judge whether it works well or not. Therefore the result should be easy to interpret and sensible to us. In this reason CIELAB colour space was deployed as it CIELAB was designed to be perceptually uniform with respect to human color vision[2].

MATLAB provided *rgb2lab* function so that RGB information was transformed into LAB information by *rgb2lab*, however, the dimension of data was still same as 3. After transformation, clusters was figured out with the algorithm. Note that 4 different Ks are, 2,3,5,7.

3. Results and Discussion

k=2	L	a	b
C1	51.5486	1.1319	-7.1723
C2	11.395	-5.0829	4.9602
k=3	L	a	b
C1	54.3418	1.406	-5.8459
C2	10.739	-5.1441	5.0882
C3	43.7303	0.3494	-10.7666
k=5	L	a	b
C1	51.0917	1.0861	-7.3394
C2	8.5679	-5.1682	5.2276
C3	30.7169	-3.8615	2.4362
C4	57.3952	1.7116	-4.4314
C5	41.8233	0.2666	-12.4499
k=7	L	a	b
C1	51.1083	1.0935	-7.3567
C2	41.8334	0.2728	-12.4933
C3	37.0892	-2.689	0.508
C4	4.799	-3.1099	2.5803
C5	10.6786	-6.9648	7.6448
C6	22.6348	-5.1249	4.3479
C7	57.3954	1.7116	-4.4309

Plane

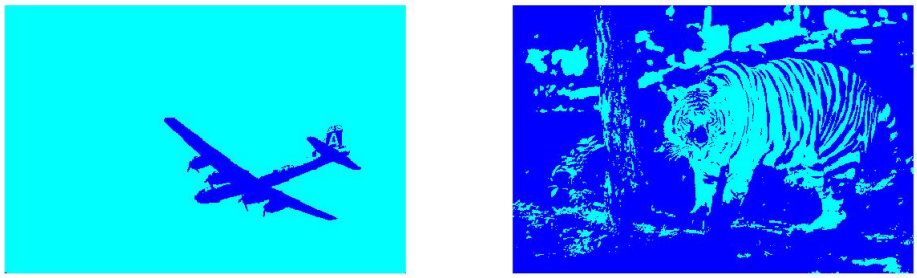
k=2	L	a	b
C1	26.255	-6.649	6.6941
C2	55.499	-0.9159	19.2957
k=3	L	a	b
C1	67.2741	0.7545	21.655
C2	40.3213	-4.7994	14.2335
C3	22.9522	-6.5926	4.9633
k=5	L	a	b
C1	32.6418	-5.674	8.5835
C2	42.3429	-17.3604	19.5652
C3	79.8929	-2.3546	21.3452
C4	49.8176	4.0334	18.853
C5	19.8796	-6.4313	3.7745
k=7	L	a	b
C1	26.2638	-6.9402	8.8463
C2	18.0405	-6.1644	1.9483
C3	82.4837	-3.0951	21.5109
C4	37.104	-6.7073	5.1518
C5	55.6549	4.6389	20.1412
C6	39.3144	2.3614	16.7303
C7	42.8506	-17.6778	19.9021

Tiger

Figure 1: Mean values of Each Clusters

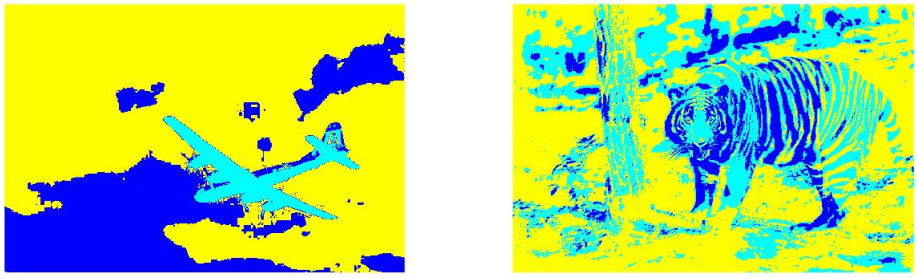
The result of segmentation is given below. Implementation was very easy and execution time was also quite fast to all Ks. If we notice that L* for the lightness and a* and b* for the greenred and blueyellow color components, the mean values (figure 1) show some representative aspects of each clusters. For example, every cases of K, lighter color ($L > 51$) exits. And when we increased K, the lighter cluster might be divided

into depending on other colours properties (a*, b*). It can be shown that cluster 1 of $K = 2$ is divided into cluster 1 and cluster 3 of $K = 3$. Actually, this reveals that sky are parted from cloud area. Which is also taken place in 'tiger' image. The last thing to mention is after $K = 5$, the increasing K only effects to divide local area so that the overall structure of segmentation is not improved but rather it looks fuzzy.



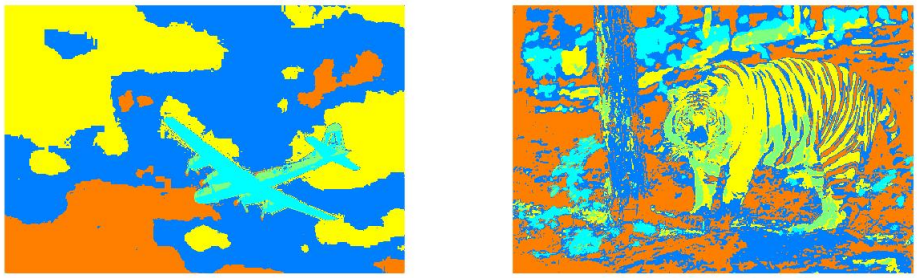
(a) Plane (b) Tiger

Figure 2: K-means: $K = 2$



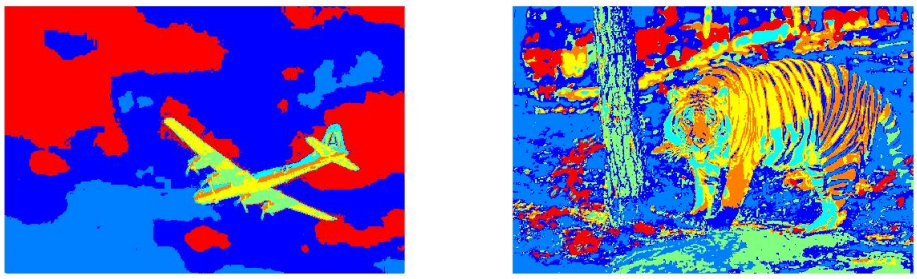
(a) Plane (b) Tiger

Figure 3: K-means: $K = 3$



(a) Plane (b) Tiger

Figure 4: K-means: $K = 5$



(a) Plane (b) Tiger

Figure 5: K-means: $K = 7$

Problem 2. Describe MoG and implement it. Apply it to image segmentation, evaluating your own code for at least three different values of k (e.g., k = 2, 3, 5).

Solution 1. Description

Unlike K-means algorithm which assigns each data points to only one closest cluster, in real world, it is not easy to divide data set exactly into K groups mutually exclusively. Mixture model considers things are composed of several distributions at the same time so that even though the probability density function can not be able to be achieved with certain probability model, however, it consists of a few some probability density functions. In this sense, mixture of Gaussian is the model that several different Gaussian distributions are overlapped containing different mean and covariance respectivly. Then we can define linearly combinationed Gaussian function as follows:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{5}$$

where each Gausian density $\mathcal{N}(\mu_k, \sigma_k)$ is called a component of the mixture and the parameter π_k is called mixing coefficients. If we integrate both side of equation,

$$\int p(x)dx = \int \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)dx = 1 \tag{6}$$

Since each component $\mathcal{N}(\mu_k, \sigma_k)$ and p(x) are normalized, the sum of mixing coefficients should be 1. Also $p(x) \geq 0$ and $\mathcal{N}(x|\mu_k, \sigma_k) \geq 0$ which implies that $\pi_k \geq 0$ for all k. Therefore we can think of π_k as a probability. Then we can say this by slightly different way:

$$p(x) = \sum_{k=1}^{K} p(k)p(x_n|k) \tag{7}$$

where the same with equation except $p(k)$ represents π_k, which is the prior probability of picking the k^{th} component and $p(x|k) = \mathcal{N}(\mu_k, \sigma_k)$ as the probability of x conditioned on k.

Then let's introduce some latent variable r which is composed of K elements $r \in \mathbb{R}^K$ so that a particular element is equal to 1 and others all zero. Therefore $z_k \in \{0, 1\}$ and $\sigma_k z_k = 1$. Now we can define the joint distribution $p(x|z)$ in terms of marginal distribution p(z) and a conditional distribution $p(x|z)$. The marginal distribution over z can be given as:

$$\int p(x, z)dz = \int p(z)p(x|z)dz$$
$$p(z_k = 1) = \pi_k$$

In some sense, z_k represents that the corresponding x_n to z_k is included in k^{th} group, however, it doesn't need to be only one group so that π_k can be considered as the probability of k^{th} groups prior probability like as we defined before. Because z uses one hot encoding, we can also write the distribution as follows:

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Also, the conditional distribution of x give for the particular z_k is a Gaussian:

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)$$
$$p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}$$

Now we can see this:

$$\begin{aligned} p(x|\pi,\mu,\Sigma) &= \sum_{k=1}^{K} p(z_k=1|\pi)p(x|z_k=1) \\ &= \sum_{k=1}^{K}\Big[\prod_{k=1}^{K}\pi_k^{z_k}\prod_{k=1}^{K}\mathcal{N}(x|\mu_k,\Sigma_k)^{z_k}\Big] \\ &= \sum_{k=1}^{K}\pi_k\mathcal{N}(x|\mu_k,\Sigma_k) \end{aligned}$$

Before solving the maximization problem, we need to define another term called **responsibility**, such that component k takes for explaining the observation x. We see π_k as the prior probability of $z_k = 1$ and this responsibility variable is kind of posterior probability of observed x. Simply we can see how much the particular Gaussian explain the observed value x. In other words, observed value x is composed of several different Gaussian containing its own responsibilities summed up to 1. The responsibilities can be defined as follows:

$$r_{k,n} = \frac{\pi_k\mathcal{N}(x|\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j\mathcal{N}(x|\mu_j,\Sigma_j)} \tag{8}$$

Let X be $N \times D$ matrix and the latent variable Z is $N \times K$ matrix. If we assume data set is i.i.d, then log-likelihood is given by:

$$\begin{aligned} \mathcal{L} &= \log p(X|\pi,\mu,\Sigma) \\ &= \sum_{n=1}^{N}\log p(x_n|\pi,\mu,\Sigma) \\ &= \sum_{n=1}^{N}\log\Big(\sum_{k=1}^{K}\pi_k\mathcal{N}(x|\mu_k,\Sigma_k)\Big) \end{aligned}$$

Maximum likelihood estimates of π_k, μ_k, Σ_k are computed as follows:
(1) μ_k

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mu_k} &= -\sum_{n=1}^{N}\Big(\frac{\pi_k\mathcal{N}(x|\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j\mathcal{N}(x|\mu_j,\Sigma_j)}\Big)\Sigma_k^{-1}(x_n-\mu_k) \\ &= -\sum_{n=1}^{N}r_{k,n}\Sigma_k^{-1}(x_n-\mu_k) \\ &= 0 \end{aligned}$$

The last term of first equality can be achieved with using *equation (81)* in [3].

$$\frac{\partial f}{\partial\mu} = -\frac{1}{2}\big(\Sigma^{-1}+(\Sigma^{-1})^T\big)\times(x-\mu)\times(-1) \tag{9}$$

Since covariance matrix is symmetric, $\Sigma^{-1} = (\Sigma^{-1})^T$. Thus we can see that $\frac{\partial f}{\partial\mu} = \Sigma^{-1}(x-\mu)$. Therefore,

$$\mu_k = \frac{\sum_{n=1}^{N}r_{k,n}x_n}{\sum_{n=1}^{N}r_{k,n}} \tag{10}$$

(2) Σ_k

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Sigma_k^{-1}} &= -\sum_{n=1}^{N}\Big(\frac{\pi_k \mathcal{N}(x|\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j \mathcal{N}(x|\mu_j,\Sigma_j)}\Big)\frac{1}{2}\big(\Sigma_k - (x_n-\mu_k)(x_n-\mu_k)^T\big) \\ &= -\sum_{n=1}^{N} r_{k,n}\big(\Sigma k - (x_n-\mu_k)(x_n-\mu_k)^T\big) \\ &= 0\end{aligned}$$

The last term of first equality can be derived as follows:

$$\begin{aligned}\frac{\partial \log(\mathcal{N})}{\partial \Sigma^{-1}} &= \frac{\partial\big(-\frac{k}{2}\log 2\pi - \frac{1}{2}(\log|\Sigma| + (x-\mu)^T\Sigma^{-1}(x-\mu))\big)}{\partial \Sigma^{-1}} \\ &= -\frac{1}{2}\Big[\frac{\log|\Sigma|}{\partial \Sigma^{-1}} + \frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{\partial \Sigma^{-1}}\Big]\end{aligned} \tag{11}$$

Then with using *equation (37)* in [3],

$$\begin{aligned}\frac{\partial \Sigma\Sigma^{-1}}{\partial \Sigma^{-1}} &= \Sigma\frac{\partial \Sigma^{-1}}{\partial \Sigma^{-1}} + \Sigma^{-1}\frac{\partial \Sigma}{\partial \Sigma^{-1}} \\ &= \Sigma + \Sigma^{-1}\frac{\partial \Sigma}{\partial \Sigma^{-1}} \\ &= \frac{\partial I}{\partial \Sigma^{-1}} \\ &= 0\end{aligned} \tag{12}$$

Therefore we can have:

$$\frac{\partial \Sigma}{\partial \Sigma^{-1}} = -\Sigma^2 \tag{13}$$

If we use *equation (57) and (61)* in [3] and the equation (13), the equation (11) will be:

$$\begin{aligned}\frac{\log|\Sigma|}{\partial \Sigma^{-1}} + \frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{\partial \Sigma^{-1}} &= \Big(\frac{\partial \log|\Sigma|}{\partial \Sigma} \times \frac{\partial \Sigma}{\partial \Sigma^{-1}}\Big) + (x-\mu)(x-\mu)^T \\ &= ((\Sigma^T)^{-1} \times -\Sigma^2) + (x-\mu)(x-\mu)^T \\ &= -\Sigma + (x-\mu)(x-\mu)^T\end{aligned} \tag{14}$$

Therefore, we can conclude:

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{k,n}(x_n-\mu_k)(x_n-\mu_k)^T \tag{15}$$

(3) π_k

This can be regarded as constrain optimization problem since $\Sigma_{k=1}^{K}\pi_k = 1$. So apply Lagrange multiplier(λ) with the objective function then:

$$\mathcal{L}'(\pi,\lambda) = \mathcal{L} + \lambda[1 - \Sigma_{k=1}^{K}\pi_k] \tag{16}$$

Therefore, $\nabla_{\pi_k,\lambda}\mathcal{L}'(\pi,\lambda) = 0$ needs to be calculated:

$$\frac{\partial \mathcal{L}'(\pi,\lambda)}{\partial \lambda} = 1 - \Sigma_{k=1}^{K}\pi_k = 0 \tag{17}$$

$$\frac{\partial \mathcal{L}'(\pi, \lambda)}{\partial \pi_k} = \sum_{n=1}^{N} \left(\frac{\mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \right) - \lambda = 0 \tag{18}$$

In equation (18) if we multiply π_k on both sides:

$$\pi_k = \frac{N_k}{\lambda} \tag{19}$$

If we sum up (19) with using the constrain (or the result of equation (17)), $1 = \frac{N_1+N_2,+...+N_K}{\lambda}$. Therefore $\lambda = N$. Thus, we have:

$$\pi_k = \frac{N_k}{N} \tag{20}$$

By the equation (10), (15) and (20), the update phase will be conducted and the equation (8) can assign each responsibilities. There is a problem with mle but it will be discussed in discussion section.

Algorithm 2: Mixture of Gaussian

input : Dataset $\mathcal{D} = \{x_1, ..., x_N\}$

output: $\pi_k, \mu_k, \Sigma_k, r_{n,k}$ s.t $k = \{1, ..., K\}$, $n = \{1, ..., N\}$

Initialize μ_k with the result of K-means clustering

Initialize Σ_k with identity covariance and μ_k be uniformly ditributed.

while *convergence or iteration* $<$ *max_iteration* **do**

  $r_{k,n} = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}$

  $\pi_k = \frac{N_k}{N}$

  $\mu_k = \frac{\sum_{n=1}^{N} r_{k,n} x_n}{\sum_{n=1}^{N} r_{k,n}}$

  $\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{k,n} (x_n - \mu_k)(x_n - \mu_k)^T$

end

Return

2. Method

As K-means clustering had already been performed so that the result of K-mean algorithm was used to set the initial points of MoG. Other conditions are nearly same with K-means. Clustering was proceeded with 4 different Ks, 2,3,5,7.

3. Results and Discussion

Since π is kind of prior probabilities of observed x, so that the value is bigger than others if the clustered area is bigger especially in 'plane' image when $K = 2$, the sky has over 93.3%. However, with same conditions, 'tiger' image is separated almost same ratio, which is because the color difference is not much big. Interesting thing to see is that unlike K-means' result, the clustered image of MoG shows that different clusters also share same area, and though the colorbar implies that that areas components (clusters) have different *responsibilities*. If we carry out similar way of analysis to figure 6 like how we did in K-means, it is easily to find that when K has been increased, the large group is divided into similar color groups. If this has happened, π is also changed however not the exact ratio. It seems that if there is a change of only one group, the overall structure of mixtures are rearranged with changing μ and Σ.

Between 'Plane' and 'Tiger' images, 'tiger' shows characteristics of MoG better than the former. It could be interpreted as if the more data points are hardly divided, (assumed similar colors are massed around like in 'tiger' image), the more appropriate MoG can be used to. In other words, if the data do not have many interruption between different groups, it would be better choice to K-means since it is much easier to interpret the result.

The last thing to comment is that even though the initial point was used with the result of K-means, when K is getting larger, there were some error for computing covariance matrix Σ. This could be because of the presence of singularities. Suppose if the mean of any component of mixtures, say t^{th} is exactly same with one point of data set and the number of element assigned t^{th} group is only this particular one. As the

remaining data could be covered other Gaussians' so that this t^{th} Gaussian doesn't need to consider other data, σ_t is going to zero for trying to maximize the MLE. However, if we think about multivariate density function, the total probability goes to infinite when variance goes to zero. Therefore, there are always some chances to face this kind of situations so that it will cause computational errors.

k=2	L	a	b	π
C1	51.6486	1.1798	-7.2746	0.9335
C2	19.6071	-4.266	3.4901	0.0665
k=3	L	a	b	π
C1	42.7319	0.0995	-12.1276	0.1348
C2	53.1543	1.3565	-6.4467	0.8003
C3	18.73	-4.3427	3.6455	0.0648
k=5	L	a	b	π
C1	52.1221	1.207	-6.9452	0.6866
C2	6.9438	-5.1179	5.3451	0.0319
C3	32.3074	-3.2826	1.5011	0.036
C4	57.1107	1.8528	-4.47	0.1407
C5	41.132	0.0915	-13.2254	0.1048
k=7	L	a	b	π
C1	52.1112	1.2061	-6.9499	0.687
C2	41.0531	0.0919	-13.2896	0.1036
C3	38.9331	-2.3607	-0.0972	0.0275
C4	4.0417	-3.1914	3.1007	0.014
C5	9.1208	-7.4799	8.2682	0.011
C6	12.2802	-5.5389	5.5803	0.0161
C7	57.1107	1.8528	-4.47	0.1407

Plane

k=2	L	a	b	π
C1	25.5094	-6.4031	4.7731	0.5732
C2	45.7693	-3.3497	17.2512	0.4268
k=3	L	a	b	π
C1	57.4814	-8.8951	19.922	0.1343
C2	36.5873	-0.8226	10.5272	0.4301
C3	24.5665	-8.1528	6.6483	0.4356
k=5	L	a	b	π
C1	28.7628	-5.6693	4.1714	0.2509
C2	35.5081	-12.3531	15.032	0.1682
C3	76.2244	-3.1522	22.6788	0.0522
C4	42.6996	2.3598	15.2567	0.269
C5	21.1834	-7.9699	4.7587	0.2597
k=7	L	a	b	π
C1	25.0546	-8.1966	6.8628	0.2109
C2	15.6162	-7.0564	1.8256	0.0963
C3	79.1066	-2.2218	23.0588	0.0458
C4	28.8325	-6.2642	3.4168	0.2118
C5	50.71	4.6045	19.2276	0.1474
C6	33.1353	-0.4515	10.4484	0.1425
C7	37.468	-12.9132	16.3298	0.1454

Tiger

Figure 6: Mean values and π of Each Clusters

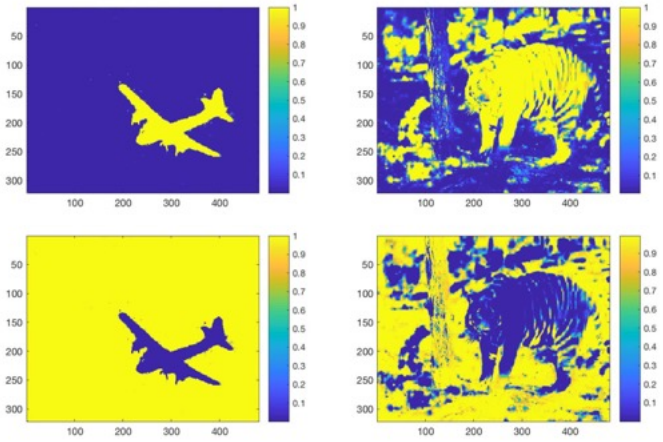


Figure 7: MoG with K=2

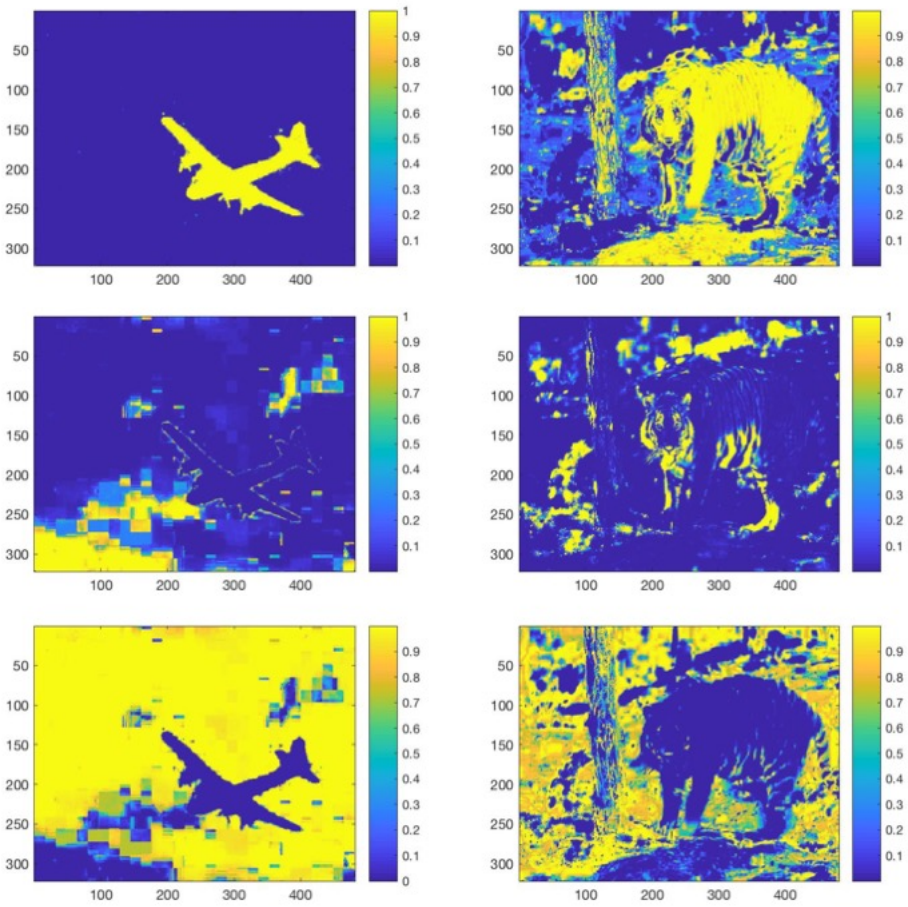


Figure 8: MoG with K=3

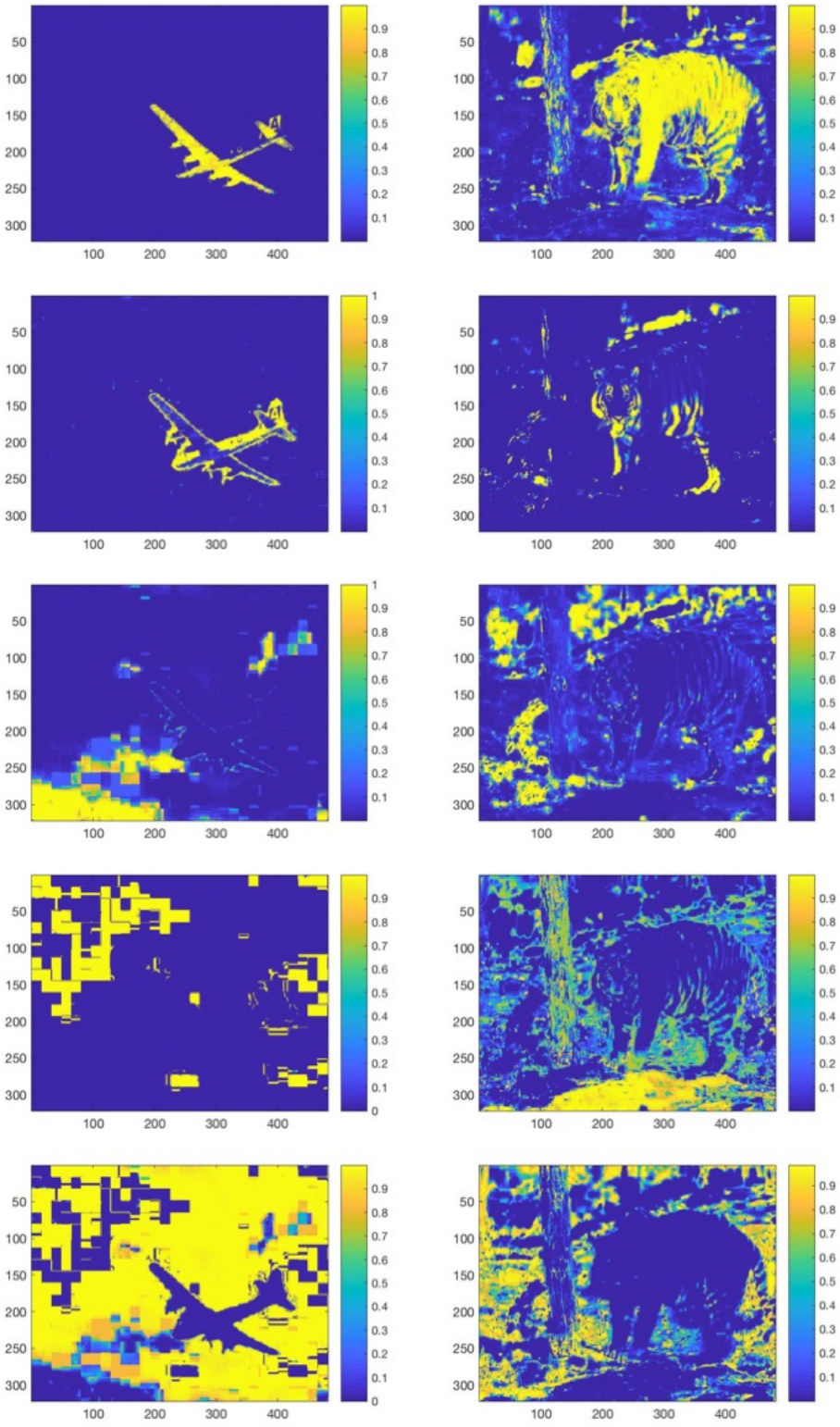


Figure 9: MoG with K=5

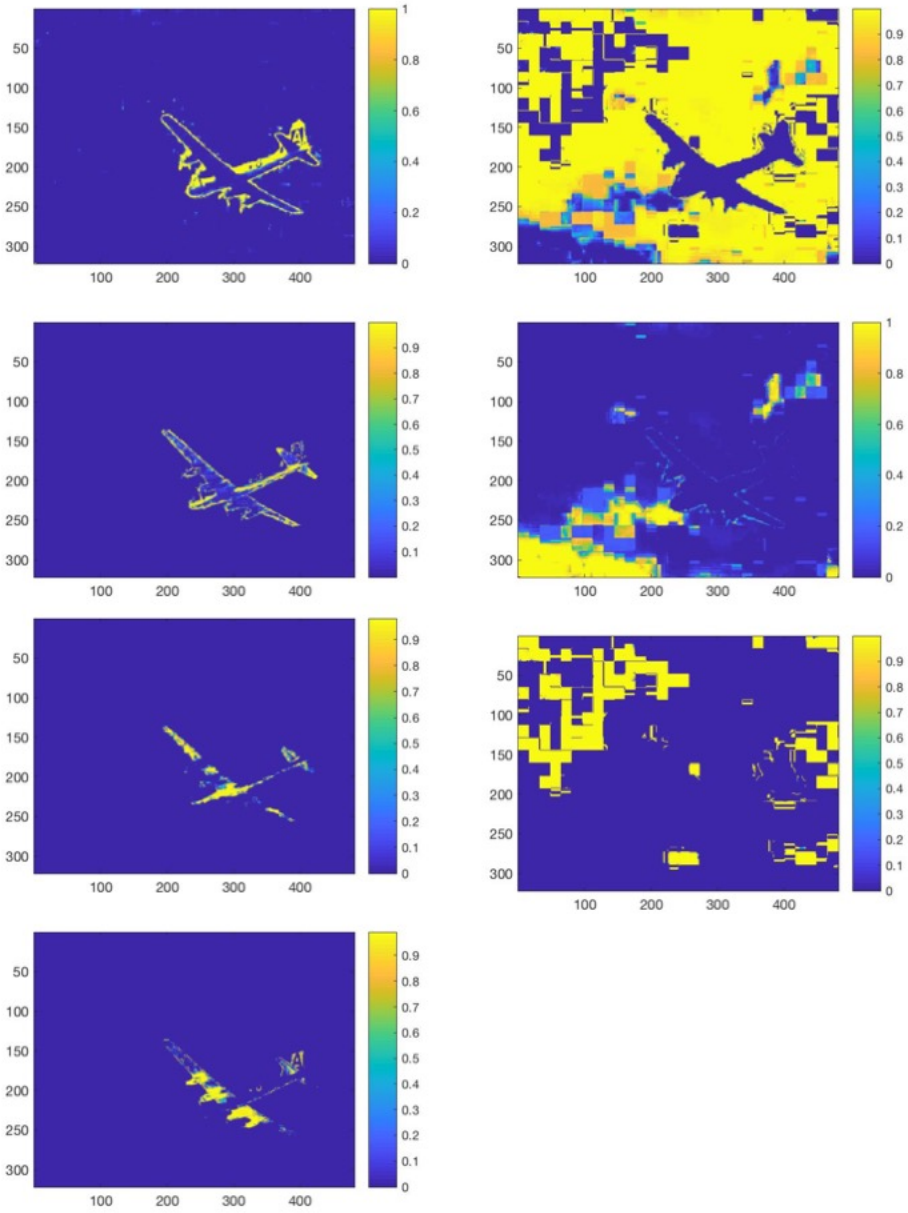


Figure 10: MoG with K=7, Plane

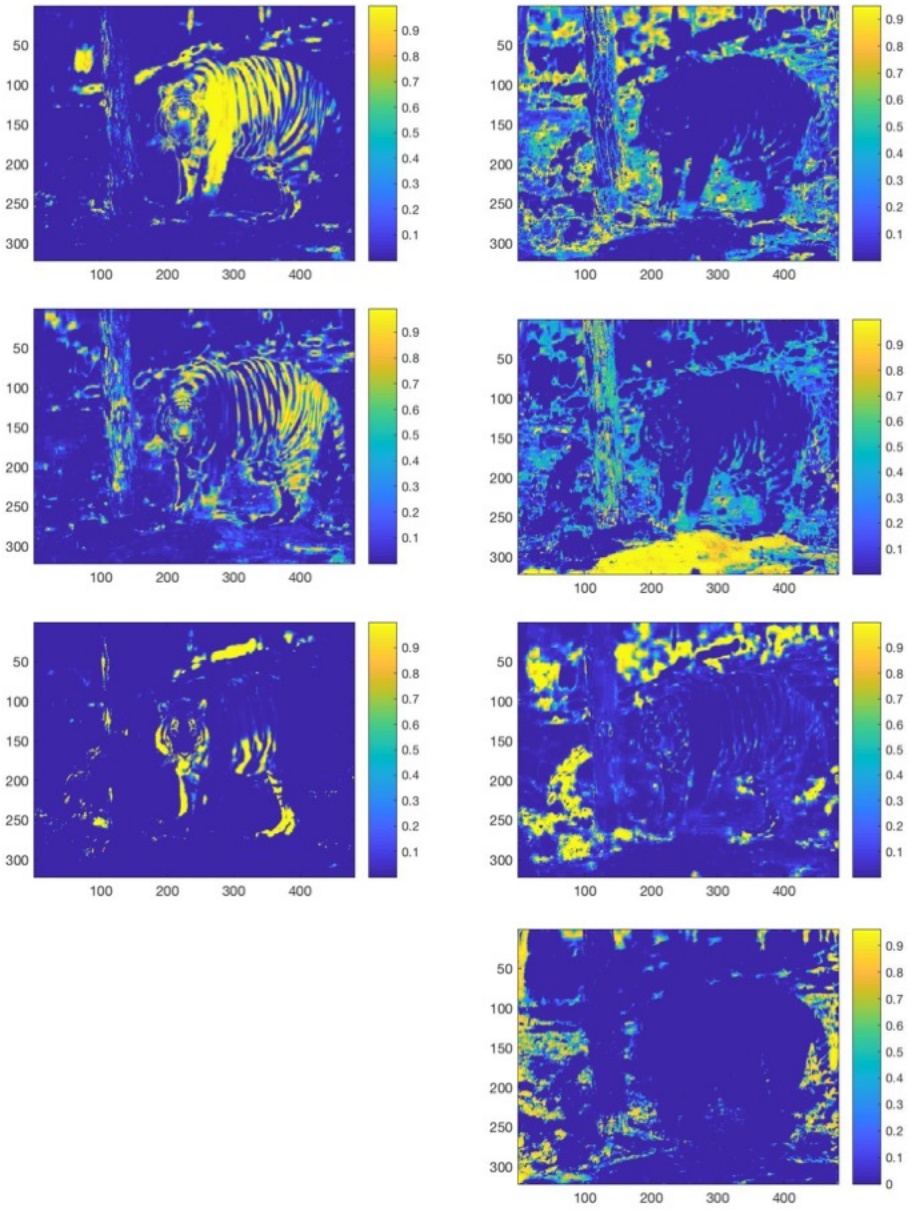


Figure 11: MoG with K=7, Tiger

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*.

[2] Wikipedia *https://en.wikipedia.org/wiki/Lab_color_space*.

[3] Kaare Brandt Petersen *The Matrix Cookbook*.