CS5340 **Name:** EDDAMANI Soufiane
Semester I, 2017/18 **ID** : A0176636U
Assignment **Name:** LANGE Robin
19/11/2017 **ID** : A0165184B

1. **Data Denoising**

(a) **Model, Local Evidence term and Pairwise Potential**

We are using Gibbs sampling for data denoising. Thus we have to model the full conditional probability. Each image is seen as a Markov Random Field. And we condition each variable on its Markov Blanket, that is its nearest neighbors.

Pairwise Potential $\psi_{st}(x_s, x_t)$:

In the case of a two-state variable, we used the **Ising model** for the pairwise potential, representing the neighboring effect. We choose $\psi_{st}(x_t, x_s) = \exp(\beta x_s x_t)$.

Local Evidence $\psi_t(x_t)$:

This part is modelling the noise. We choose a gaussian noise centered around the latent variable designing the "clear" image $\psi_t(x_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(y_t-x_t)^2}{2\sigma^2}$ where y_t is the corrupted version.

We can therefore derive the full conditional where $\mathbf{y}$ is the corrupted version and $\theta = (\sigma, \beta)$ is the parameter couple:

$$
\begin{aligned}
p(x_t = +1 \mid \mathbf{x_{-t}}, \mathbf{y}, \theta) &= \frac{\psi_t(+1) \prod_{s \in nbr(t)} \psi_{st}(+1, x_s)}{\sum_{x_t} \psi_t(x_t) \prod_{s \in nbr(t)} \psi_{st}(x_t, x_s)} \\
&= \frac{\psi_t(+1) \exp\left[\beta \sum_{s \in nbrs(t)} x_s\right]}{\psi_t(+1) \exp\left[\beta \sum_{s \in nbrs(t)} x_s\right] + \psi_t(-1) \exp\left[-\beta \sum_{s \in nbrs(t)} x_s\right]} \\
&= \frac{1}{1 + \exp\left[-2\beta \sum_{s \in nbrs(t)} x_s - \log\left(\frac{\psi_t(+1)}{\psi_t(-1)}\right)\right]} \\
&= \operatorname{sigm}\left[2\beta S_t + \log\left(\frac{\psi_t(+1)}{\psi_t(-1)}\right)\right]
\end{aligned}
$$

where $\operatorname{sigm}(x) = \frac{1}{1+\exp(-x)}$ and $S_t = \sum_{s \in nbrs(t)} x_s$.

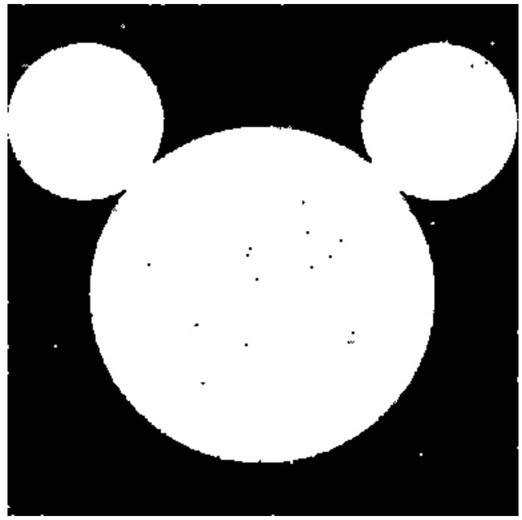
Using this probability density we can apply Gibbs sampling.

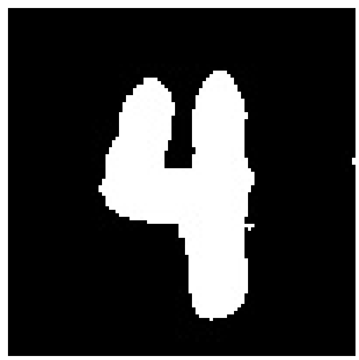
(b) **Brief Description of the Functions**

- Additionnal library used: `itertools` in order to easily generate image coordinates.
- `sigmoid(x)` and `normal(x,mu,sigma)` respectively refer to $\operatorname{sigm}(x)$ and $\mathcal{N}(y_t \mid x_t, \sigma)$ in the previous part.

- `normalize_img(image)`, `unormalize_avg_img(image_list)` and `img_to_data(image)` are pre-processing and post-processing functions to reshape data and map {0,255} to {-1,1} in order to use the model.
- `neighbors_ix(pos,image)` returns an array with the neighboring indices of a position `pos` in `image`.
- `gibbs(X0,Beta,Y,sigma,niter=10)` computes the probability calculated before for each position and samples from it. Parameters chosen (σ, β) are in $[8, 15]$ and $[3.5, 6]$ respectively depending on the image.

(c) **Output Images from the Denoising Algorithm using Gibbs Sampling**





2. **Expectation-Maximization Segmentation**

In this part we use N the number of the variables (i.e. number of pixels), K the number of clusters, and d the dimension of the used vectors and matrices (i.e. observed data, means, covariances). We have $K = 2$ for the foreground/background segmentation and $d = 3$ for Lab-colors.

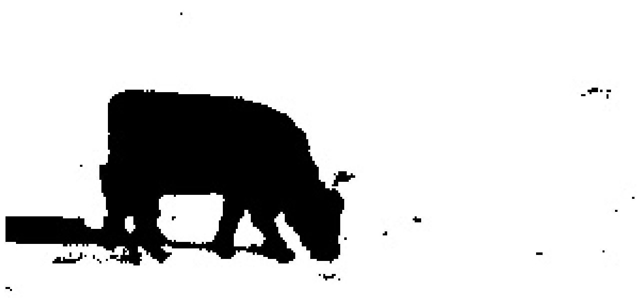
(a) **K-Means**:
We used K-Means to provide $(\boldsymbol{\mu}_k)_{k\in\{0,1\}}$ values for the EM algorithm to converge faster:

- `Astep` We compute $\mathbf{R}$ which is a $N \times K$ matrix with:
$$r_{nk} = \mathbb{1}(k = \text{argmin}_j \mid\mid \mathbf{x}_n - \boldsymbol{\mu}_j \mid\mid^2)$$
- `Ustep` We use the previously computed matrix to update the means computing:
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$
- We repeat the steps above until the convergence. That is, until all the components of $\mid \boldsymbol{\mu}_k^{t+1} - \boldsymbol{\mu}_k^t \mid$ for all k are below $\mathcal{E}_{KMeans}$, which was set to 10^{-5}.

(b) **Expectation-Maximization**
We initialise $(\boldsymbol{\mu}_k)_{k\in\{0,1\}}$ with the output values of the K-Means algorithm, each π_k to $1/K$ and each $\boldsymbol{\Sigma}_k$ to the identity matrix.

- `Estep` We compute $\boldsymbol{\Gamma}$ which is a $N \times K$ matrix with:
$$\gamma_{nk}^{old} = \frac{\pi_k^{old}\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{old}, \boldsymbol{\Sigma}_k^{old})}{\sum_k \pi_k^{old}\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{old}, \boldsymbol{\Sigma}_k^{old})}$$
- `Mstep` We maximise the expectation of the log-likelihood:
$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}^{old} \left[\log \pi_k + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]$$
To do so we compute:
$$\pi_k = \frac{N_k}{N}$$
$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma_{nk}^{old}\mathbf{x}_n$$
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma_{nk}^{old}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top$$
with $N_k = \sum_{n=1}^{N} \gamma_{nk}^{old}$
- We repeat the steps above until the convergence. That is, until $\frac{|\mathcal{Q}^{t+1}(\boldsymbol{\theta},\boldsymbol{\theta}^{old}) - \mathcal{Q}^t(\boldsymbol{\theta},\boldsymbol{\theta}^{old})|}{|\mathcal{Q}^t(\boldsymbol{\theta},\boldsymbol{\theta}^{old})|}$ is below $\mathcal{E}_{EM}$, which was set in the range $[10^{-5}, 10^{-2}]$.

(c) **Output Images from EM Segmentation Algorithm**



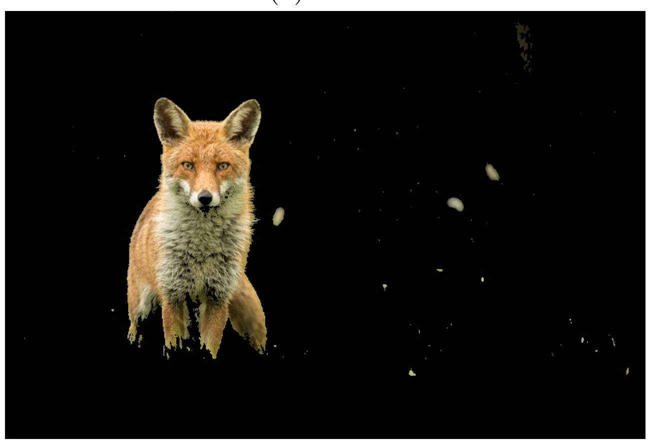
(a) Mask

(b) Foreground

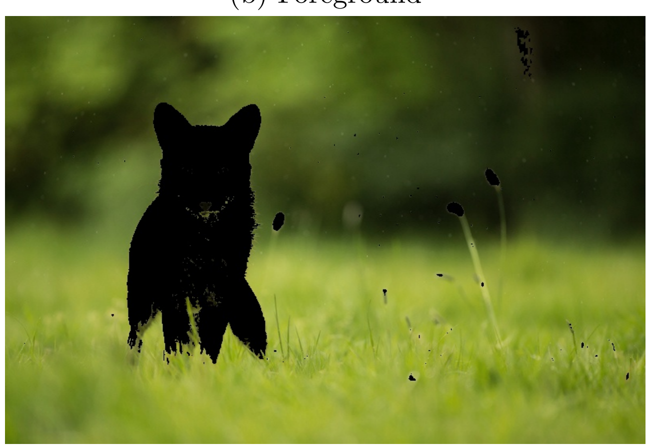
(c) Background

Figure 1: Output for cow.jpg

(a) Mask

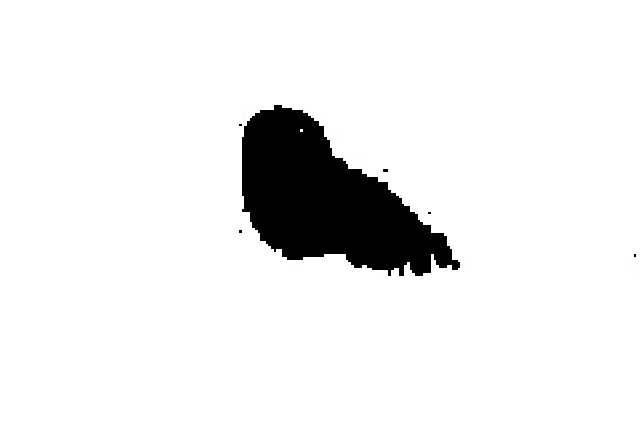


(b) Foreground



(c) Background

Figure 2: Output for fox.jpg

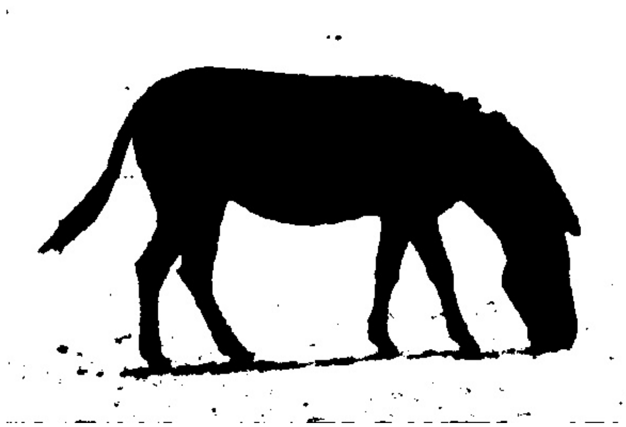


(a) Mask

(b) Foreground

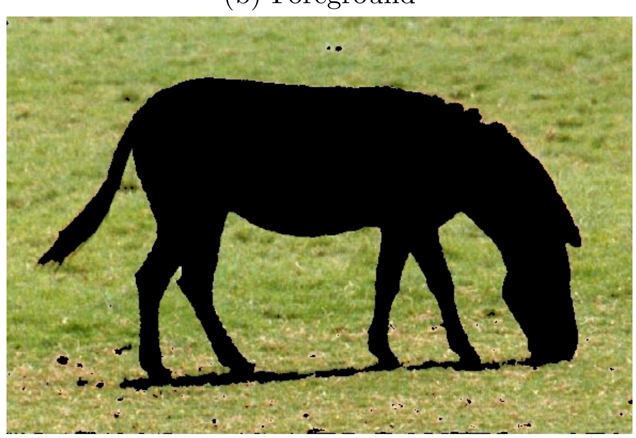
(c) Background

Figure 3: Output for owl.jpg



(a) Mask

(b) Foreground



(c) Background

Figure 4: Output for zebra.jpg