Computer Vision Assignment 3

Avani Gupta (2019121004) **Date: 19th March**

Contents

- **Grabcut**
- **MRF**

1. GrabCut

Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n:\, n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

User editing

- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.

Reference: Rother, C., Kolmogorov, V., & Blake, A. (2004). " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, *23*(3), 309-314.

2. Examples on Images

Ceramic

gamma= 50, iterations = 3, components = 5

original image

user marks bg

grabcut result

user selects fg(white), bg(black)

final output

2.1 Ablations

2.1.1 Increasing gamma

gamma= 250, iterations = 3, components = 5

Observation

As the value of Gamma is increased, the 2nd term of the Gibbs energy increases leading to a high penalty for cutting the edges. This results in the neighboring pixels having similar intensity levels not being cut.

2.1.2 Decreasing components

gamma= 50, iterations = 3, components = 1

2.1.3 Increasing components

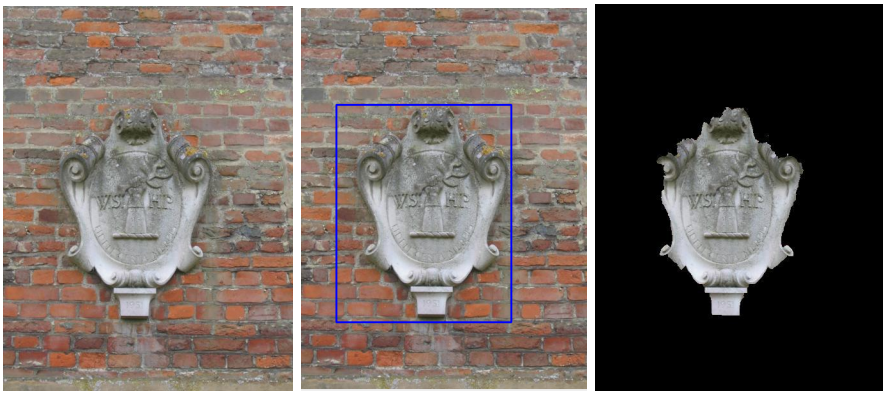
gamma= 50, iterations = 3, components = 10

2.1.4 Increasing components, Decreasing gamma

gamma= 20, iterations = 3, components = 10

memorial

gamma= 50, iterations = 5, components = 5



gamma= 250, iterations = 5, components = 5

gamma= 50, iterations = 5, components = 10

Observation: When the number of GMM components is increased, more color pixels are clustered in foreground and background regions resulting in the probability of the pixel belonging to foreground or background getting more accurate.

gamma= 20, iterations = 5, components = 10

gamma= 50, iterations = 5, components = 1

Banana1

gamma= 50, iterations = 5, components = 5

gamma = 250, iterations = 5, components = 5

gamma = 50, iterations = 5, components = 10

gamma = 50, iterations = 5, components = 1

Banana3

gamma = 50, iterations = 5, components = 5

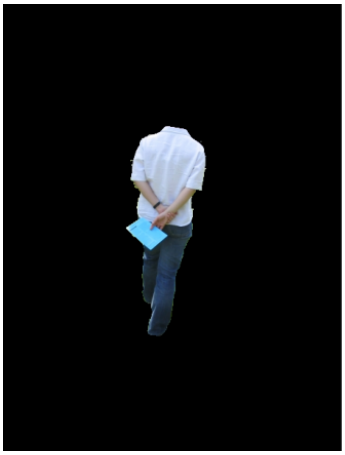
gamma = 250, iterations = 5, components = 5

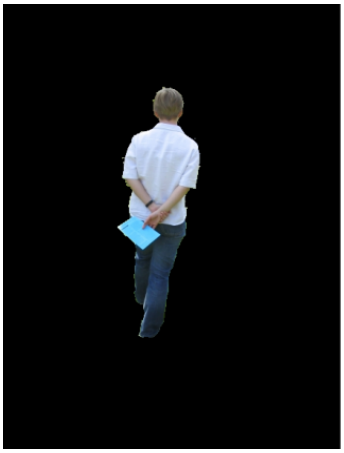
gamma = 50, iterations = 5, components = 10

gamma = 50, iterations = 5, components = 1

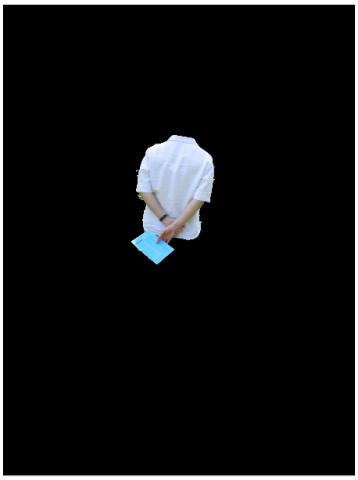
Person7

gamma = 50, iterations = 5, components = 5

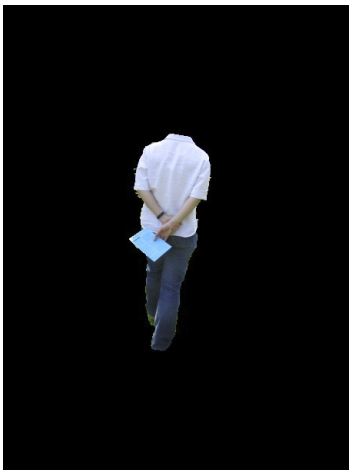




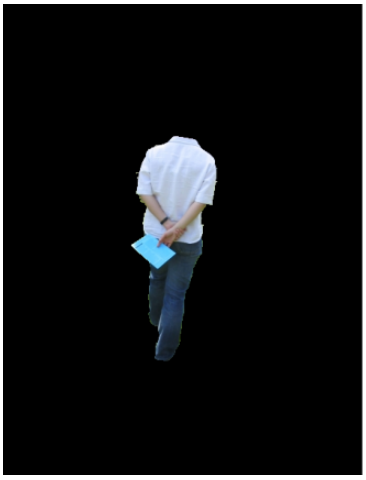
gamma = 250, iterations = 5, components = 5



gamma = 50, iterations = 5, components = 10

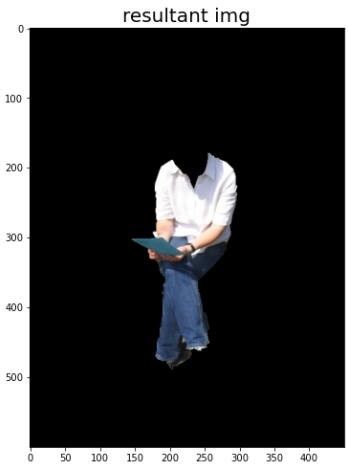


gamma = 50, iterations = 5, components = 1

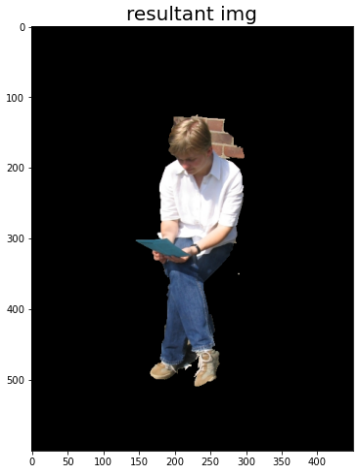


Person6

gamma = 50, iterations = 5, components = 5

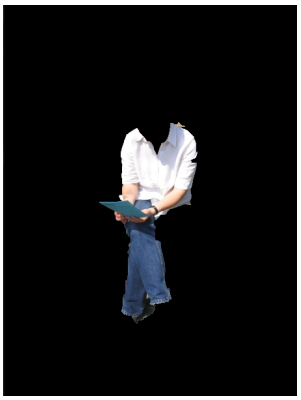


After grabcut

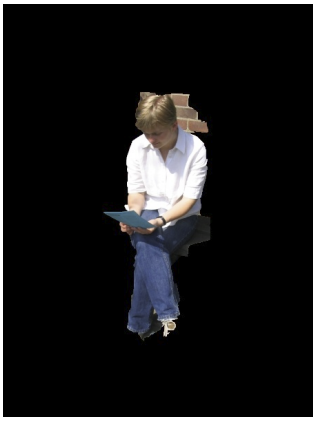


After user-interaction

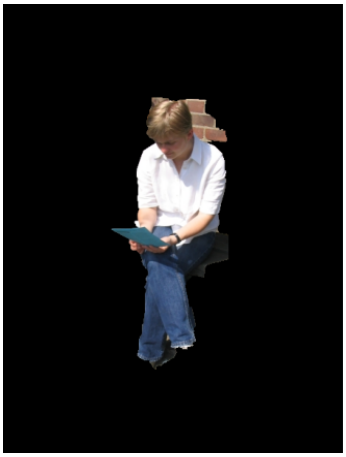
gamma = 250, iterations = 5, components = 5



gamma = 250, iterations = 5, components = 10



gamma = 50, iterations = 5, components = 1



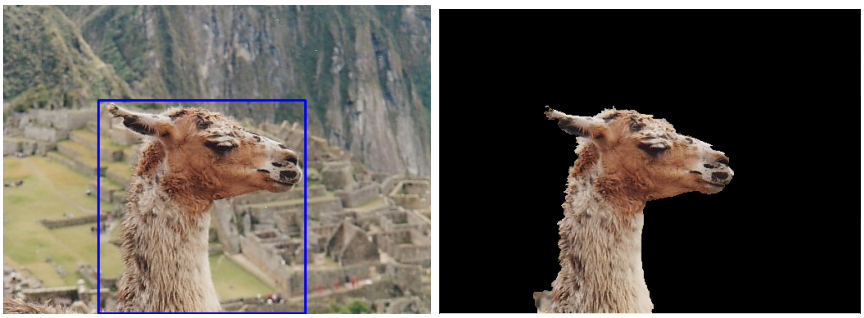
Other images

gamma = 50, iterations = 5, components = 5

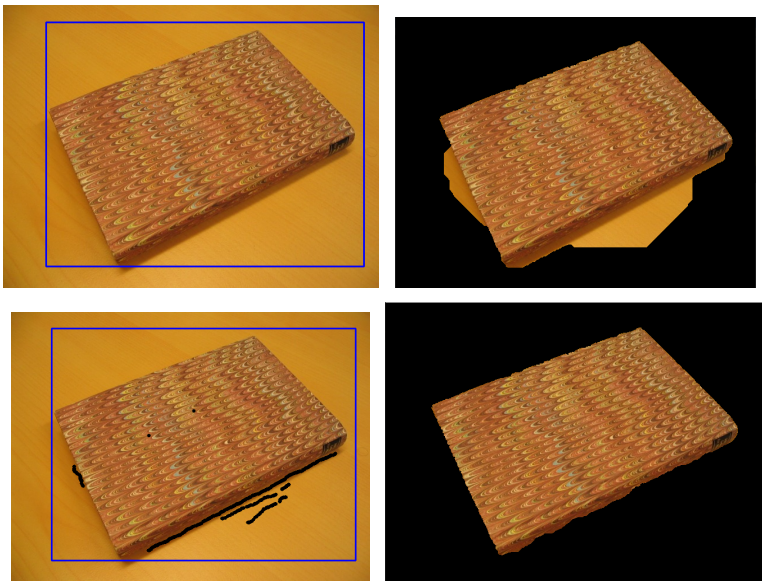
Bush

Person 4

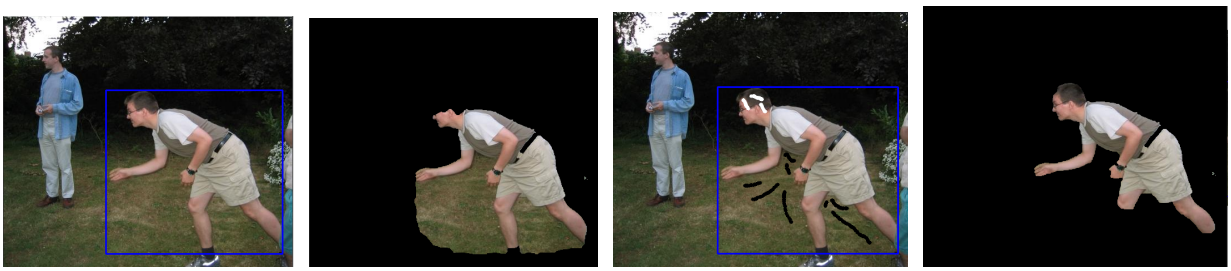
Lama



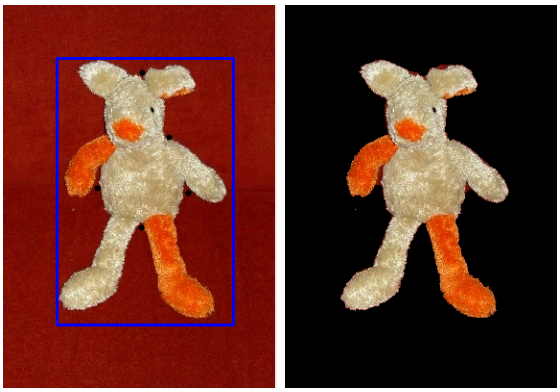
Book



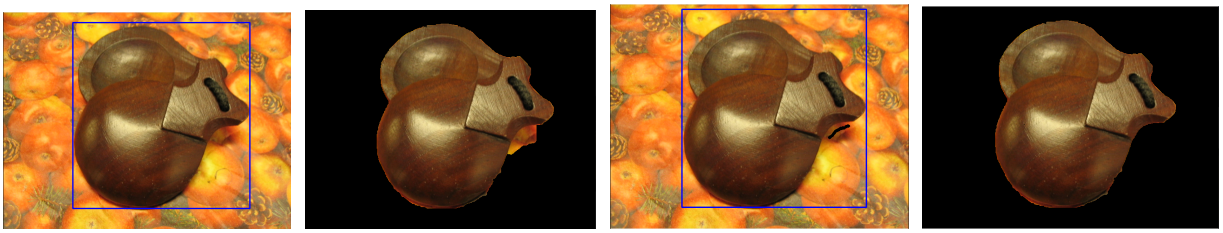
Bool



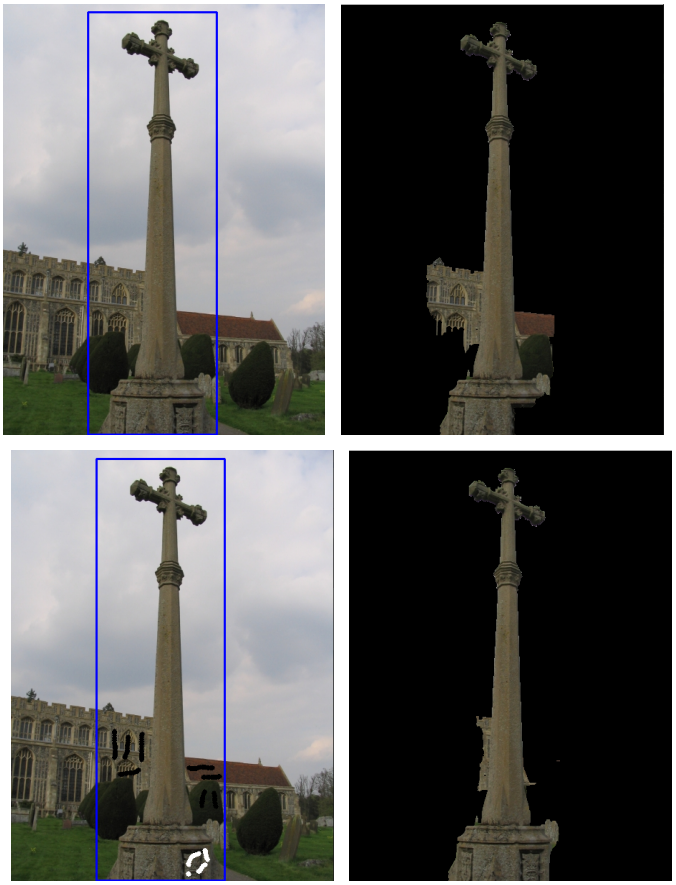
Teddy



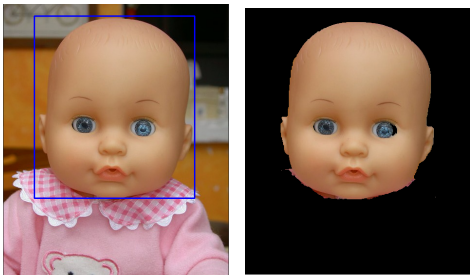
Music



Cross



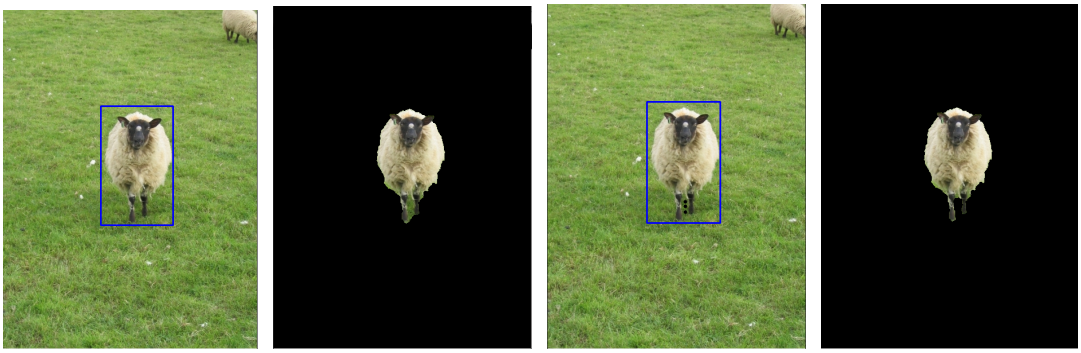
doll



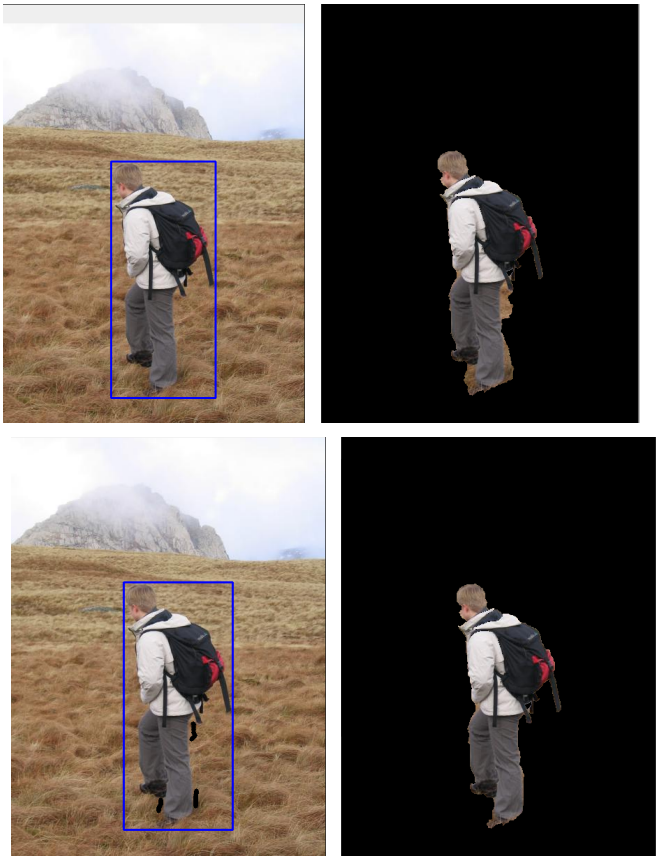
Person1

person2

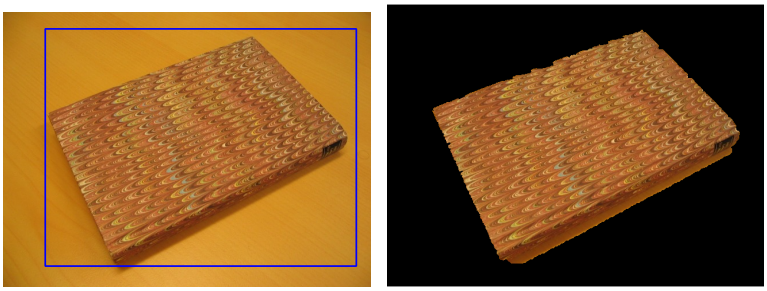
sheep



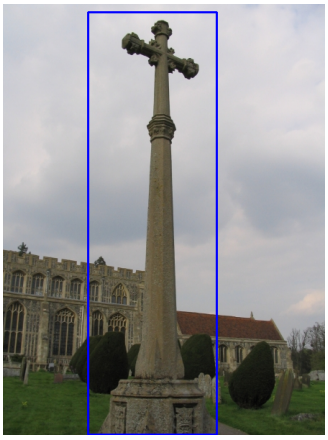
Person3



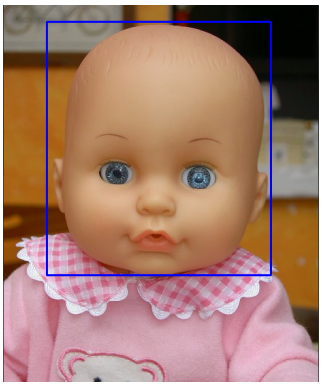
book



cross



doll



scissors

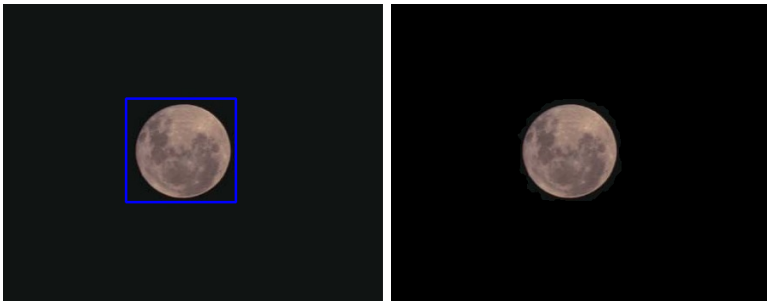
stone1

stone 2

grave

flower

Full Moon



MRFs for Image denoising

Steps

1. Add random noise to the image
2. Use MRF for recovering the image as described here

References

- https://blog.statsbot.co/probabilistic-graphical-models-tutorial-d855ba0107d1
- Markov Random Fields
- http://stanford.edu/class/ee367/Winter2018/yue_ee367_win18_report.pdf

Formulas

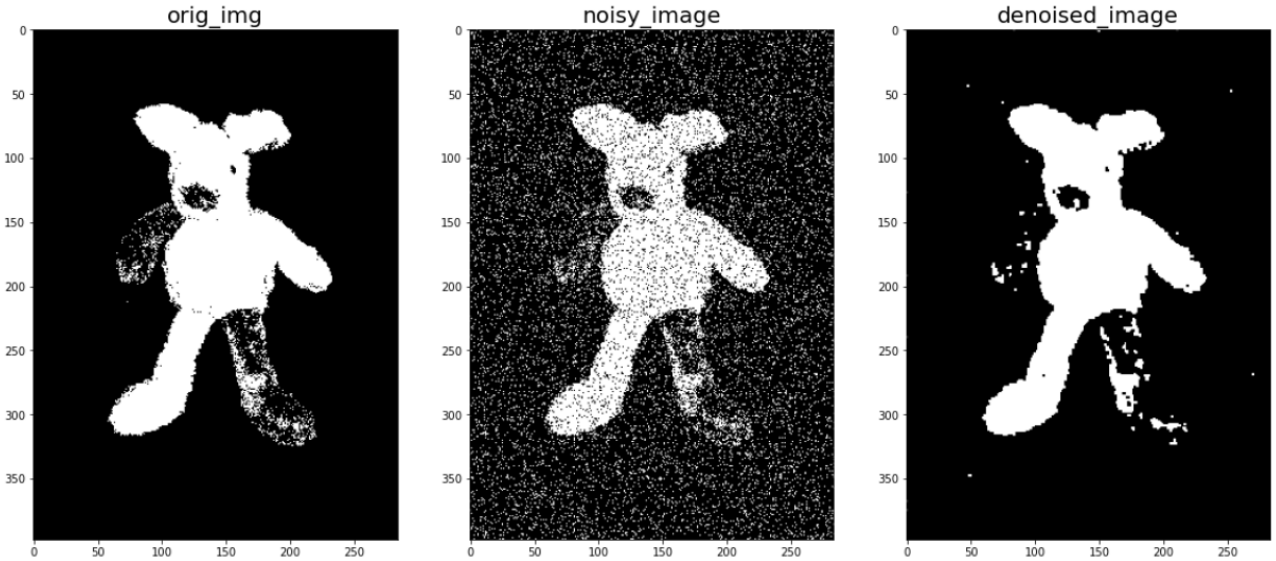
$$
\begin{aligned}
Y^* &= \arg\max_Y P(Y|X) \\
&= \arg\max_Y \log P(Y|X) \\
&= \arg\max_Y [\log P(X,Y) - \log P(X)] \\
&= \arg\max_Y \log P(X,Y)
\end{aligned}
$$

$$p(X,Y) = \frac{1}{Z} \prod_{ij} \phi(X_{ij}, Y_{ij}) \prod_{(ij,kl)} \phi(Y_{ij}, Y_{kl})$$

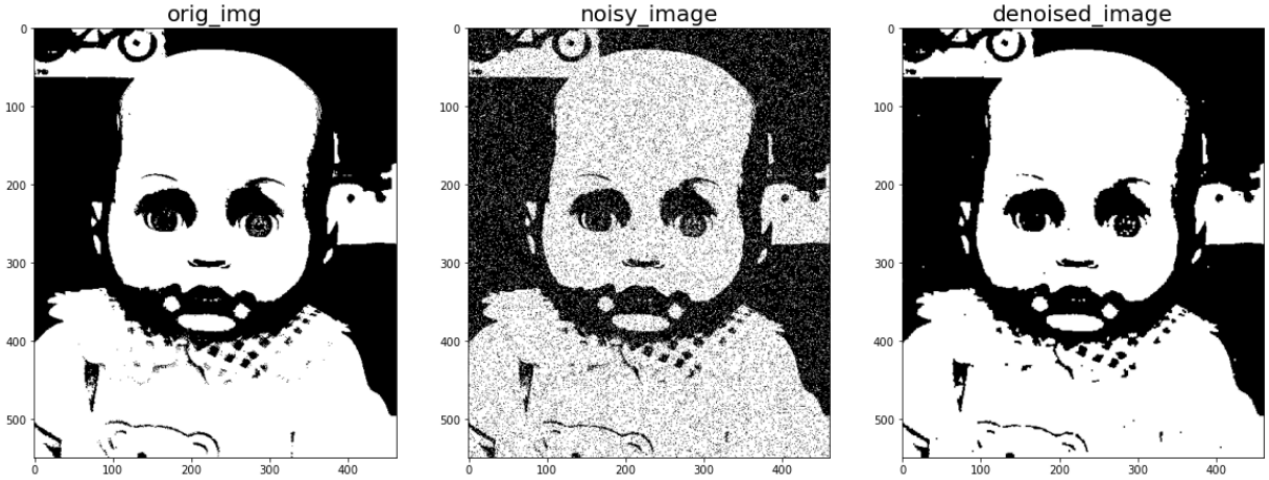
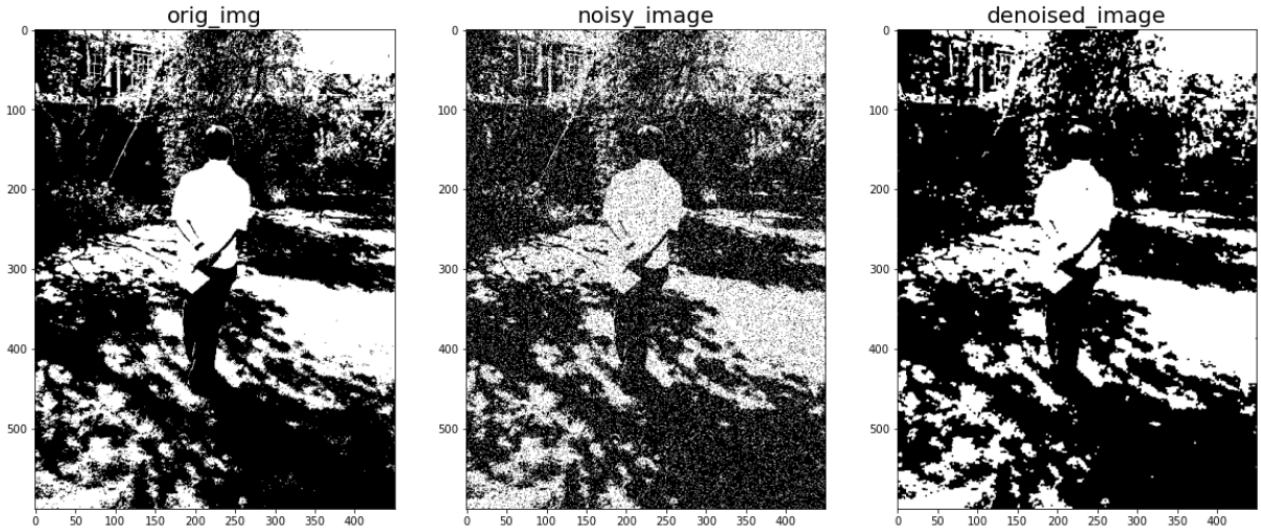
$$Y^* = \arg\max_Y \log p(X,Y) = \arg\max_Y \sum_{ij} w_e X_{ij} Y_{ij} + \sum_{(ij,kl)} w_s Y_{ij} Y_{kl}$$

Results

```
Percentage mismatched pixels is  1.3465213390898152
```




```
Percentage mismatched pixels is  0.9178435407943605
```



```
Percentage mismatched pixels is  5.45
```