Home Assignment 3

Shahin Mammadov
Abumansur Sabyrrakhim

April 21, 2021

The work was done in a team of two – by **Shahin Mammadov** and **Abumansur Sabyrrakhim**. The work distribution was following:

- **Shahin Mammadov**: implemented `__init__`, `getShape`, `getArray`, `mutualInformation` methods, as well as `addGaussian.py`.
- **Abumansur Sabyrrakhim**: implemented `setBinSize`, `coloredChannels`, `splitChannels`, `cropImage`, `getEntropy` methods.
- The rest, including experiments and report, was done together.

Mutual Information

Mutual information is a measure of image matching, that does not require the signal to be the same in the two images. It is a measure of how well you can predict the signal in the second image, given the signal intensity in the first [1].

In the output folder there are 3 folders for each picture(ignore the ones with Gaussian). Answers for this tasks are inside of those folders and are named accordingly.

IMAGE 1. LANDSCAPE

Figure 0.1: The input image

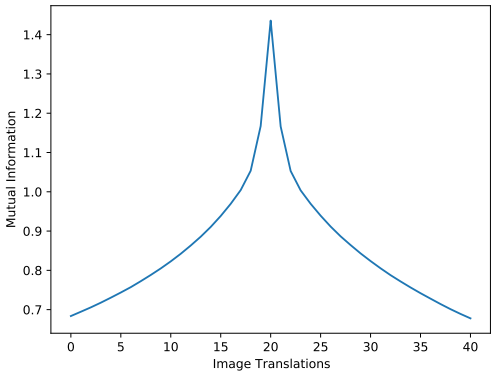

Figure 0.2: The Mutual Information Plot

According to Figure 0.2, we can see that we reach maximum value of MI at the image translation equal to 20, which means that at image translation = 20, since both images are cropped, the MI value finds the optimal combination of red and green.

Figure 0.3: Red

Figure 0.4: Green

Figure 0.5: Green cropped

IMAGE 2. FLOWERS

Figure 0.6: The input image

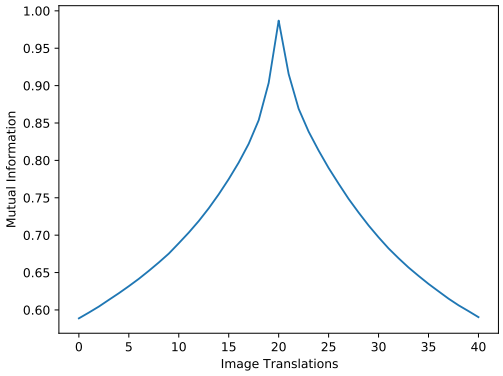

Figure 0.7: The Mutual Information Plot

Here again, according to Figure 0.7, we can see that since the optimal combination of red and green channels is reached at image translation equal to 20, the peak of mutual information value is at 20.

Figure 0.8: Red

Figure 0.9: Green

Figure 0.10: Green cropped

IMAGE 3. PARROT

Figure 0.11: The input image

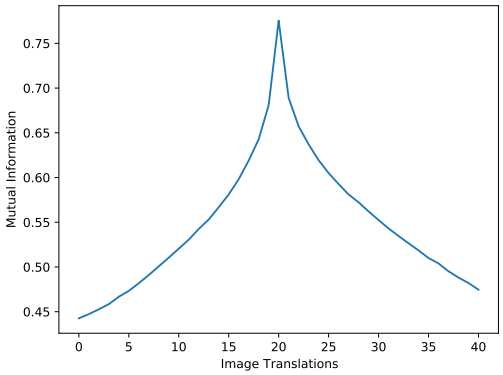

Figure 0.12: The Mutual Information Plot

In this example as well, you can see from Figure 0.12, the peak of MI value is reached at image translation equal to 20.

Figure 0.13: Red

Figure 0.14: Green

Figure 0.15: Green cropped

PIXEL NOISE APPLIED TO PARROT IMAGE

Output for this task are in the other folders that starts with Gaussian. “01” refers to how much Gaussian was applied (in this case 0.1) and “03” refers to “0.3” Gaussian applied. Bin refers to bin size which in our cases are 256, 512, 1024.

In this experiment we used Gaussian noise, that was applied by using algorithm you can find in `addGaussian.py` file. As you can see, the peak of Mutual Information is reached at Image translation equal to 20.

Gaussian = 0.1

Based on the results of our experiment, we can observe that the value of bin does not impact image noise, since it is clear that all images at Gaussian value 0.1 have approximately same noise level.

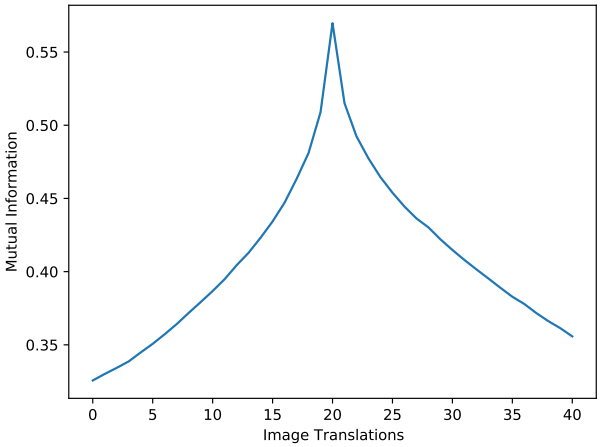

Figure 0.16: MI plot of Gaussian = 0.1, bin = 256

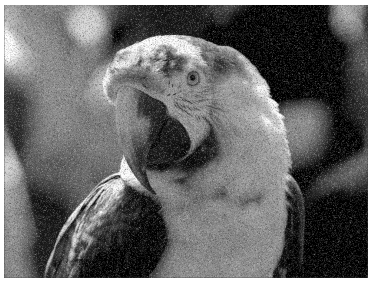

Figure 0.17: Red

Figure 0.18: Green

Figure 0.19: Green cropped

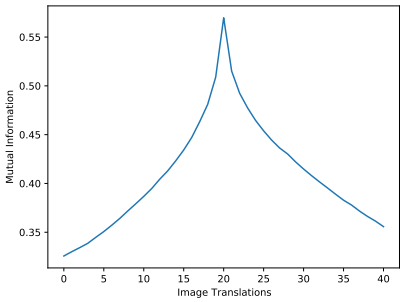

Figure 0.20: MI plot of Gaussian = 0.1, bin = 512

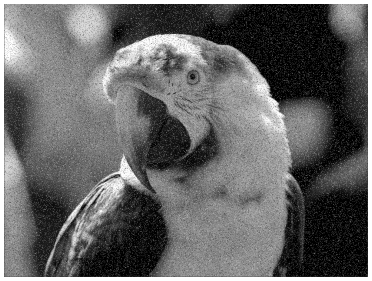

Figure 0.21: Red

Figure 0.22: Green

Figure 0.23: Green cropped

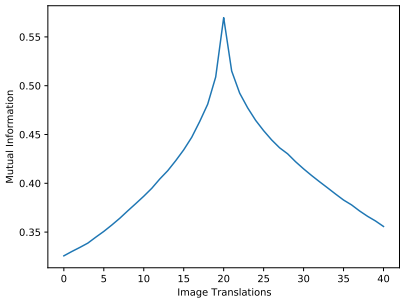

Figure 0.24: MI plot of Gaussian = 0.1, bin = 1024

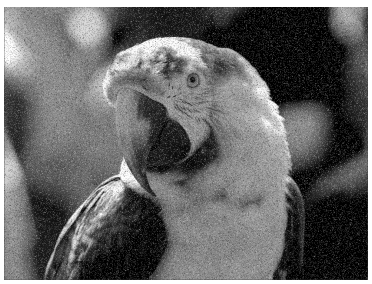

Figure 0.25: Red

Figure 0.26: Green

Figure 0.27: Green cropped

Gaussian = 0.3

After increasing Gaussian value to 0.3, we can clearly observe bigger amount of "salt-and-pepper" noise, which is also about the same for all three bin values.

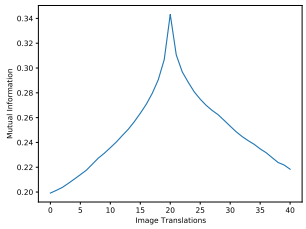

Figure 0.28: MI plot of Gaussian = 0.3, bin = 256

Figure 0.29: Red

Figure 0.30: Green

Figure 0.31: Green cropped

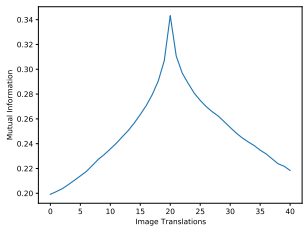

Figure 0.32: MI plot of Gaussian = 0.3, bin = 512

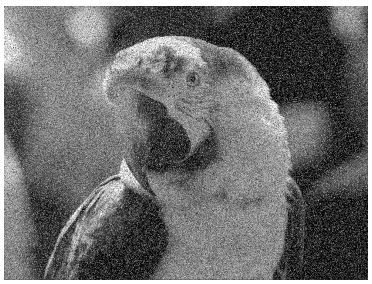

Figure 0.33: Red

Figure 0.34: Green

Figure 0.35: Green cropped

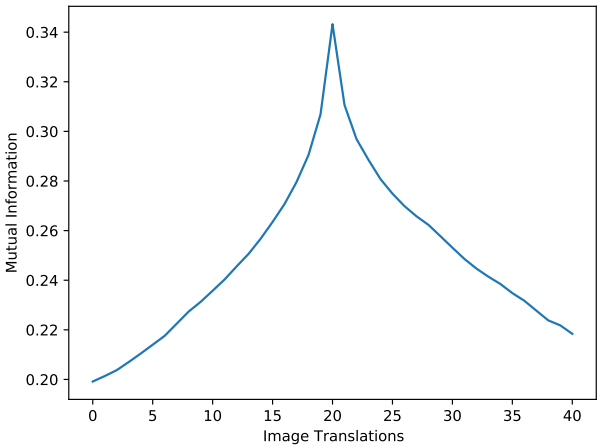

Figure 0.36: MI plot of Gaussian = 0.3, bin = 1024

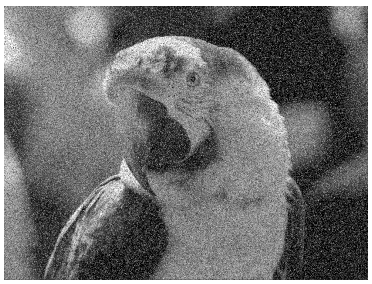

Figure 0.37: Red

Figure 0.38: Green

Figure 0.39: Green cropped

REFERENCES

[1] The Definition of Mutual Information
`https://matthew-brett.github.io/teaching/mutual_information.html`

[2] `https://en.wikipedia.org/wiki/Mutual_information#In_terms_of_PMFs_for_discrete_distributions`