Technical University of Cluj-Napoca **Author:** Zaharia Tudorita
Semestral Project for Image processing course **Supervisor:** Fuzes Attila

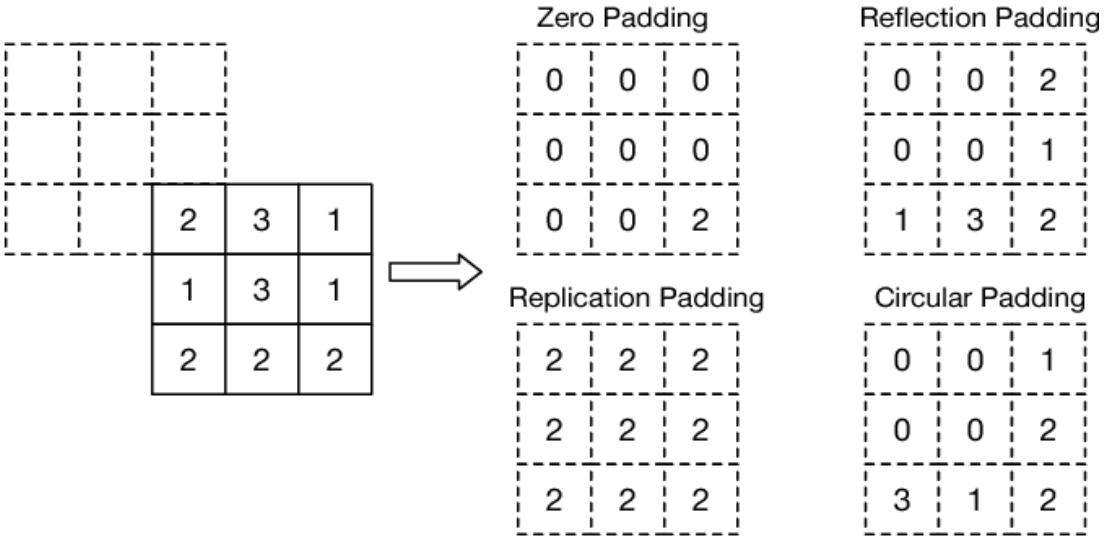
Texture Segmentation with Mask filters

1. Introduction

Image segmentation plays a significant role in understanding visual scenes. This task involves partitioning an image domain into subdomains based on a different mathematical criterion (Bo-Young Park, 2013) . The main issue concerning this algorithm is to highlight a texture out of an image, considering that a texture represents a surface with properties distinctive from the rest of the picture. Based on these constraints, the objective proposed for this project is to create a mask applied to an image having the purpose to segment and highlight the texture.

2. Theoretical Background

Before explaining in deep what the algorithm does, there are a few concepts needed to be explained.
Entropy – in image processing refers to how much uncertainty a pixel holds (M.Joseph Prakash, 2013). The grade of uncertainty is computed based on the grayscale values and how much it differs from the neighboring pixels.
Firstly, we compute adjacency matrix for each pixel. For border pixels we will use reflection padding, a technique in which the outside pixels get the values of the position-based symmetrical pixel (ROBERT M. HARALICK, 1973).



The segmentation usually happens on binary images. As a consequence, there will be needed algorithms converting to black and white and processing binary images such as morphological operations.
The mathematical closing operation is defined as $A \bullet B = (A \oplus B) \ominus B$. This operation is used for noise removal, as it helps to filter out small holes.
The mathematical opening operation is defined as $A \circ B = (A \ominus B) \oplus B$. This operation is used for noise removal similar to closing operation, however it removes small objects.

3. Design and implementation

In the previous diagram, there is presented the algorithm of thresholding (MATLAB, 2022). Therefore, the algorithm from pre-processing an image until segmenting will be presented and explained.

Texture segmentation Algorithm

Step 1. The input image will be denoted as $img(i,j)$ representing a grayscale picture.
Step 2. The image will be filtered using the local entropy function H. This function computes the local entropy for an array P of normalized values from the 3x3 matrix of neighbors.

$$H(i,j) = -\sum_{k=0}^{255} P(x_k) log P(x_k)$$
$$img^*(i,j) = H\ (i,j)\ \ for\ each\ pixel$$

Step 3. We denote a threshold T a number used to transform $img(i,j)$ into a binary image based on the formulae

$$img(i,j) = \begin{cases} 0, & if\ img^*(i,j) < T \\ 1, & if\ img^*(i,j) > T \end{cases}$$

Step 4. Then we perform the morphological operation of opening to filter out the small objects. For that we will use a Kernel K specific for dilating the 4-neighbors.

$$img(i,j) = img(i,j)\ \oplus \mathbf{K}$$

Step 5. Similarly, we use the morphological operation of closing to filter out small holes by using the same Kernel.

$$img(i,j) = img(i,j)\ \ominus \mathbf{K}$$

Step 6. The resulting image should have an object completely separated from the texture. We apply the largest contour algorithm from OpenCV library and extract the largest border object (OpenCv, 2022).

$$img(i,j) = \max(\ Contour[x], Contour[y])\ \ \forall\ x,y\ \in Contour\ of\ img$$

Step 7. Finally, we draw the outside of contour, which should symbolize the texture around the object.

4. Experimental Results and Discussion

In this section, I would like to discuss that this method is quite theoretical and most of the implementation rather than focusing on efficiency or accuracy, tries to apply mathematical concepts to achieve the goal of segmentation and to make the future user acknowledges basic principles of segmentation.

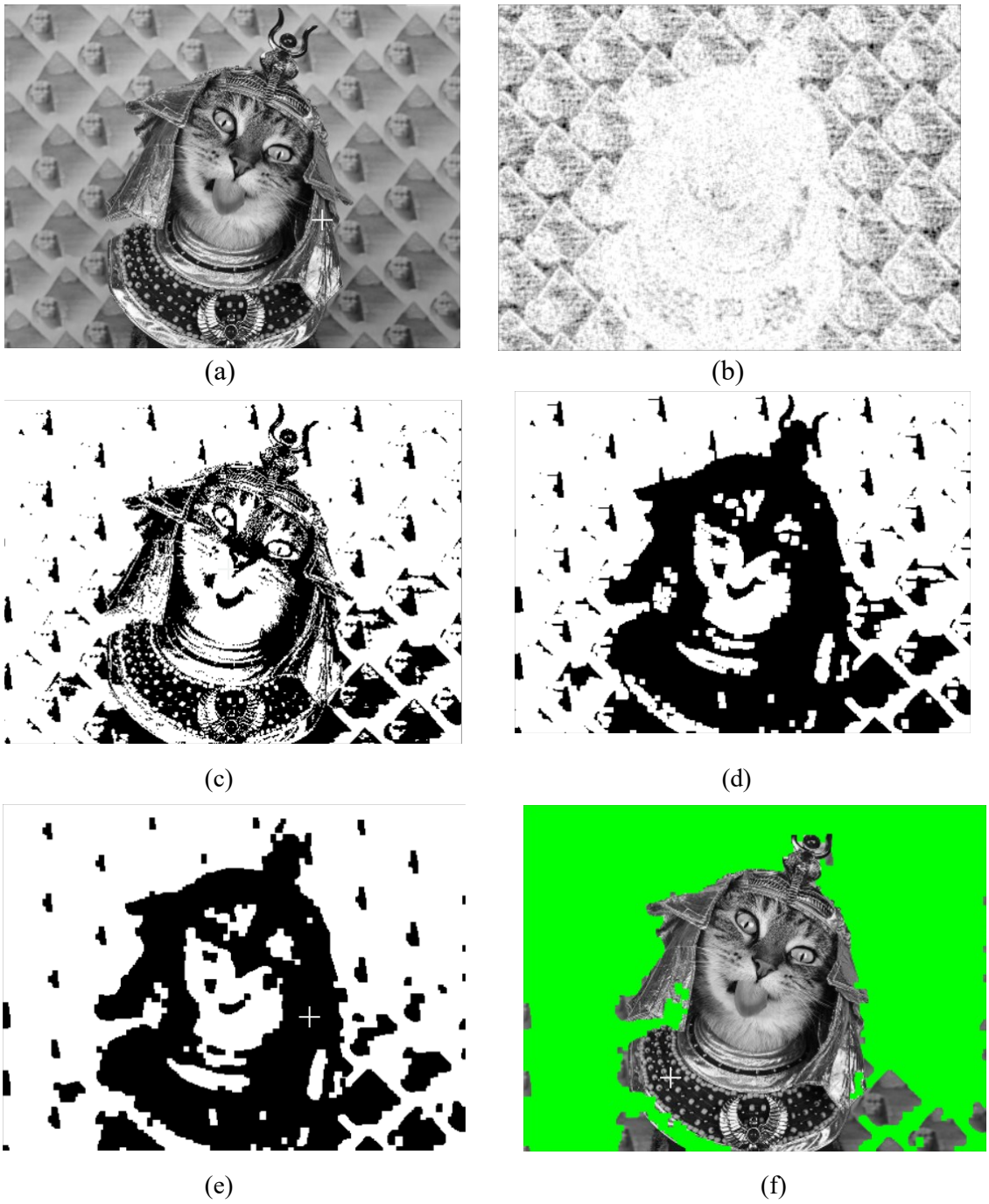


Fig. 1: (a) Original image, (b) Entropy image, (c) Binary Image, (d) Area-opened image, (e) Area-closed image, (d) Largest contour extracted and the background colored with Green.

As it can be observed in the test case, the opening and closing of image are not done properly, as a result We needed to achieve a better outcome using the largest contour. The main backlash of this method is that we cannot segment textures having multiple objects, because it will cut out the smaller objects leaving only one. As a future improvement, the morphological operations could be done more accurately in order to filter out the small objects floating in the image.
Furthermore, the main implementation was done purely using morphological operations due to some error in test cases segmentation when applying the entropy filter. Because of that, the image with entropy filtered is showcased only for the user to see how it looks like, however it is not used in the current code as it is still in the process of development.

5. Conclusions

The algorithm proposed by this paper is described in length by the MATLAB website however it has some small modifications, and it is unstable having low accuracy. There is however substantial evidence that it achieves the segmentation based on textures as presented. The output image shows an object separated from a texture. For better test results, the implementation ought to improve the morphological operations and correctly apply entropy without the need of contouring largest object.

6. Bibliography

1. Bo-Young Park, H.-H. K.-W. (2013). A Multilabel Texture Segmentation Based on. *Hindawi*, 7.
2. Eitan Sharon, A. B. (2003). Segmentation and Boundary Detection Using Multiscale Intensity Measurements. *IEEE*, 8.
3. M.Joseph Prakash, D. K. (2013). MORPHOLOGY BASED TECHNIQUE FOR TEXTURE. *Signal & Image Processing : An International Journal (SIPIJ)*, 8.
4. MATLAB. (2022, January 1). *Texture Segmentation using Texture Fields*. Retrieved from MathWorks: https://www.mathworks.com/help/images/texture-segmentation-using-texture-filters.html
5. OpenCv. (2022). *Structural Analysis and Shape Descriptors*. Retrieved from docs.opencv: https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a
6. ROBERT M. HARALICK, K. S. (1973). Textural Features for Image Classification. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 12.