

Implementation of Backpropagation for Ulap-ulap Pattern Recognition

Ni Luh Yulia Alami Dewi^{a1}, I Wayan Santiyasa^{a2}

^aInformatics Department, Mathematics and Science Faculty, Udayana University
Bali, Indonesia

¹yulia.alamidewi01@gmail.com

²santiyasa@unud.ac.id

Abstract

Ulap-ulap is one of the symbols used to indicate that a building has been carried out Mlaspas ceremony. Mlaspas is one of the ceremonies performed to purify and clean a building. Ulap-ulap itself consists of various types depending on the building where it is placed, for example the ulap-ulap placed on the Pelinggih building will be different from the ulap-ulap placed on the Bale building. So that the pattern contained in each type of Ulap-ulap is different. The purpose of this research is to be able to do pattern recognition on Ulap-ulap images. The method used in this study is Backpropagation, and for its implementation, the MATLAB 7.5.0 (R2007b) application is used. This study used 18 images of Ulap-ulap, including 15 training data and 6 test data. The stages of the process carried out are for Ulap-ulap pattern recognition, the first is data collection, then image processing, and finally the pattern recognition. Recognition of the Ulap-ulap image pattern with Backpropagation, resulted in an accuracy of 83.333%.

Keywords: Image Processing, Pattern Recognition, Backpropagation, Ulap-Ulap, MATLAB

1. Introduction

Etymologically, *Ulap-ulap* comes from "Balinese language which in Indonesian means to call by hand / *ngauikin aji lima*". This means summoning a certain power or holy light from Ida Sang Hyang Widhi Wasa to animate the building, in the sense of sacred buildings such as *Sanggah* and *Pura* so that the Gods want to stay in the sacred building. In house building, so that the negative forces that are in the building, such as worker's footprint, and evil spirits are lost, so that people can safely and peacefully live there^[1]. In the process of making Balinese script during *Ulap-ulap*, the function of the motor nerves will increase because our mind is concentrating, and the hands are moving simultaneously^[2].

Ulap-ulap are placed on buildings that have been carried out by the *Mlaspas* ceremony. *Mlaspas* is one of the ceremonies performed to purify and clean a building. *Ulap-ulap* has 142 different types^[2]. The types of *Ulap-ulap* are different according to the place where they are placed, for example the ones that are placed on the *Pelinggih* building will be different from the ones that are placed on the Bale building. So that the patterns contained in each type of *Ulap-ulap* are different. Based on these problems, the author wants to do research on pattern recognition from *Ulap-ulap* images.

In this research, one of the artificial neural network methods that will be used to perform the *Ulap-ulap* image pattern recognition is Backpropagation. In this research, the pattern of the *Ulap-ulap* that will be recognized is *Ulap-ulap Padma*, *Ulap-ulap Pelinggih*, and *Ulap-ulap Bale*. In this research, the MATLAB application will be used to implement each process for *Ulap-ulap* pattern recognition. MATLAB itself is a software used for programming, analysis, as well as matrix-based technical and mathematical computing^[3].

2. Research Methods

2.1. Data Collection

In this study, the image data used were *Ulap-ulap Padma*, *Ulap-ulap Pelinggih*, and *Ulap-ulap Bale*. The amount of image data is 21 images, including 15 training data and 6 test data. The *Ulap-Ulap* image data was obtained from *Griya Giri Murti's Ulap-Ulap* book. The following is the *Ulap-ulap* image data used.

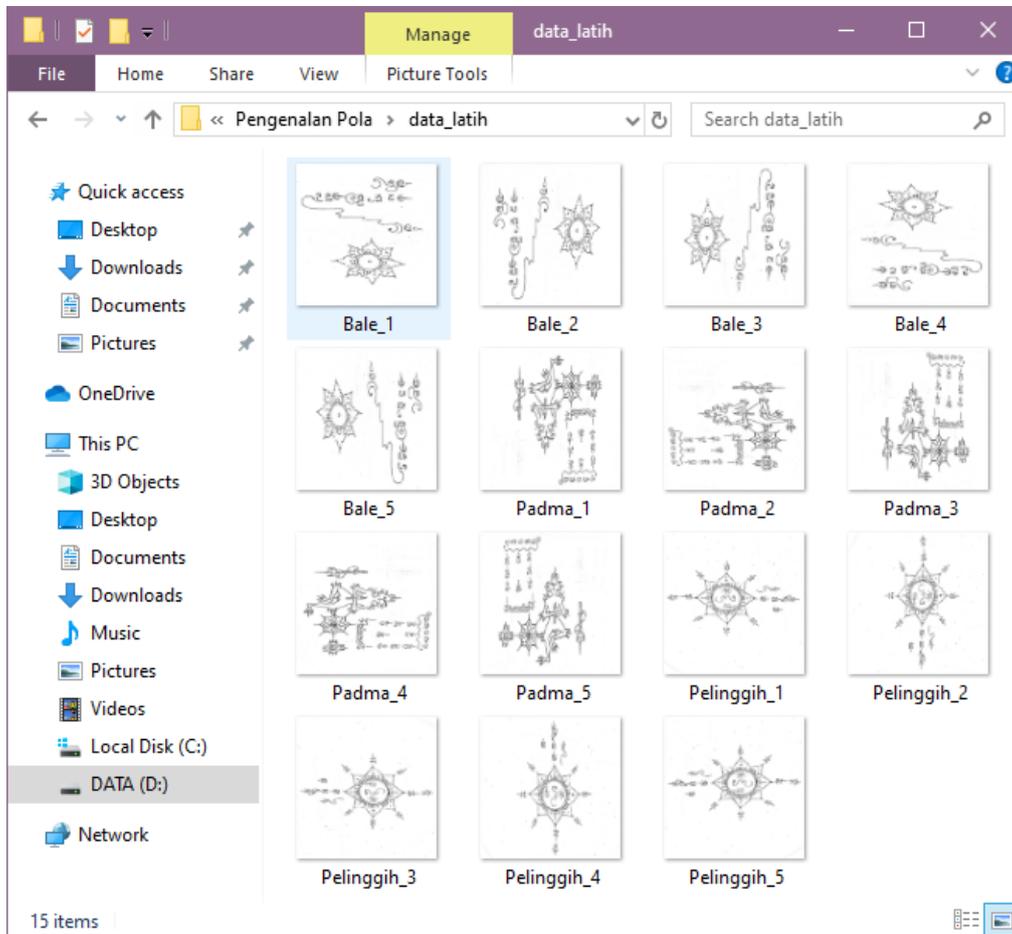


Figure 2.1. Data Training

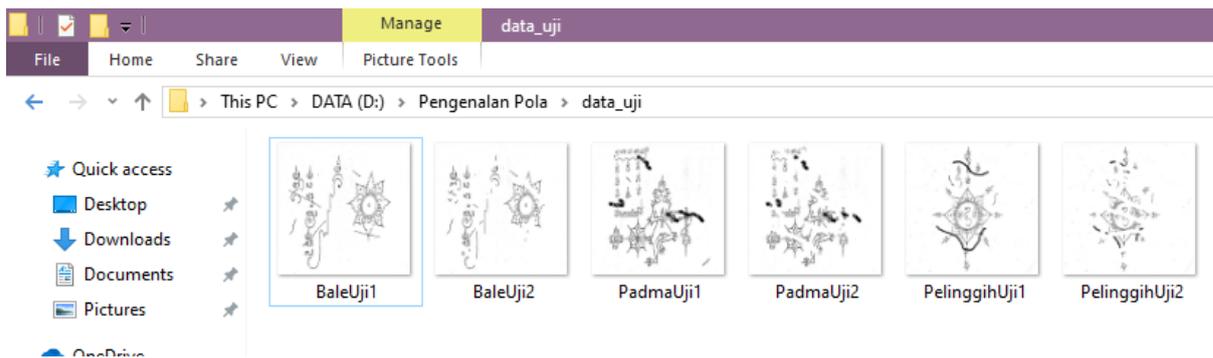


Figure 2.2. Data Testing

2.2. Image Processing

In the image processing process, the Grayscale process will be carried out in the *Ulap-ulap* image first. Digitally, grayscale can be represented in the form of a two-dimensional array. Each element in the array shows the intensity (grayscale) of the image at the corresponding coordinate position. If an image is represented in 8 bits, then the image has 2^8 or 256 levels of grayscale^[4]. In MATLAB, to change the image to grayscale, you can use the `rgb2gray` command. The following is the command used to convert the image to grayscale.

```
%Grayscale  
image = rgb2gray(image);
```

Figure 2.3. Grayscale

After that, the grayscale image will be converted into binary using the Threshold process. Threshold is a process to turn a gray scale image into an image with binary value. The value of one will represent white, while the value zero will represent black^[5]. In MATLAB, to perform a threshold, you can use the `im2bw` command. The following is the command used to convert the grayscale image into binary form using a threshold.

```
%Biner  
image = im2bw(image, 0.8);
```

Figure 2 4. Threshold

In this study, the value of T at the threshold process, the authors use 0.8. After that, the image resulting from the threshold process will be converted into a vector form. Before converting to vector, the image resulting from the threshold process will be reduced in size by 90%. The image size is reduced so that the image data can be carried out by the pattern recognition process with Backpropagation in MATLAB, if the image size is too large, then the pattern recognition process cannot be carried out either. After that, it will be converted into a vector form. The following are the commands used.

```
%Perkecil 90%  
[baris,kolom] = size(image);  
image = imresize(image,[round(baris*0.9) round(kolom*0.9)]);  
%cek ukuran yg baru  
[baris,kolom] = size(image);  
%Ubah ke vektor  
image = reshape(image,1,(baris*kolom),[]);
```

Figure 2.5. Converted into Vector

After that, the image resulting from image processing will be saved in the `.mat` file format.

2.3. Pattern Recognition

a. Training Process

In the training process, training data will be trained using `nntool` in MATLAB. The first is to load the training data and target training in the command window. Then the training data will be transposed. The following commands are used.

```
>> load data_train.mat  
>> newdatalatih = transpose(data_train);  
>> load target_latih.mat
```

Figure 2. 6. Load Data Train and Target Training, Transpose Data Training

After that, the Backpropagation architecture was made using `nntool`. Type `nntool` in the command window, after that it will open as shown below.

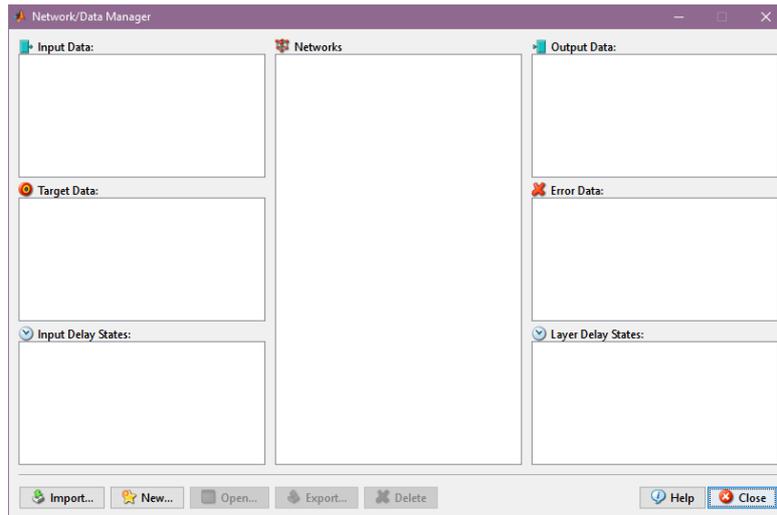


Figure 2.7. nntool

Then do the Backpropagation architecture creation by selecting the Import button. The following is a way of making the Backpropagation architecture used in this study.

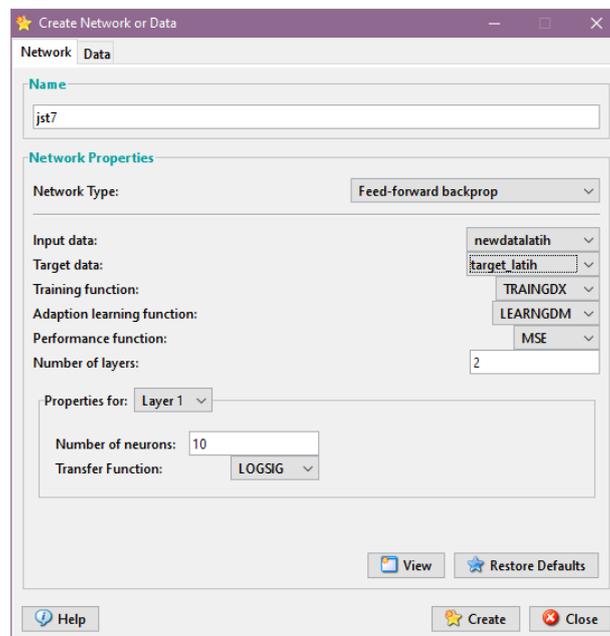


Figure 2.8. Create Network

In the picture above, it can be seen that the name of the network used is jst7, the network type is Feed-forward backprop or it can be said that the network type used is Backpropagation. Then the input data used is newdatalatih, the target data used is target_training. After that, the training function used is Traingdx, the Performance Function is MSE, the number of Hidden Layers used is 2 with each layer having 10 neurons, and the activation function used is Logsig or Sigmoid. The following is the Backpropagation architecture used.

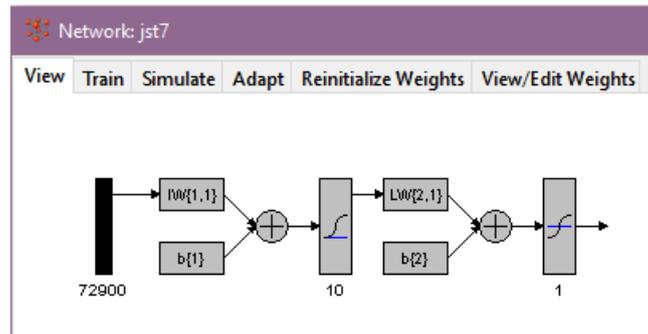


Figure 2.9. Backpropagation Architecture

The parameters in the Backpropagation architecture used in this study are as follows.

Table 2. 1. Backpropagation Parameters

No	Parameter	Score
1	Hidden Layer	2
2	Activation Function	Sigmoid
3	Error Goal	0.001
4	Number of Epoch	1000
5	Learning Rate	0.1

After that, the training process is carried out. The following are the results of the training process.

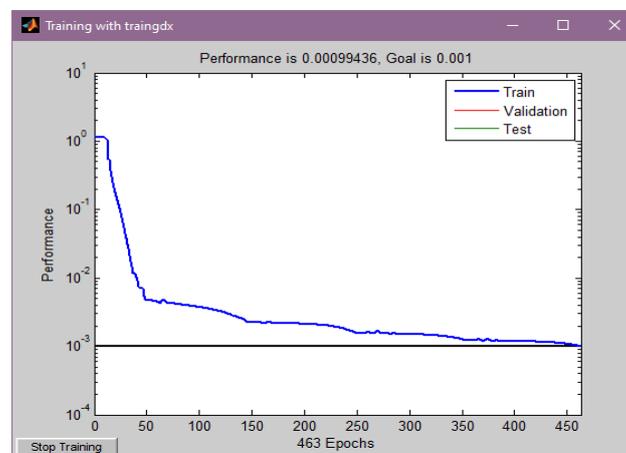


Figure 2.10. Result of the Training Process

From the picture above, it can be seen that the target error is reached on the 463 epoch with MSE = 0.00099436.

b. Testing Process

In the testing process, the test data image that has been processed in image processing and the test target will be loaded in the command window. The following orders were carried out.

```

Command Window
New to MATLAB? Watch this Video.
>> load data_uji3.mat
>> load target_uji.mat
    
```

Figure 2.11. Load Data testing and Target Testing

After that, the test data will be transposed. The following commands are used

```
>> newdatauji90 = transpose(data_uji3);
```

Figure 2.12. Commands for Transpose Data Testing

After that, the test data will be carried out by the pattern recognition process with the Backpropagation architecture that has been previously made. Tests are carried out using the `round` and `sim` commands in MATLAB. The following commands are carried out.

```
>> hasil = round(sim(jst7,newdatauji90))
```

Figure 2.13. Command for Testing Process

3. Result and Discussion

In this study, 18 ulap-ulap image data were used, including 15 training data and 6 test data. Implementation is done using the MATLAB application. In this study, the MATLAB application used is MATLAB 7.5.0 (R2007b). Before carrying out the pattern recognition process, the image processing process is first carried out. Image processing result data will be transraned first with the backpropagation architecture created. After that, the testing phase will be carried out. The results obtained are as follows.

```
>> hasil = round(sim(jst7,newdatauji90))

hasil =

     3     2     1     3     1     3

>> [target_uji;hasil]

ans =

     3     2     1     3     1     2
     3     2     1     3     1     3
```

Figure 3.1. Result of the Pattern Recognition

From the figure, it is known that, of the 6 test data carried out, there is 1 test data that cannot be recognized, namely the 6th data. So the accuracy obtained by the backpropagation architecture that has been made for Ulap-ulap pattern recognition is 83.333%.

4. Conclusion

From the results of the research that has been done, it can be said that Backpropagation can be used to carry out the *Ulap-Ulap* pattern recognition process. This can be seen from the accuracy obtained, which is equal to 83,333%. In the future, it may be possible to carry out pattern recognition research with other methods.

References

- [1] I. W. Watra, N. A. Wandri and Sukrawati, *Ulap-Ulap dan Rerajahan dalam Agama Hindu di Bali*, Denpasar: Paramita, 2008.
- [2] I. B. Anom, *Ulap-Ulap Manut Wawangunan*, Tabanan, 2014.
- [3] A. Tjolleng, *Pengantar Pemrograman MATLAB*, Jakarta: PT. Elex Merdeka Komputindo, 2017.
- [4] A. Basuki and J. Palandi, *Pengolahan Citra Digital Menggunakan Visual Basic*, Yogyakarta: Graha Ilmu, 2005.
- [5] R. C. Gonzales and R. E. Woods, *Digital image processing*, In: Prentice hall New Jersey, 2002.