IJITEB
Vol. 7
No. 2
2025

International Journal of Information Technology and Business

http://ejournal.uksw.edu/ijiteb

Design of Batik Motif Detection System Using Deep Learning Method

Janinda Puspita Anidya [1*], Hindriyanto Dwi Purnomo [2]

[1-2] Faculty of Information Technology, Universitas Kristen Satya Wacana, Indonesia

Abstract: Batik in Indonesia is growing very rapidly, almost every region or city in Indonesia has a variety of batik motifs. The batik motifs owned by each region are their own wealth and heritage in each region that must be preserved and maintained properly. So the Indonesian people need to collaborate with each other to place batik preservation as a top priority in the field of preserving the nation's culture. Because knowledge of batik motifs in Indonesia is important in order to maintain the culture of the Indonesian nation, including knowledge of batik motifs in each of their respective regions, so there is a need to make it easier for humans to recognize batik motifs in regions in Indonesia quickly and easily. This study aims to build a system model that can assist humans in recognizing batik motifs in Indonesia through this batik motif detection system. This research produces a model that can detect batik motifs from every area in Central Java. The conclusion of this study is a batik detection model that can help and introduce to the public about various batik motifs from each region

Keywords: Batik Motif, Model, Detection System

1. Introduction

Technological developments that are increasingly advanced and rapidly growing encourage in creating an intelligence that can help humans in doing a job or activity. The world today is in the digital age. An era where almost every aspect of human life is closely related to computing technology. With the development of the times, humans will continue to develop knowledge and technology to help and ease human work.

And one of the fields of research that is currently growing is artificial intelligence or better known as Artificial Intelligence. Artificial intelligence that is owned by humans is growing rapidly in modeling it in machines and programmed to be able to walk and think like humans. So this encourages the author to create a system that can be useful in providing knowledge and developing knowledge in a program. In Indonesia, batik is growing rapidly, almost every region or city in Indonesia has a variety of batik motifs.

The batik motifs owned by each region are the wealth or heritage of each region, and in each motif makes batik has a unique and interesting value. Various kinds of batik motif designs that are owned by each region are a sign that the batik contains meaningful symbols or illustrations. Even though batik is worldwide, there is still someone who has not been able to find out more specifically the batik motifs from each region. Of course, batik is a form of Indonesian cultural heritage that must be preserved.

Indonesian batik has economic value because of its high sales prospects. In its superiority, batik is seen as being able to improve the welfare of the community, therefore it is appropriate for Indonesian people to preserve batik. So the government and the Indonesian people need to collaborate with each other to place the preservation of batik as a top priority in the field of preserving the nation's culture. Because knowledge about batik motifs in Indonesia is important in order to maintain the culture of the Indonesian nation, including knowledge of batik motifs in each of their respective regions, and the ability to know batik motifs is very important when we have to introduce the wealth of Indonesian batik motifs to foreign tourists as because of our cultural heritage, the batik motifs that we introduce can be understood by foreign tourists that the motif is a heritage owned by the area, or just like when we want to make batik, then we must know we will make batik from which regional motifs we want.

Based on the existing problems, it can be drawn an outline that the community needs a program to make it easier to recognize batik motifs in regions in

Indonesia. So that the author is able to create a program or system that can facilitate the public in introducing batik motifs in Indonesia through this batik motif detection system. This system will apply a method to detect batik motifs quickly and accurately, so that users who are not familiar with batik motifs from each region will get information on batik motifs more quickly and accurately through this batik motif detection system[1][2].

2. Literature Review

The research entitled batik motif detections using pattern recognition method determines the fastest and most accurate pattern recognition method in detecting and interpreting batik motifs by making comparisons between pattern recognition. This study aims to provide experimental data including the application of the method to program the batik motif detection application. The purpose of this experiment is to create an optional method other than manual search on batik motif detection. Batik motif detections using pattern recognition method are proven to increase user satisfaction for detecting batik motifs[3].

The next research entitled deep learning convolutional neural network for object recognition classification using mxnet discusses the creation of a batik motif image data classification model using a deep learning method with a convolutional neural network architecture. CNN classification with an architecture of two convolution layers and two fully connected layers with 80% training data and 20% testing data results in an accuracy of 52.50% for batik image classification. Deep learning convolutional neural network for object recognition classification using mxnet is able to perfectly recognize batik motifs from all data[4].

In the study entitled classification of batik motifs using a convolutional neural network, this was carried out with a quantitative approach, the data taken were batik datasets obtained on the internet, and the research flow started from dataset retrieval to the batik class classification process according to motifs. This study uses CNN to determine three classes of batik, namely kawung, megamendung, and parang , and the testing methods used are data loss and data accuracy. Accuracy obtained using the CNN and CNN methods in combination with grayscale has a different level of accuracy, and the best accuracy value is obtained using the CNN grayscale combination method with an accuracy rate of 70%. The value of the accuracy results from the research is still possible to be improved by finding the best classification method for batik motifs, and for future research with other methods, changing image textures to other shapes and increasing the number of data sets can be done to get better results[5].

Furthermore , in the research entitled batik as the cultural identity of the Indonesian nation in the era of globalization, this research is a qualitative descriptive type, and the main data source of this research is the variety of batik in Indonesia, while secondary sources are taken from books, journals and other virtual sources as a literature review. , while the data types are text in the form of words, phrases, clauses, sentences and narratives. The researcher uses library research to collect data, then, the data is collected through recording techniques, then, the data that has been obtained is grouped into certain domains, the researcher sets three steps. First, the data that has been collected is analyzed descriptively hermeneutic, this is done to find, find and describe the history of Batik in Indonesia, second, researchers look for elements that support Batik into Indonesian cultural heritage, third, researchers look for the relationship between batik and globalization and finally converging on cultural identity. With regard to the noble cultural heritage in the form of batik art, the existence of batik has ups and downs in the search and discovery of its cultural identity, and awareness of this cultural identity goes through a long process starting from unexamied cultural identity, cultural identity search to cultural identity achievement[6].

Based on the research that has been described previously on the detection of batik motifs, a study was conducted that discussed the batik motif detection system using deep learning methods and convolutional neural network algorithms to be able to create a fast and accurate batik motif detection system. The design of the batik motif detection system is designed by building a model using tensorflow as a framework, using hard as an open source deep learning API, and using libraries that support model design, and using deep learning and CNN algorithms where the model can receive input and determine what aspects or objects. only contained in an image that the machine can use to "learn" to recognize the image, and distinguish one image from another by testing the model to produce an accurate model.

3. Research Method

The research was conducted through research stages which were divided into five stages, namely data collection, system design and model design, batik detection system implementation, system testing and analysis of test results, and finally writing a research report.

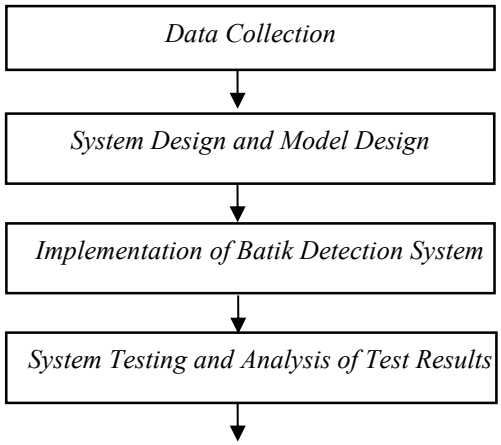




Fig. 1. Research Stages

The research stages in figure 1 can be explained as follows: The first stage is the stage of data collection. The initial stage in making a batik motif detection system is data collection, namely collecting datasets in the form of batik motifs from each region. Batik motifs are collected based on the need for a batik motif detection system. With the main goal is to achieve efficiency in project completion.

In making batik detection, the first is to collect datasets in the form of batik motifs from each region in Central Java. These batik motifs are collected and grouped based on the motifs of each region. There are 10 regions in Central Java that the authors took batik motifs to group and make datasets, namely from Banyumas, Cilacap, Jepara, Jogjakarta, Kebumen, Pati, Pekalongan, Rembang, Surakarta and Tegal areas. The author collects datasets by taking reference pictures of batik motifs from the internet, the authors take batik motifs through website sources where batik motifs from the area originate, for example such as a collection of batik motifs from the city of Pati, the original batik motifs owned by the city of Pati.

The author collects datasets of batik motifs by downloading images of batik motifs and grouping batik motifs according to their respective regions. Collecting 100 images of batik motifs from each region until the entire area is collected into 1000 images of different batik motifs to be used as a dataset. After successfully collecting images of batik motifs from each region, the authors resize the image so that each image has the same size. Namely resizing the image to be 225 in height and 300 in width.

The second stage is system design and model design. After the dataset has been collected, the next step is system design and model design, namely designing model requirements using tensorflow. The system and model are designed using google colab. The third stage is the implementation of the batik detection system. This implementation stage is by building a batik system model on Google Colab by importing a library, making reading datasets, making the number of images appear, displaying one batik motif image, sharing data for training and validation, displaying target classes, displaying size images, and building model, making training, predictive test and finally making detection of the origin of batik motifs.

The fourth stage is system testing and analysis of test results. This testing stage is to test the training by looking at the accuracy of the system. When the accuracy is increasing, the system and the model built are good enough. And analyzing between training and validation, by looking at the continuity rate when the process is running, the author can compare the accuracy of this model with fairly good results, although validation sometimes still goes up and down.

The fifth stage is writing a research report. The last stage in this research stage is writing a research report, where the results obtained and carried out when designing the batik detection model are included in the research report. The design or deep learning architecture used in this research can be explained by the author, the first thing, the author imports Tensorflow and other libraries, using Tensorflow as a framework, Numpy library, matplotlib module and PIL library. Uses hard as an open source deep learning API and uses a sequential model and uses regularizers and dropouts to reduce overfitting. After that, input the dataset by entering the import module pathlib and copying the path of the batik motif folder on the drive and placing it in data_dir.

This batik detection model uses a data collection of 1000 photos of batik motifs, containing 10 sub directories, namely 10 cities in Central Java with each directory containing 100 photos of batik motifs. Next is to display the total data of the dataset, that is, there are 1000 total images. After successfully displaying the entire dataset, the detection model displays one image of a batik motif. Then the detection model configures training and validation, namely the sharing of dataset files for training and validation.

Furthermore , the class name can be found in class_names according to the directory name and alphabetical order, the class displayed is 10 cities from each batik motif. The detection model then displays the batch image and batch label in train_ds. The display of the model, with num_classes = 10, which is 10 areas of batik motifs. And using the sequential model, padding same and activation relu.

The six lines of code below define a convolution base using a common pattern: Conv2D and MaxPooling2D layer stacks. As input, CNN takes the shape tensor (image_height, image_width, color_channels), ignoring the batch size. Color_channels refers to (R,G,B). Configure CNN to process input shapes (256, 256, 3), which are batik motif image formats.

This can be done by passing the input_shape argument to the first layer. To complete the model, we can insert the last output tensor of the convolution basis (of the form (4, 4, 64)) into one or more Dense layers to perform the classification. The dense layer takes a vector as input (i.e. 1D), while the current output is a 3D tensor. First, we'll flatten (or unroll) the 3D output into 1D, then add one or more Dense layers on top.

This model uses the Adam optimizer. Adam optimizer and loss.SparseCategoricalCrossentropy loss function. To see training accuracy and validation for each training period, provide a metric argument. The output of each Conv2D and MaxPooling2D layer is a 3D tensor shape (height, width, channel). The width and height dimensions tend to shrink as you go deeper into the network.

The number of output channels for each Conv2D layer is controlled by the first argument (e.g., 32 or 64). Usually, as width and height shrink, we can (computationally) add more output channels in each Conv2D layer. The grid summary shows that (56, 75, 64) the output is flattened into a shape vector (66304) before passing through the two Dense layers. The evaluation of this batik motif detection model uses the accuracy evaluation model and ROC Curves.

By testing using epoch 10 the results of the training test can be displayed on the accuracy results and display a chart of the results of the training. Then the output of the batik motif prediction test display can be successfully displayed according to the inputted batik motif. And the output of the detection of batik motif images that have been inputted can display the results of the detection of batik motifs that have been inputted. Preprocessing data in this batik motif detection model load images from disk using tf.keras.preprocessing.image_dataset_from_ directory.

With the batch image and batch label in train_ds, Image_batch is a tensor of the shape (32, 225, 300, 3). This is a collection of 32 images with a shape of 225x300x3 (the last dimension refers to the RGB color channel). Label_batch is a tensor of the shape (32,), it is a label corresponding to 32 images. And the RGB channel values are in the range [0, 255], not ideal for neural networks in general, so the author should try to make the input values small.

So the authors standardize the values to be in the range [0, 1] by using the tf.keras.layers.experimental.preprocessing.Rescaling layer. Next, the author can train the model using the dataset just prepared[7][8].

4. Result And Discussion

The results of this batik detection system are able to recognize the original batik motif that is inputted by the user by producing the original output of the batik that has been inputted. The results and discussion can be seen in the image below.

1. This batik detection system uses tensorlow as a framework, uses the Numpy library, the matplotlib module and the PIL library.

```
[ ]  #Import
     import tensorflow as tf
     import numpy as np
     import matplotlib.pyplot as plt
     import PIL
```

Program Code. 1. Library Phyton

2. This batik detection system uses hard as an open source deep learning API and uses a sequential model and uses regularizers and dropouts to reduce overfitting.

```
[ ]  from tensorflow import keras
     from tensorflow.keras import layers
     from tensorflow.keras import Sequential
     from tensorflow.keras import regularizers
     from tensorflow.keras.layers import Dropout
```

Program Code. 2. Open source

3. Next, read the dataset by importing the pathlib module and copy the path to the batik motif folder on the drive and paste it in data_dir.

```
[ ]  #Membaca dataset
     import pathlib
     data_dir = "/content/drive/MyDrive/MOTIF_BATIK"
     data_dir = pathlib.Path(data_dir)
```

Program Code. 3. Read dataset

4. Displays the total number of data in the dataset, as follows, 1000 is the number of data in the dataset.

```
image_count = len(list(data_dir.glob('*/*.jpg')))
image_count
```

```
1000
```

Program Code. 4. Display the number of datasets

5. Showing one of the images of batik motifs, such as the following images of batik motifs from Rembang.

```
#tampilkan salah satu image
img = list(data_dir.glob('Rembang/*.jpg'))
PIL.Image.open(str(img[0]))
```



Program Code. 5. Display one image

6. Then configure training and validation and share dataset files for training and validation.

```
batch_size = 32
img_height = 225
img_width = 300

train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split = 0.2,
    subset = "training",
    seed = 123,
    image_size = (img_height, img_width),
    batch_size = batch_size
)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split = 0.2,
    subset = "validation",
    seed = 123,
    image_size = (img_height, img_width),
    batch_size = batch_size
)
```

```
Found 1009 files belonging to 10 classes.
Using 808 files for training.
Found 1009 files belonging to 10 classes.
Using 201 files for validation.
```

Program Code.6. File sharing for training and validation

7. Display the target class, which is 10 areas of each batik motif.

```
#Cari target Classes
class_names = train_ds.class_names
class_names
```

```
['Banyumas',
 'Cilacap',
 'Jepara',
 'Jogja',
 'Kebumen',
 'Pati',
 'Pekalongan',
 'Rembang',
 'Surakarta',
 'Tegal']
```

Program Code. 7. Class Display

8. Display batch images and batch labels in train ds. Image batch is a tensor of the shape (32, 225, 300,3). This is a collection of 32 images with a shape of 225x300x3 (the last dimension refers to the RGB color channel). Label_batch is a tensor of the shape (32,), it is a label corresponding to 32 images.

```
for image_batch, label_batch in train_ds:
  print(image_batch.shape)
  print(label_batch.shape)
  break
```

```
(32, 225, 300, 3)
(32,)
```

Program Code. 8. Display batch images and batch labels

9. Display of the model build, with num_classes = 10 that is 10 area of batik motifs. And use the model sequential, padding same and activation relu. The six lines of code below define a convolution base using a common pattern: Conv2D and MaxPooling2D layer stacks. As input, CNN takes the form tensor (image_height, image_width, color_channels), ignore batch size. Color_channels refers to (R,G,B). Configure CNN to process input shapes (256, 256, 3), which are batik motif image formats. Can be done by passing the input_shape argument to the first layer. To complete the model, we can insert the last output tensor of the convolution basis (of the form (4, 4, 64)) into one or more Dense layers to perform the classification. The Dense layer takes a vector as input (i.e. 1D), while the current output is a 3D tensor. First, we'll flatten (or unroll) the 3D output into 1D, then add one or more Dense layers on top. This model uses the Adam optimization. Adam optimizer and loss.SparseCategoricalCrossentropy loss function. To see training accuracy and validation for each training period, provide a metric argument[9].

```
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_13 (Rescaling)     (None, 225, 300, 3)       0
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 225, 300, 256)     7168
_________________________________________________________________
dropout_25 (Dropout)         (None, 225, 300, 256)     0
_________________________________________________________________
max_pooling2d_39 (MaxPooling (None, 112, 150, 256)     0
_________________________________________________________________
conv2d_40 (Conv2D)           (None, 111, 150, 128)     295040
_________________________________________________________________
max_pooling2d_40 (MaxPooling (None, 56, 75, 128)       0
_________________________________________________________________
conv2d_41 (Conv2D)           (None, 56, 75, 64)        73792
_________________________________________________________________
max_pooling2d_41 (MaxPooling (None, 28, 37, 64)        0
_________________________________________________________________
flatten_13 (Flatten)         (None, 66304)             0
_________________________________________________________________
dense_39 (Dense)             (None, 32)                2121760
_________________________________________________________________
dropout_26 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_40 (Dense)             (None, 64)                2112
_________________________________________________________________
dense_41 (Dense)             (None, 10)                650
=================================================================
Total params: 2,500,522
Trainable params: 2,500,522
Non-trainable params: 0
_________________________________________________________________
```

Program Code. 9. Model build view

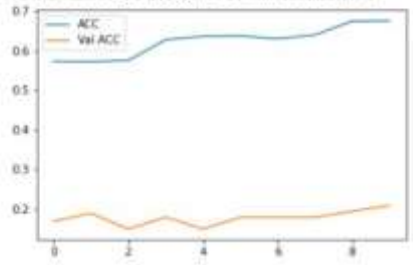
The image above explains that the output of each Conv2D and MaxPooling2D layer is a 3D tensor shape (height, width, channel). The width and height dimensions tend to shrink as you go deeper into the network. The number of output channels for each Conv2D layer is controlled by the first argument (e.g., 32 or 64). Usually, as width and height shrink, we can (computationally) add more output channels in each Conv2D layer. The grid summary shows that (56,

75, 64) the output is flattened into a shape vector (66304) before passing through the two Dense layers.

10. The results of the training, with 10 epochs and displaying the chart of the training. The evaluation of this batik motif detection model uses the accuracy evaluation model and ROC Curves.

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
epochs_range = range(epochs)
plt.plot(epochs_range, acc, label="ACC")
plt.plot(epochs_range, val_acc, label = "Val ACC")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f5c6a266a90>
```



Program Code. 10. Training display

11. The prediction test display or the system can display the inputted batik motif output.



Program Code. 11. Output batik motifs

12. The results of the detection of batik motif images that have been inputted are showing batik from Rembang.

```
img_array = keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array,0)

predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])
np.argmax(score)
print ("Batik: " , class_names[np.argmax(score)])
```

```
Batik:  Rembang
```

Program Code. 12. Batik motif detection results

5. Conclusion

Based on the results of the design and build of this batik motif detection system model, it can be concluded, namely:

1. This batik motif detection system can detect the origin of the batik motif from the inputted area quickly and well.
2. The inputted batik motif is successfully displayed as output, so that it can be detected.
3. This batik motif detection system has good accuracy and continues to improve.
4. The best temporary accuracy value is 0.6757 and the validation accuracy is 0.2090. The value of the accuracy results in this study can still be improved.

6. References

[1] *Wikipedia*, "Sistem," 2020. [Online]. Available: https://id.wikipedia.org/wiki/Sistem. [Accessed: Dec. 3, 2020].

[2] *Wikipedia*, "Batik," 2020. [Online]. Available: https://id.wikipedia.org/wiki/Batik. [Accessed: Dec. 3, 2020].

[3] T. A. P. Sidhi, B. Y. Dwiandiyanta, and F. K. S. Dewi, "Batik motifs detection using pattern recognition method," *Jurnal Buana Informatika*, vol. 11, no. 1, pp. 55–62, Apr. 2020, doi: 10.24002/jbi.v11i1.3234.

[4] M. C. Dewi, "Deep learning convolutional neural network untuk klasifikasi pengenalan objek menggunakan MxNet," 2018.

[5] R. Mawan, "Klasifikasi motif batik menggunakan convolutional neural network," *Jurnal Nanaloka*, vol. 1, no. 1, pp. 45–50, 2020, doi: 10.36802/jnanaloka.2020.v1-no1-45-50.

[6] Iskandar, "Batik sebagai identitas kultural bangsa Indonesia di era globalisasi," 2017.

[7] *TensorFlow*, "Image classification," 2020. [Online]. Available: https://www.tensorflow.org/tutorials/images/classification.

[8] *TensorFlow*, "Load and preprocess images," 2020. [Online]. Available: https://www.tensorflow.org/tutorials/load_data/images.

[9] *TensorFlow*, "Convolutional neural network (CNN)," 2020. [Online]. Available: https://www.tensorflow.org/tutorials/images/cnn.