# Development of The Thai Cloth Pattern Recognition Algorithm and Mobile Application

Nutteera Ummartyotin
Panishsa Charoenwong
Nattanun Aramchatmongkol

Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2018
KMITL-2018-IC-B-003-005

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

COPYRIGHT 2018
INTERNATIONAL COLLEGE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use

**Thesis - Academic Year 2018**

Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang

**Title:** Development of The Thai Cloth Pattern Recognition Algorithm and Mobile Application

**Authors:**


1. Nutteera Ummartyotin Student ID: 58090015
2. Panishsa Charoenwong Student ID: 58090024
3. Nattanun Aramchatmongkol Student ID: 58090067


King Mongkut's Institute of Technology Ladkrabang

Approved for Submission

..............................................

(Dr.Ukrit Watchareeruetai)
Advisor

Date 24/6/2019

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Acknowledgments

This project could not be completed if we did not have Dr. Ukrit Watchareeruetai as our advisor. We also would like to thank my colleagues for providing suggestions in the presentation and discussions. Moreover, we are thankful to our lecturers at International College for providing support and knowledge. We would like to give special thanks to our families for their support, motivation, and encouragement. However, for those who have not been mentioned, we thank them for supporting for us. This project could not have been completed without their collaboration.

King Mongkut's Institute of Technology Ladkrabang

This material is reserved for educational use only, not allowed for commercial use.



Forbidden to modify the content, and cite the document when use

# Abstract

Thai cloth is a well-known product of Thailand. It represents the beauty of Thai culture through its detailed and delicate patterns. Ratchaburi's Won Teen Chok is one such product where each pattern reveals a unique history and the livelihood of the people from that region of which has distinguishing features and names. However, the names of these patterns have been slowly forgotten and nowadays not many people know them.

Therefore, the authors aim to develop a system that helps young people, people interested in Thai culture, foreigners, weavers, artists, and artisans identify Thai cloth patterns and also contributes to the preservation of Thai culture.

The system can identify 8 different Ratchaburi's Won Teen Chok cloth patterns from the primary pattern of the cloth. A dataset consist of 1386 images of 8 Chok patterns and the deep learning model, ResNet-50, were implemented to recognize different primary patterns through various images the cloth's primary pattern and produce the name of the pattern as a result. Data augmentations were applied in order to increase the number of images in dataset and improve the accuracy of the model from 78.80% to 87.05%.

This material is reserved for educational use only, not allowed for commercial use.



Forbidden to modify the content, and cite the document when use

# Table of contents



This material is reserved for educational use only, not allowed for commercial use.



Forbidden to modify the content, and cite the document when use

King Mongkut's Institute of Technology Ladkrabang

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use

# List of figures



This material is reserved for educational use only, not allowed for commercial use.



Forbidden to modify the content, and cite the document when use

# List of tables



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use

# Chapter 1
# Introduction

## 1.1 Problem Description

Clothing is one of human's basic needs. It is something that has existed alongside us since even the historical times. Many types of clothes can be seen in various places and in different regions or climates, its appearance can vary significantly. Thai clothes are just one of the various categories of existing clothes but apart from the cloth itself, the pattern drawn or imprinted on it can diverse into multiple categories, each with its own origin. Not many people have knowledge on these various patterns, even the names alone are a hard work to find out.

Sometime we might encounter a particular pattern on a piece of cloth that is so fascinating we want to learn or look for more information, but only little to none of the information concerning it can be found Most of the time these clothes are just lying there for show or in some shop without anyone knowing the name of these patterns, let alone their origin. Sometimes that particular pattern might be so captivating you just want to know more of it but what you can do is to take a picture of the pattern and browse through and search for it by yourself. Searching for the pattern by oneself without even knowing what it is called is quite a difficult task and requires a lot of effort so most people would just ignore it and put it in the back of their heads. These situations can happen to designers who after seeing those patterns and developing some new ideas, need to do research and learn more about that particular pattern but end up struggling because no one even knows what the pattern is called, only the design of the pattern itself can be recognized.

This is the origin of the project idea. This project is developed in order to help solve these kinds of problems by building a tool that can be used to take pictures at any moment and use them to determine the cloth's pattern. This information is important for the users to find out more on that particular pattern on the cloth for any further usage



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

and can contribute to further projects. Moreover, this can assist in the campaign of the preservation and promotion of Thai culture and provide an easy access to Thai cloth with something that anyone can learn and use.

After we did some research and experimentation. To solve this problem, we decided to create a web application in front of the UI to get the input image of the Thai cloth from the person who wants to know the name of the pattern. Then, classify the pattern of the Thai cloth and return the result of pattern name as an output. In the background, we using the CNN model to train and classify the output.

## 1.2 Objective

In order to briefly specify the criteria for the project's success for this semester, several goals have been set. The following list shows all the goals which are needed to be satisfied.

- Predict the name of cloth using convolutional neural network (CNN) model
- Obtain at least 70% accuracy in the prediction outcomes of the pattern names
- Build a mobile application

## 1.3 Scope of Work

The scope of this thesis is listed as follows:

- The information about the cloth will be obtained from the Office of Sericulture Conservation and Standard Conformity Assessment, The Queen Sirikit Department of Sericulture, Ministry of Agriculture and Cooperatives
- The following Ratchaburi's Won Teen Chok main patterns shall be covered 8 designs :
  - Hak-Nok-Koo
  - Karb-Son-Hak
  - Kong-Keng
  - Kong-Keng-Son-Sia
  - Krab-Dok-Kaew



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

- Krab
- Dok-Sia
- Na-Mhon

## 1.4 Report Structure

In this report, there are seven chapter.The following content shall be covered 7 chapter:

- Chapter 2 Background Knowledge
- Chapter 3 Related Works
- Chapter 4 Proposed Methods
- Chapter 5 Implementation
- Chapter 6 Experimentations
- Chapter 7 Conclusion



King Mongkut's Institute of Technology Ladkrabang
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Chapter 2
# Background Knowledge

In this chapter, the background concepts that are related to this study will be briefly explained.

## 2.1 Convolutional Neural Network

In deep learning, there are many different architectures. A convolutional neural network (CNN) [4] is one of the most popular neural network architecture. Architecture of the network is inspired by the connectivity and structure of visual cortex in the brain. The network was given its name from the most important operation in the network which is convolution. It is most used for images recognition and classification because it does not require manual feature extraction. A CNN consist of three layers: input layer, hidden layers and output layer [5]. Figure 2.1 shows architecture of convolutional neural network.

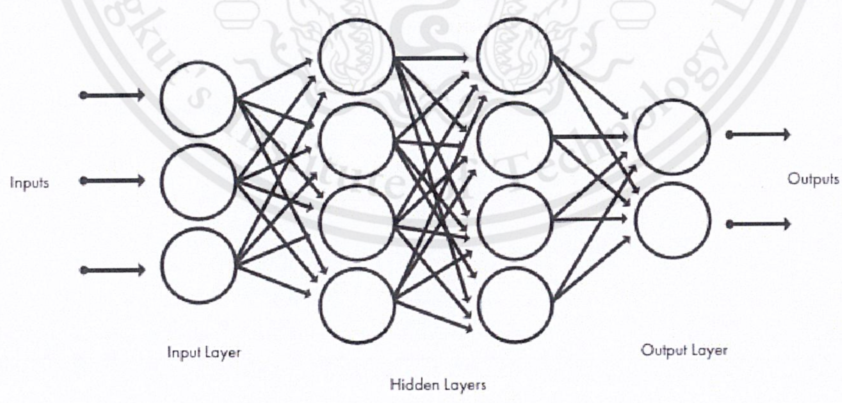



Figure 2.1: Architecture of convolutional neural network (Source: MathWorks, n.d.)

The output layer is a fully connected layer that classify the output. An approach to represent an output such as softmax takes output of the previous layer and determines



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

which features most correlate to a particular class. The number of nodes in the output layer typically equals to the number of class that the model have to predicted.

The hidden layers are usually composed of convolutional layers, pooling layers, fully connected layers [6].

### 2.1.1 Convolutional Layers

This layer applies a convolution operation to extract feature from the input image. Convolution of an image is performed by sliding the square matrix called filters or kernels over the input image to compute each element in the image. Different filters can perform different operations such as edge detection, blur and sharpen by changing the value in the filter. The result from convolution operation called Convolved Feature or Activation Map or Feature Map will pass through non-linear operation such as Rectified Linear Unit (ReLU), tanh, or sigmoid to make the output non-linear before passing to the next layer.

### 2.1.2 Pooling Layers

This layer simplifies the output from previous layer by reducing the number of parameters or weights that a network have to learn when the images are too large while retains the important information. This also reduces a computation in the network and controls overfitting. Many types of pooling are:

- Max Pooling - The most common approach. Max pooling takes the largest value from feature map of the previous layer.
- Average Pooling - Take the average value from feature map of the previous layer.
- Sum Pooling - Take the sum of all value from feature map of the previous layer.

### 2.1.3 Fully Connected Layers

The fully connected layer is also referred to as dense layer. It is a layer that all neuron in a layer is connected to every neuron in another layer. The fully connected layer will use the output from convolutional and pooling layers to classifying the input image into different classes in dataset. Furthermore, fully connected layer also combine the feature from convolutional and pooling layers to improve classification of the image.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

## 2.2 ResNet-50

Residual neural network 50 or ResNet-50 [16] is a pre-trained model which consists of 50 layers of convolutional layer. However, Deeper neural networks are more difficult to train due to vanishings gradients problem, where gradients begins to vanish exponentially as it propagates back result in slow training of the front layers. Therefore, ResNet solved this problem by implementing layer skips or shortcuts as shown in Figure 2.2. By skipping over layers, gradients can propagates through the shortcut allowing the earlier layers to train as well as speed up leaning as there are fewer layers for gradients to propagates through.

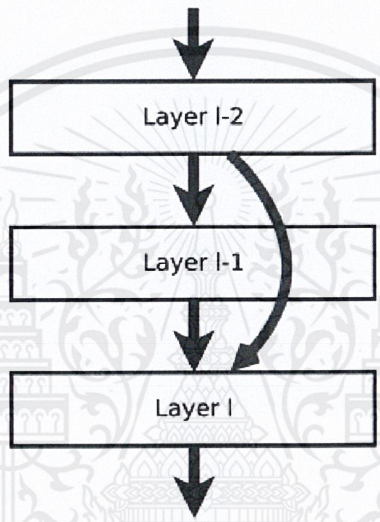



Figure 2.2: ResNet architechture (Source: drawn by John Erling Blad, 2017)



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

## 2.3 Transfer Learning

Transfer learning [14] is a machine learning method that transfer knowledge gained from solving one or more tasks and applied it to solve a new but related task. Transfer learning method aims to improve learning in this new task by applied knowledge gained from previous task.

According to Yosinski, Clune, Bengio, and Lipson, [15] The features learned in first layer of many deep neural networks trained on images are not specific to a task. Features that usually appears in first layer are similar to Garbor filters and feature blobs. These features are general and can be used with other task. In contrast, features in final layer of the neural network are specific to the task. Therefor, general features in the first layer of a trained neural network or a pre-trained model is transferable to another neural network.

Furthermore, the benefits of transfer learning method are as followed:

- decreases amount of time needed to train the neural network.
- Improves performance of the neural network.
- Eliminates the need of large dataset to train the neural network.

## 2.4 Data Augmentation

### 2.4.1 Geometric Transformations

Geometric Transformations [9] is one of method of image preprocessing to Rotation, size and shape change by change the coordinates of objects.

#### Scaling

Scaling is just resizing or changing the shape of the image. It is a polygonal handle by multiplying the coordinates $x, y$ of each vertex with expansion factor (scaling factor) $s_x$ and $s_y$ to create a new coordinates

$$x' = x \cdot s_x, y' = y \cdot s_y \tag{2.1}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.2}$$



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

Equation (4.3) is an equation from create new coordinates which x' is new coordinate of x by multiply of scaling factor of x($s_x$) and also y' is new coordinate of y by multiply of scaling factor of y($s_y$). Equation (4.4) is an equation of creating new coordinate in matrix form.

**Rotation**

2D rotation is achieved by changing the position of the object along the circular path on the plane.

The rotation angle, $\theta$, and the position of the rotation point is determined to be the center point of an image and It will rotate the object around this point.

Rotation of an image for an angle is achieved by the transformation matrix of the form in Eq. (4.5)

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.3}$$

### 2.4.2 Smoothing Images

**2D Convolution ( Image Filtering )**

As in one-dimensional signals, images also can be filtered with various low-pass filters (LPF), high-pass filters (HPF) etc. LPF helps in removing noises, blurring the images etc. HPF filters helps in finding edges in the images.

**Image Blurring (Image Smoothing)**

Image blurring is achieved by convolving the image with a low-pass filter kernel. It is useful for removing noises. It actually removes high frequency content (eg: noise, edges) from the image. So edges are blurred a little bit in this operation. (Well, there are blurring techniques which doesn't blur the edges too).

**Median Blurring**

Median Blurring takes median of all the pixels under kernel area and central element is replaced with this median value. This is highly effective against salt-and-pepper noise in the images. Interesting thing is that, in the above filters, central element is a newly calculated value which may be a pixel value in the image or a new value. But in median



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

blurring, central element is always replaced by some pixel value in the image. It reduces the noise effectively. Its kernel size should be a positive odd integer.

## 2.5 Training Convolutional Neural Network

Training of the model was achieved by backpropagration [7] approach. Firstly, the filters and weights were initiated with random values. The training image was then goes through convolution, non-linear, and pooling operation and finds the output probabilities for each class. For the first training image, weights are randomly assigned therefor the output probabilities are also random. The total error, $E_{total}$, will be calculated at the output layer from

$$E_{total} = \frac{1}{2}\sum(\mathrm{t}-\mathrm{y})^2 \tag{2.4}$$

Where t is the target output for a training sample, and y is is the actual output of the output neuron. The network then use backpropagation to calculate the gradients of $E_{total}$ with respect to weights between neurons i and j, $\omega_{ij}$, in the network by applying

$$\frac{\partial E_{total}}{\partial \omega_{ij}} = \frac{\partial E_{total}}{\partial out_j}\frac{\partial out_j}{\partial net_j}\frac{\partial net_j}{\partial \omega_{ij}} \tag{2.5}$$

Where $out_j$ denotes the output after application of the activation function, and $net_j$ denotes the output before application of the activation function. Then the weights were updated using gradient descent to minimize the error. Finally, repeat the following process with different training image until the error stops decreasing and thus optimized weight of the model to correctly classify images.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Chapter 3
# Related Works

There are numerous studies on cloth pattern recognition. This section mainly focuses on the related studies on cloth pattern recognition using machine learning algorithm.

## 3.1 Cloth Pattern Recognition with Four Features (RSSM)

Durga Rao, Sudhavani, Balakrishna, and Gouthami [1] proposed a novel Radon Transform Descriptor, Scale Invariant Feature Transform (SIFT), Mathematical Morphology based global feature and a schema to extract Statistical Descriptor to capture global features of clothing patterns. They combined the extracted global and local features to recognize clothing patterns by using a Support Vector Machines (SVMs) classifier. the features are extracted base on the data set on clothing pattern recognition including four-pattern categories plaid, striped, patternless, and irregular. Their approach achieves 96% recognition accuracy.

## 3.2 Batik Motif Classification Using Color-Texture-Based Feature Extraction and Backpropagation Neural Network

Suciati, Pratomo, and Purwitasari [2] used color-texture-based feature extraction and backpropagation neural network to develops a software to automatically identify motifs of batik image. Batik is an Indonesian's traditional cloth which has been recognized as one of the world cultural heritage. Currently, there are hundreds of different batik motif which can be classified into 7 groups. Their experiment achieves 0.37 in rate of Tanimoto Distance.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

## 3.3 Recognizing Clothes Patterns and Colours for Blind People Using Neural Network

Rao, Sudhavani, Balakrishna, and Gouthami [3] used the support vector machine algorithm for recognizing the pattern and the color. They obtained the features of the image and extracted features using the Radon Signature descriptor is to extract statistical properties, the wavelet subbands are used to extract global features of clothing patterns. They combined this with local features that are obtained from scale invariance feature transform to recognize complex clothing patterns.

## 3.4 Pattern Extraction from Northern Thai Fabrics Using Flexibly Matching Segments

Vasantapan and Chouvatut [12] used flexibly matching segment for detecting and recognizing 31 patterns of Sarong Teenjok textiles and 25 pattern of Lanna textiles. Their method improves robustness when detecting and recognizing a pattern with small deviation, orientation or different shadows and shades of the appeared traceries. Their experiment achieved 95.20 % of accuracy.

## 3.5 Loei Fabric Weaving Pattern Recognition Using Deep Neural Networks

Narong and Wichai [13] mention Loei Fabric are Thai national culture woven fabrics and each province has its own unique style of woven pattern. It is very hard to distinguish the pattern itself by just looking. This issues however can be solve by using deep neural network, which they use GoogLeNet (v1-v4). The result shows that the beast accuracy network is the Inception_v4 which has the lowest errors rate about 6.94%.

## 3.6 Conclusion

In five researches, the systems can recognize clothing patterns. The first and the third research papers developed model to recognize four-pattern categories plaid, striped, patternless, and irregular by using a Support Vector Machines (SVMs) classifier. And the second paper developed model to recognize seven categories of Batik motif by using



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

backpropagation neural network. The four and the fifth research papers develops algorithm to detect Thai cloth textiles. The fourth research used matching segment but the fifth research used deep neural network. But there is no research on using a convolutional neural network with transfer learning for detecting Thai cloth pattern. We expect our proposed method with will get better performance with user friendly mobile application.



King Mongkut's Institute of Technology Ladkrabang
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Chapter 4
# Proposed Methods

## 4.1 Predictive Model

The algorithm part of this system used a convolutional neural network ResNet-50 loaded with the pretrained ImageNet weights. The network has 50 layers of convolutional layer followed by 1024 ReLU neurons. The final pooling and fully connected layer in the original model was not included and the output layer has 8 nodes with softmax activation function. All layers were freeze except the last 4 layers. The network was compiled with adaptive moment estimation (Adam) optimization algorithm with cross-entropy cost function.

The model used in this system was build from Keras with Tensorflow as a backend. In Keras, neural network are defined as a sequence of layer. The Sequential class is the container for these layer. Begins by create an instance of the Sequential class.

```
model = sequential ()
```

Listing 4.1: Create an instance of the Sequential class

Resnet-50 model was defined and added into the model. The weight of is set to pretraining on ImageNet. The fully-connected layer at the top of the network was not included and input shape must be defined if include_top is set to false.The input shape should have exactly 3 inputs channels, width, and height.

```
res_conv = ResNet50(weights = "imagenet", include_top=False, input_shape = (img_cols,
    img_rows,3))
model.add(res_conv)
```

Listing 4.2: Defined and added Resnet-50 model

Because the early layer of the model usually correspond to general features while the last few layer is problem specific, All layers were freeze except for last 4 layers to prevent it weights from being modified.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

```
for layer in vgg_conv.layers[:-4]:
    layer.trainable = False
```

Listing 4.3: Freeze layers

The output or weight from convolutional layer must be flattened into 1-dimensional before passing them to fully connected layer. The fully connected layer has 1024 nodes, each node activated by ReLU function.

```
model.add(Flatten())
model.add(Dense(1024))
model.add(Activation('relu'))
model.add(Dropout(0.5))
```

Listing 4.4: The fully connected layer

The output layer must have a number of nodes corresponding to the number of class in the dataset. In this implementation, the out layer has 8 nodes with softmax classification.

```
model.add(Dense(8))
model.add(Activation('softmax'))
```

Listing 4.5: The output layer

The model can be compiled with Adam optimizers, cross entropy for loss function, and specify metrix to be evaluated by the model during training and testing.

```
model.compile(optimizer = tf.train.AdamOptimizer(),
 loss = 'categorical_crossentropy', metrics=['accuracy'] )
```

Listing 4.6: Compile the model

model.fit() method was called to train the model by passing training data, number of training epochs. Verbosity can be set by passing integer 0 = silent, 1 = progess bar, or 2 = one line per epoch [8].

```
model.fit(X_train, Y_train, epochs = 200, verbose =2)
```

Listing 4.7: Train the model



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

## 4.2 Source Code

The following is an overall code for implementing Resnet-50 predictive model in this system.

```
import tensorflow as tf
from tensorflow import keras
from keras. utils import np_utils
from keras.models import Sequential
from keras. layers . core import Dense, Dropout,
Activation , Flatten
from keras. optimizers import SGD,RMSprop,adam
from keras. applications import ResNet50

res_conv = ResNet50(weights = "imagenet", include_top=False,
input_shape = (img_cols,img_rows,3))

model = Sequential ()
model.add(res_conv)
for layer in res_conv. layers [:-4]:
    layer . trainable = False

model.add( Flatten () )
model.add(Dense(1024))
model.add( Activation ( 'relu ' ))
model.add(Dropout(0.5) )

model.add(Dense(num_classes))
model.add( Activation ('softmax'))

model.compile( optimizer=tf . train .AdamOptimizer(), loss=' categorical_crossentropy ' , metrics=['
    accuracy' ])
model. fit (X_train , y_train , epochs=num_epoch, verbose=2, callbacks =[ TestCallback (( X_test ,
    y_test )) ])
```

Listing 4.8: overall code for implementing Resnet-50 predictive model



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Chapter 5
# Implementation

## 5.1 System Overview

The result of this research will be presented using a system comprised of client-side and server-side. Figure 4.1 is a simple client-server architecture. The mobile application will be used to input images then sent to the server. After the server finishes processing the image, it will respond with the analyzed result back to the mobile application.


Figure 5.1: Method overview



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

## 5.2 Server Development

The server side is responsible for handling the input of Thai cloth image from a mobile application then classifying image by using the model that implement on the server. After sending an image to the model, the model will process and send the result of the prediction of the Thai cloth pattern back to the mobile application.

Client–server model [17]. is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

### 5.2.1 Client (Mobile Application)

A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

**React-Native**

React Native[18] to build native mobile apps using JavaScript and React.

### 5.2.2 Server

A server host runs one or more server programs which share their resources with clients.

**Flask**

Flask [19] is a microframework for Python based on Werkzeug, Jinja 2 and good intentions.

Werkzeug is a utility library for the Python programming language, in other words a toolkit for Web Server Gateway Interface (WSGI) applications, and is licensed under a BSD License. Werkzeug can realize software objects for request, response, and utility functions. It can be used to build a custom software framework on top of it.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

## 5.3 Application Features

Since the project will plan to use with server, the model will be implemented on server as well. The application will be implemented in Android Application with React-Native language.

In the home page of the application, where users can view the 3 buttons camera, gallery and instruction. User can choose to take a new image from their mobile phone camera or from their photos gallery. Then they must crop the image focus on main pattern, after that they can confirm the image. After classification by system, user will be shown the result.

Figure 5.2 shows the home page of the application, where users can choose the actions of the application that can see the information menu about the information of Thai cloth pattern on information page in Figure 5.3 or the instruction of how to use the application , classify the pattern by taking the photo on photo page in Figure 5.4 or classify the pattern by choosing an image from photo library on library page in Figure 5.5. In addition, if the user chooses to classify method to classify the pattern after the user chooses an image to the predicted process the application will return the predicted result on the Result page in Figure 5.6.

In the Home page, the user can click on the information button to link to the information page in Figure 5.7. To see the information of the application such as the instruction of how to use the application or the information about Thai cloth pattern.

In Home page, user can click on camera button to link to the Camera page to take a photo of Thai cloth that want to classify the pattern in Figure 5.8. or user can click on the library button to link to the Library page to choose an image of Thai cloth that wants to classify the pattern in Figure 5.9. In this both function the application will allow the user to crop an image to capture the main pattern in Figure 5.10. After the user chooses and crops an image the application will send an image to predict the pattern from the model on the server then reply back and show the result of Thai cloth pattern on the Result page in Figure 5.6.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use


Figure 5.2: Home page

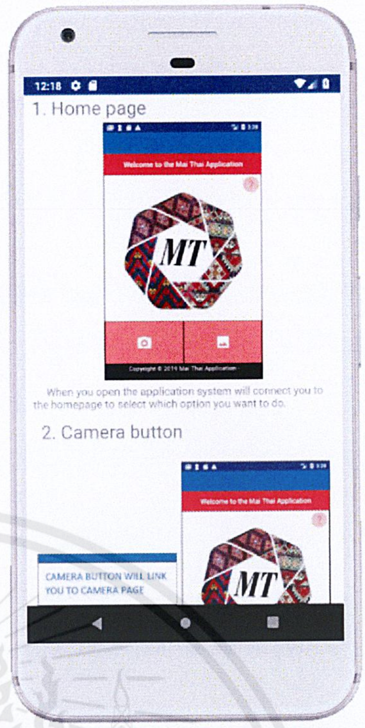



Figure 5.3: Information page

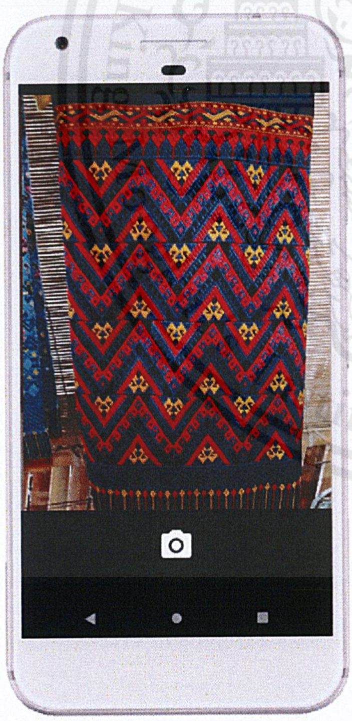


Figure 5.4: Camera page


Figure 5.5: Library page



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

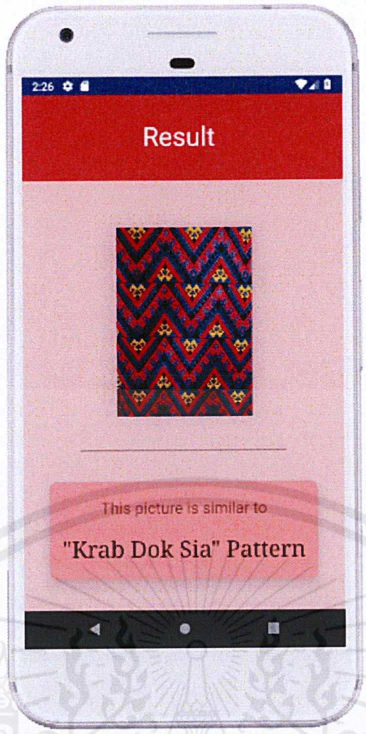



Figure 5.6: Result page

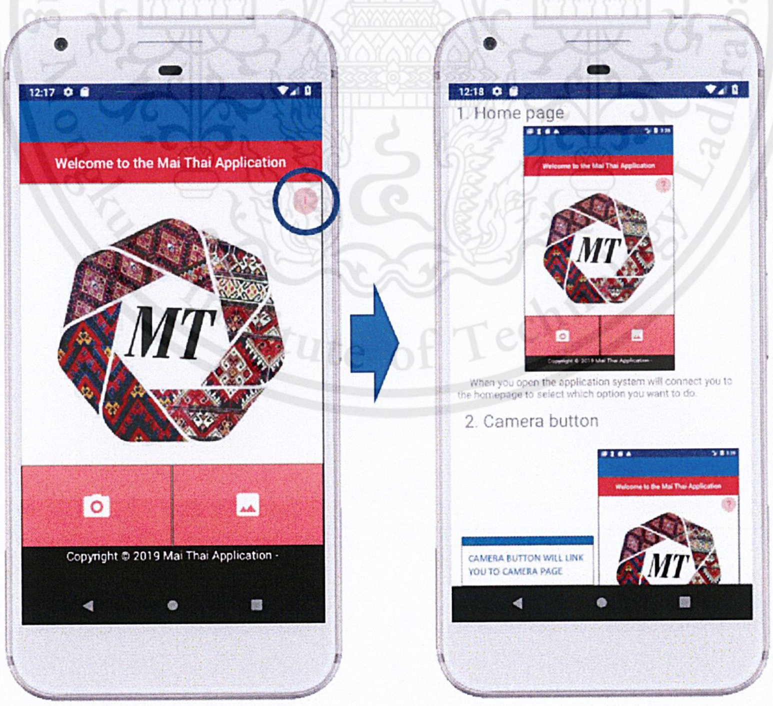



Figure 5.7: The information button link to the Information page



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

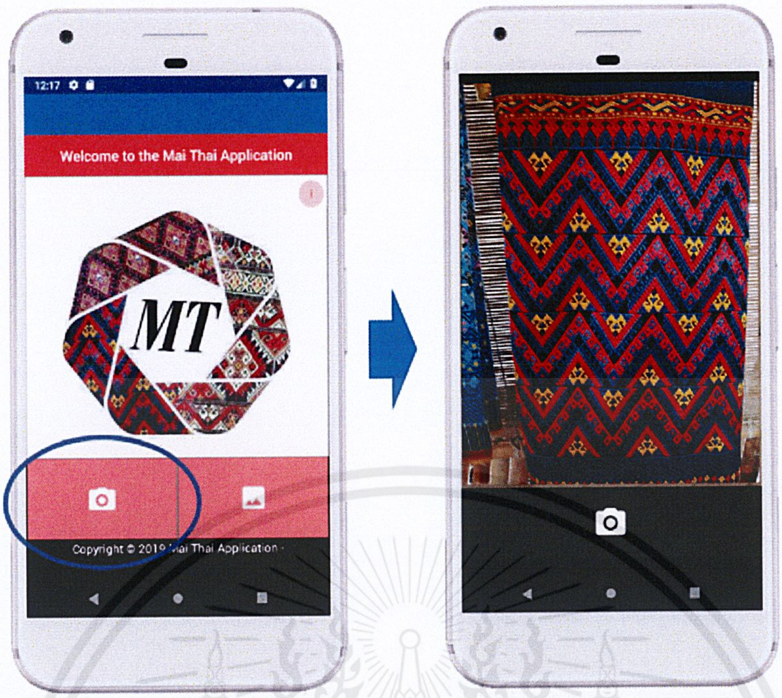



Figure 5.8: The camera button link to the Camera page

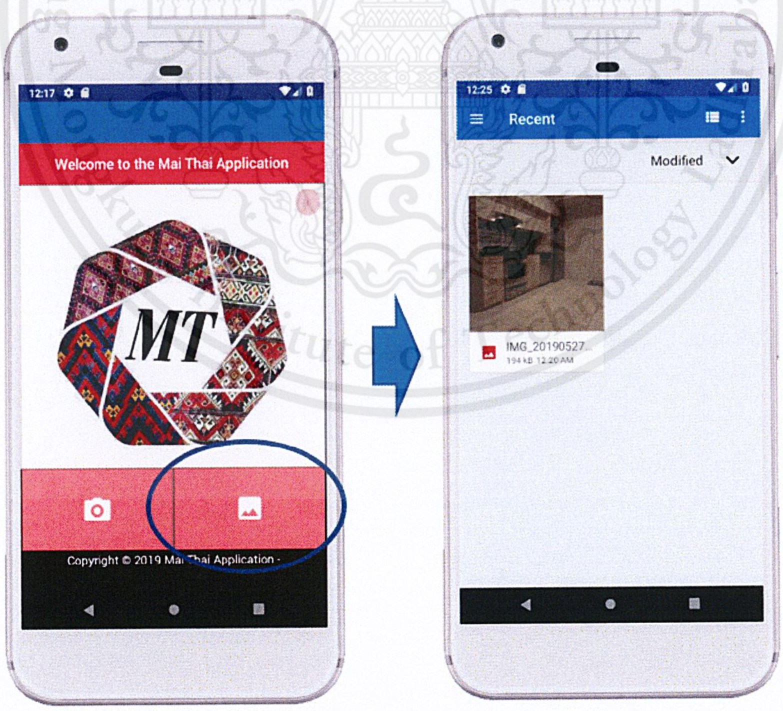



Figure 5.9: The library button link to the Library page



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

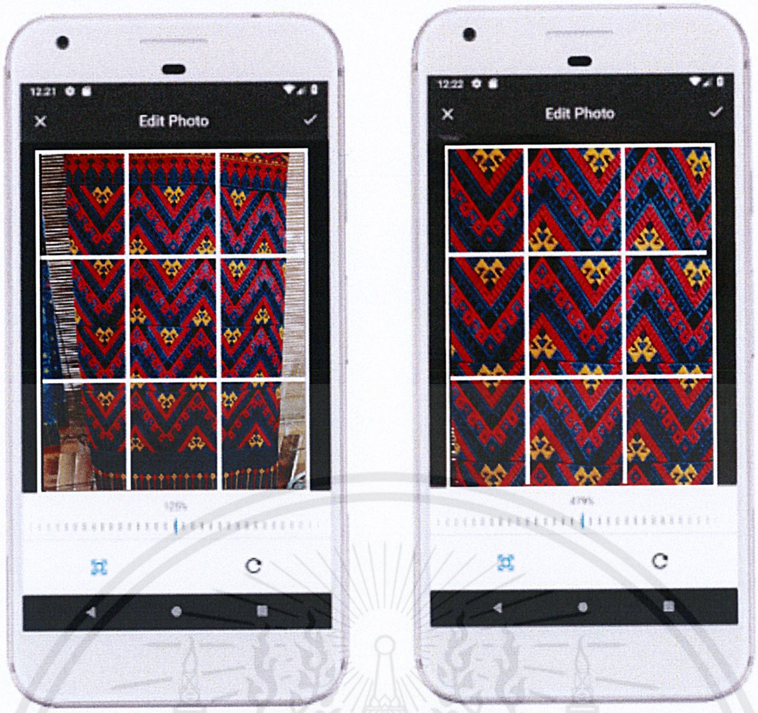



Figure 5.10: The crop function to allow capture the main pattern.

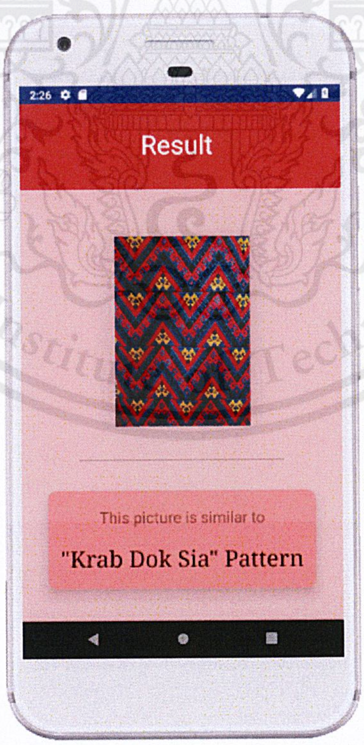



Figure 5.11: The result page to show the prediction of Thai cloth pattern



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# 5.4 System Diagram

## 5.4.1 Activity Diagram

Figure 5.12 shows activity of main function of system. Users starts by either select an image from in device or use camera to snap an new image. The user then view the crop page, the user then zoom and crop main pattern of cloth. The system predict name of the pattern and show result to user.

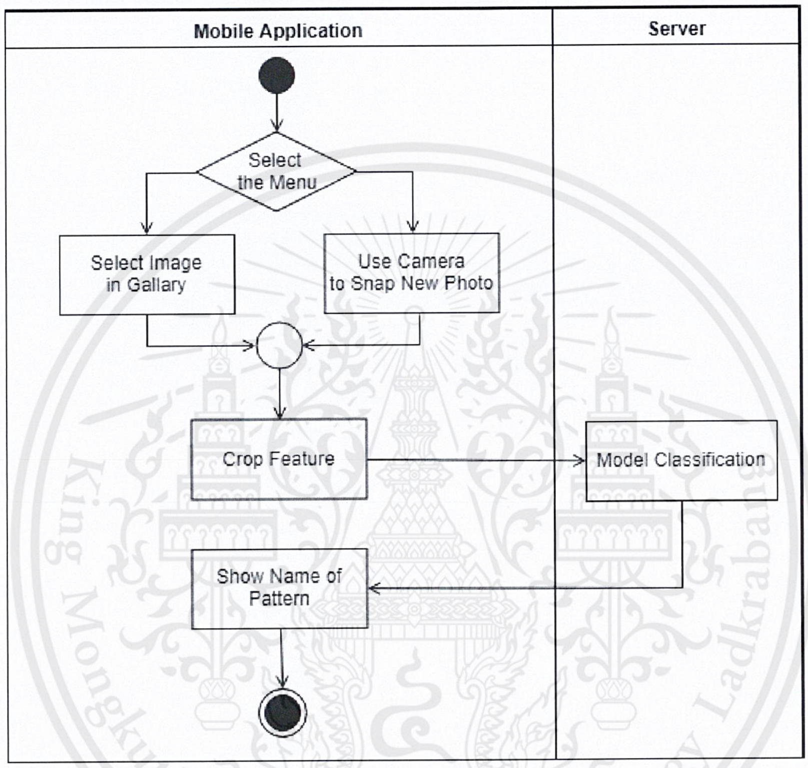



Figure 5.12: Activity diagram: Predict name of cloth pattern



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Chapter 6
# Experimentations

## 6.1 Dataset

We use Thai Won Teen Chok cloth as the data set. There are 8 main patterns of unique Chok designs of Tai Yuan which were passed down from ancient. The data was collected from Ban Khu Bua museun by digital camera, resulted in 1,386 images in total.

### 6.1.1 Original Dataset

**Na Mhon** is a main single design. The design is often used at the hem of Sin Teen Chok or on Chok pillow.

Figure 6.1: Example of Na-Mhon pattern



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

**Karb** is a main single design. It is similar to bract of bamboo. The design is often used at the hem of Sin Teen Chok or on Chok pillow.

Figure 6.2: Example of Karb pattern

**Dok Sia** is a main single design. The design is often used at the hem of Sin Teen Chok or on Chok pillow.

Figure 6.3: Example of Dok Sia pattern



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

**Krab Dok Kaew** is a main single design. The design is often used at the hem of Sin Teen Chok or on Chok pillow.

Figure 6.4: Example of Karb Dok Kaew pattern

**Kong Keng Son Sia** is a main design mixing between Kong Keng and Dok Sia pattern. The design is often used at the hem of Sin Teen Chok and shawl.

Figure 6.5: Example of Kong Keng Son Sia pattern



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

**Kong Keng** is a main single design. The design is often used at the hem of Sin Teen Chok or Sin Ta.

Figure 6.6: Example of Kong Keng pattern

**Karb Son Hak** is a main design mixing between Karb and Hak pattern. The design is often used at the hem of Sin Teen Chok.

Figure 6.7: Example of Karb Son Hak pattern



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

**Hak nok Koo** is a main design mixing between Nok Koo and Hak pattern. The design is often used at the hem of Sin Teen Chok.

Figure 6.8: Example of Hak nok Koo pattern

According to The pattern of Ratchaburi's Won Teen Chok pattern shall be covered 8 main patterns, each patterns is done through the 3 method of data augmentation which is Scaling, Rotation and Median blur in order to expand the dataset.

### 6.1.2 Scaling

The purpose of using scaling is to scale an image in difference size of an image to be our dataset in difference scale of shape of an image. To prevent blur and resolution of an image.

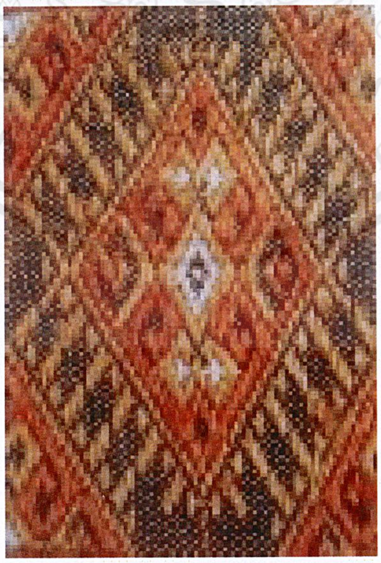


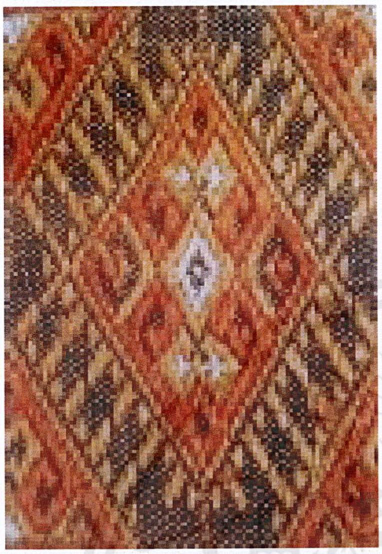
Figure 6.9: a) Original image with Scaling 0.2 and b) Random generate crop.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

### 6.1.3 Rotation

The purpose of using rotation is to be rotate an image in different position of an angle along the circular path and create a set of data. To prevent the angle of input that can be occur in different position of an angle.



a) Original image

b) Original image with Rotation 58°.

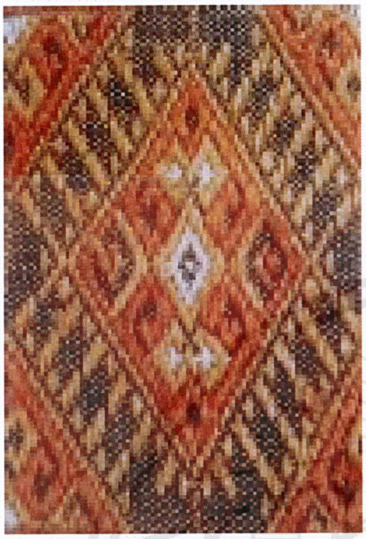
Figure 6.10: a) Original image, b) Original image with Rotation 58°



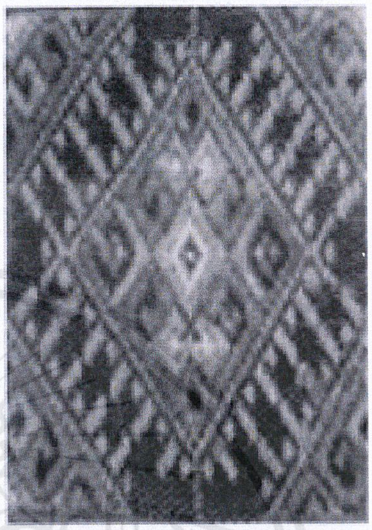
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

### 6.1.4 Median Blur

The purpose of using median blurring is to be smoothing an image in help to removing noises, blurring an images etc. especially effective against in salt-and-pepper noise in the images.



a) Original image



b) Original image with MedianBlur.

Figure 6.11: a) Original image and b) Original image with MedianBlur

## 6.2 Set Up

The predictive model used in this experiment was installed on a PC with Intel Core i5, 8GB RAM, and Microsoft Windows 10 as an operating system. Under Python3.5 environment, the predictive model was implemented by using Keras with Tensorflow as a backend. This predictive model is ResNet-50. It consist of 50 convolutional layers, followed by fully-connected layer with 1024 rectified linear units (ReLU) neurons. The output layer has 8 nodes with softmax activation function. All layers were freeze except the last 4 layers.

The dataset has 8 categories corresponding to 8 pattern of Chok designs. The number of images for each pattern is shown in Table 6.1

OpenCV and Numpy library were utilized for resizing images into 85x85 pixels, and perform other image processing. Scikit-learn was also utilized to shuffle and split dataset into training set for 80% and testing set for 20% of total number of images in the dataset, resulting in 1,109 training images and 277 testing images.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

Table 6.1: Number of images for each category of cloth pattern

| Category | Number of images |
|---|---|
| Na Mhon | 168 |
| Karb | 120 |
| Dok Sia | 204 |
| Karb Dok Kaew | 190 |
| Kong Keng Son Sia | 166 |
| Kong Keng | 120 |
| Karb Son Hak | 209 |
| Hak nok Koo | 209 |
| Total | 1386 |

## 6.3 Result and Discussion

The ResNet-50 model was first trained with the 1,249 images of size 48x48 pixels. It obtained training accuracy and testing accuracy of 97.80% and 78.80% respectively at training epochs set to 100 as shown in Table 6.2.

Table 6.2: Training and Testing accuracy of ResNet-50 with input image of size 48x48 pixels and training epoch set to 100

| Epoch | Training accuracy (%) | Testing accuracy (%) |
|---|---|---|
| 10 | 92.09 | 74.40 |
| 20 | 93.79 | 77.60 |
| 30 | 96.30 | 76.80 |
| 40 | 97.10 | 76.40 |
| 50 | 97.70 | 73.60 |
| 60 | 97.20 | 76.40 |
| 70 | 98.00 | 77.60 |
| 80 | 96.50 | 78.00 |
| 90 | 98.30 | 81.60 |
| 100 | 97.80 | 78.80 |

The next experiment was conducted in order to find appropriate input image size by training the model with images of size 85x85, 100x100 and 125x125 as shown in Table 6.3, Table 6.4, and Table 6.5 respectively. The input image of size 85x85 provided the provide the best result with training accuracy of 92.19% and testing accuracy of 82.40% at 100 training epochs. The lower training and testing accuracy of input image of size 100x100 and 125x125 possibly due to insufficient training time and complexity of the model.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

Table 6.3: Training and Testing accuracy of ResNet-50 with with input image of size 85x85 pixels and training epoch set to 100

| Epoch | Training accuracy (%) | Testing accuracy (%) |
|---|---|---|
| 10 | 85.69 | 74.40 |
| 20 | 86.89 | 82.40 |
| 30 | 89.59 | 82.00 |
| 40 | 89.29 | 83.60 |
| 50 | 86.69 | 79.60 |
| 60 | 89.79 | 79.60 |
| 70 | 91.69 | 83.20 |
| 80 | 93.09 | 84.40 |
| 90 | 91.99 | 82.80 |
| 100 | 92.19 | 82.40 |

Table 6.4: Training and Testing accuracy of ResNet-50 with with input image of size 100x100 pixels and training epoch set to 100

| Epoch | Training accuracy (%) | Testing accuracy (%) |
|---|---|---|
| 10 | 60.26 | 64.00 |
| 20 | 70.07 | 70.80 |
| 30 | 72.87 | 71.60 |
| 40 | 73.97 | 69.20 |
| 50 | 76.98 | 72.40 |
| 60 | 75.78 | 72.80 |
| 70 | 78.28 | 72.80 |
| 80 | 78.48 | 74.40 |
| 90 | 77.58 | 70.40 |
| 100 | 79.78 | 77.20 |

Table 6.5: Training and Testing accuracy of ResNet-50 with with input image of size 125x125 pixels and training epoch set to 100

| Epoch | Training accuracy (%) | Testing accuracy (%) |
|---|---|---|
| 10 | 5385 | 5600 |
| 20 | 5506 | 5760 |
| 30 | 53.85 | 49.60 |
| 40 | 57.96 | 61.20 |
| 50 | 57.76 | 60.80 |
| 60 | 57.06 | 58.40 |
| 70 | 58.36 | 62.00 |
| 80 | 57.86 | 57.60 |
| 90 | 58.26 | 59.60 |
| 100 | 59.56 | 61.20 |



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

Then The model was trained with input images of size 85x85 pixels and training epoch set to 100. Number of images in dataset was also increased from 1,249 to 1,386 images. Table 6.6 shows training accuracy and testing accuracy score of the model. The increased number of images help improve the training performance, resulted in training accuracy of 97.56% and testing accuracy of 87.05%.

Table 6.6: Training and Testing accuracy of ResNet-50 with with input image of size 85x85 pixels and training epoch set to 100

| Epoch | Training accuracy (%) | Testing accuracy (%) |
|---|---|---|
| 10 | 91.61 | 85.25 |
| 20 | 96.12 | 88.49 |
| 30 | 95.22 | 84.17 |
| 40 | 95.94 | 89.93 |
| 50 | 97.29 | 87.41 |
| 60 | 97.38 | 89.93 |
| 70 | 97.47 | 89.93 |
| 80 | 96.93 | 90.29 |
| 90 | 96.66 | 87.77 |
| 100 | 97.56 | 87.05 |



King Mongkut's Institute of Technology Ladkrabang
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

Table 6.7 shows the confusion matrix of ResNet-50 on 80 test images, each class contain 10 images.

Table 6.7: Confusion matrix of the model on 80 test images

| | | Actual Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Dok Sia | Hak Nok Koo | Krab Son Hak | Kong Keng | Kong Keng Son Sia | Krab | Krab Dok Kaew | Na Mhon |
| Predicted Class | Dok Sia | **9** | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Hak Nok Koo | 0 | **5** | 0 | 1 | 0 | 2 | 0 | 0 |
| | Krab Son Hak | 1 | 3 | **9** | 0 | 0 | 4 | 1 | 0 |
| | Kong Keng | 0 | 0 | 1 | **7** | 4 | 0 | 0 | 2 |
| | Kong Keng Son Sia | 0 | 0 | 0 | 2 | **6** | 0 | 0 | 0 |
| | Krab | 0 | 0 | 0 | 0 | 0 | **4** | 4 | 1 |
| | Krab Dok Kaew | 0 | 0 | 0 | 0 | 0 | 0 | **4** | 0 |
| | Na Mhon | 0 | 0 | 0 | 0 | 0 | 0 | 1 | **7** |

According to the confusion matrix, the model has difficulty in classifying Hak Nok Koo, Krab and Krab Dok Keaw because the these pattern are similar to each other. This also true with Kong Keng and Kong keng Son Sia.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Chapter 7
# Conclusion

Detailed and delicate patterns of Ratchaburi's Won Teen Chok have been slowly forgotten and nowadays not many people know them. Therefore, the objective of this system is to classify the eight different Chok patterns and also contribute to the preservation of Thai culture.

The propose system uses the deep learning model ResNet-50 with transfer learning techniques to recognize the patterns. Due to limited size of dataset of only 1248 images, the model was trained with 100 training epochs and obtained only 78.80% accuracy. Data augmentation techniques, including scaling, rotation, and median blur was then applied to increase the number of images to 1386 images. The model imporved significantly and obtained an accuracy of 87.05%.

For further improvement, different data augmentation technique can be applied to increase the size of dataset and make distinguishable features of each pattern more identifiable. A different deep learning algorithm such as YOLO can also be applied to detect multiple patterns in the cloth.



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

# Bibliography


[1] Durga-Rao, N., Sudhavani, G., Balakrishna, P., and Gouthami, K. (2005). *Cloth pattern recognition with four features (RSSM)*. [Online]. Available: `https://ieeexplore.ieee.org/document/7377315`

[2] Suciati, N., Pratomo, W., A., and Purwitasari, D. (2014). *Batik Motif Classification Using Color-Texture-Based Feature Extraction and Backpropagation Neural Network*. [Online]. Available: `https://ieeexplore.ieee.org/document/6913352`

[3] Thilagavathi, B., and Jarin, J. (2015). *Recognizing clothes patterns and colours for blind people using neural network*. [Online]. Available: `https://ieeexplore.ieee.org/document/7193006`

[4] *Convolutional Neural Network*. [Online]. Available: `https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html`

[5] Ujjwal, K. (2016). *An Intuitive Explanation of Convolutional Neural Networks*. [Online]. Available: `https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets`

[6] Murphy, J. (2016). *An Overview of Convolutional Neural Network Architectures for Deep Learning*. [Online]. Available: `https://pdfs.semanticscholar.org/64db/333bb1b830f937b47d786921af4a6c2b3233.pdf`

[7] Rojas, R. (1996). *Neural Networks - A Systematic Introduction*. Retrieved from `http://page.mi.fu-berlin.de/rojas/neural/index.html`

[8] *Model class API*. [Online]. Available: `https://keras.io/models/model/`

[9] *Geometric Transformations of Images*. [Online]. Available: `https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_geometric_transformations/py_geometric_transformations.html`




This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

[10] *Smoothing Images*. [Online]. Available: https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html

[11] *Phathai*. [Online]. Available: http://phathai.tripod.com/html/Phathai1.html

[12] Nattha, V., and Varin, C. (2017). *Pattern extraction from northern Thai fabrics using flexibly matching segments: Sarong Teenjok and Lanna textiles*. [Online]. Available: https://ieeexplore.ieee.org/document/7886125

[13] Narong, B., and Wichai, P. (2018). *Loei Fabric Weaving Pattern Recognition Using Deep Neural Network*. [Online]. Available: https://ieeexplore.ieee.org/document/8457365

[14] West, J., Ventura, D., and Warnick, S. (2007). *Spring Research Presentation: A Theoretical Foundation for Inductive Transfer*. [Abstract].

[15] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). *How transferable are features in deep neural networks?*. [Online]. Available: https://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf

[16] He, K., Zhang, X., Ren, S., and Sun, J. (2015). *Deep Residual Learning for Image Recognition*. [Online]. Available: https://arxiv.org/abs/1512.03385

[17] *Client–server model*. [Online]. Available: https://en.wikipedia.org/wiki/Client%E2%80%93server_model

[18] *React Native*. [Online]. Available: https://facebook.github.io/react-native/

[19] *Flask (web framework)*. [Online]. Available: https://en.wikipedia.org/wiki/Flask_(web_framework)



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use

[20] *Convolutional Neural Network 3 things you need to know*. [Online]. Available: ://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use