BHUTANESE HANDWOVEN TEXTILE PATTERN RECOGNITION AND CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

UGYEN CHODEN

A Thesis Submitted to the Graduate School of Naresuan University
in Partial Fulfillment of the Requirements
for the Master of Engineering in Computer Engineering - (Type A 2)
2021

BHUTANESE HANDWOVEN TEXTILE PATTERN RECOGNITION AND CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

UGYEN CHODEN

A Thesis Submitted to the Graduate School of Naresuan University
in Partial Fulfillment of the Requirements
for the Master of Engineering in Computer Engineering - (Type A 2)
2021

Thesis entitled "Bhutanese Handwoven Textile Pattern Recognition and Classification Using Artificial Neural Networks"

By UGYEN CHODEN

has been approved by the Graduate School as partial fulfillment of the requirements for the Master of Engineering in Computer Engineering - (Type A 2) of Naresuan University

Oral Defense Committee

.... Chair
(Choochart Haruechaiyasak, Ph.D.)

.... Advisor
(Assistant Professor Panomkhawn Riyamongkol, Ph.D.)

.... Internal Examiner
(Assistant Professor Phongphun Kijsanayothin, Ph.D.)

.... Internal Examiner
(Associate Professor Panus Nattharith, Ph.D)

Approved

....
(Associate Professor Krongkarn Chootip, Ph.D.)

Dean of the Graduate School

Title	BHUTANESE HANDWOVEN TEXTILE PATTERN RECOGNITION AND CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS
Author	UGYEN CHODEN
Advisor	Assistant Professor Panomkhawn Riyamongkol, Ph.D.
Academic Paper	M.Eng. Thesis in Computer Engineering - (Type A 2), Naresuan University, 2021
Keywords	Bhutanese Handwoven Textile, Convolutional Neural Network, Pattern Recognition, BHTP dataset, BTT dataset, Machine Learning

ABSTRACT

Bhutanese textiles symbolize the country's unity and autonomy through cultural aesthetics and distinctive features of traditional wear. The weavers and the experts can distinguish and recognize the differences in patterns instantly. However, most Bhutanese individuals, particularly the youths and the foreigners find it difficult to recognize the differences in it. Identification of such unique textiles is often learnt through practice, making it a particularly rigorous and costly form of learning, resulting in discontent among people who perform it. Bhutan also has a scarcity of digital archives and paper materials for textile identification and future references. This study aims to develop machine learning models that are capable of recognizing and classifying the Bhutanese textile types and their' patterns using Bhutanese textile datasets.

In this study, the first Bhutanese textile datasets were developed in this work using various augmentation approaches. Bhutanese handwoven textile pattern (BHTP) and Bhutanese textile types (BTT) were generated as two separate datasets. The datasets were used to test various textile recognitions models.

A PatternNet model was proposed for recognizing and classifying Bhutanese handwoven textile patterns into 10 classes. Using the CNN (Convolutional Neural Network) concept, a six-layered PatternNet model was developed and trained

on the BTP dataset. PatternNet model outperformed other models such as VGG16, AlexNet, ResNet-34, ResNet-50, SVM, and KNN with an accuracy of 99.75% and 99.25% for training and validation respectively. Similarly, VGG16 architecture outperformed KNN, SVM and AlexNet in recognition and classification of 7 types of Bhutanese textiles with the accuracy of 99% and 98.33% for training and validation respectively. PatternNet model was deployed on the flask web application and it was tested with 550 input images. The testing accuracy of the PatternNet model deployed on web application was 98.54%.

This is the first study to emphasize the feasibility of using machine learning to classify the types and patterns of Bhutanese textiles. Machine learning techniques have been shown to be effective in the development of applications of Computer Vision to make Bhutanese textile learning openly accessible. This study can be expanded to detect the defects present in textiles and generate new textile patterns or motifs.

ACKNOWLEDGEMENTS

I pay my heartfelt gratitude to His Majesty the King Jigme Khesar Namgyel Wangchuck of Bhutan and the Faculty of Engineering, Naresuan University, Thailand for providing me with this opportunity to pursue my master's degree. It is a great honour and privilege for me to be one of the recipients of the His Majesty's (HM) and Naresuan University (NU) scholarship to continue my education. I remain grateful to the Royal University of Bhutan and Gyalpozhing College of Information and Technology for giving me this opportunity to continue my education.

I am sincerely grateful to my thesis advisor, Assistant Professor Dr. Panomkhawn Riyamongkol for making this research possible. I am also grateful to Mr. Rattapoom Waranusast, a Lecturer at the Faculty of Engineering, who guided and supported me throughout the duration of my thesis. Their suggestions, prompt assistance, and direction helped me get through all the stages of my research. I am extremely thankful to my thesis committee members Associate Professor Dr. Panas Nattharith and Associate Professor Dr. Phongphun Kijsanayothin for making my thesis defence a pleasant experience, as well as for your insightful comments and ideas. I am also thankful to the Vision lab members for their unfailing assistance and encouragement. I am grateful to my seniors, Mr. Yonten Jamtsho (GCIT) and Mr. Karma Wangchuk (CST), for getting me started on my research, teaching me everything from the scratch, and leading me through all the processes required for a successful study period. Furthermore, I would like to acknowledge the Bhutanese textile stores in Thimphu and my Bhutanese friends who contributed images of their wardrobes to my dataset. I am deeply grateful to the Engineering department, as well as the faculty secretaries and staff, who have worked tirelessly to ensure the study runs smoothly. I am extending my appreciation to the Division of International Affairs and Language Development (DIALD) staff for their assistance in making my stay here comfortable and convenient. I will be eternally grateful to my beloved parents, relatives, and friends for their unwavering support and understanding throughout my entire academic career.

UGYEN CHODEN

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

BACKGROUND

1.1 Introduction: Background and Significance of the study

Bhutan is a tiny country located in the Eastern Himalayas between two gigantic countries China and India. It covers an area of 38,394 square kilometres with 71% of forest coverage and the population was projected to be 748,931 by 2020 (Bureau, 2020). Until the prominence of the Wangchuck dynasty, Bhutan had been ruled by a Tibetan lama, Zhabdrung Ngawang Namgyal. He unified Bhutan into a single state in 1634 and founded the traditional dress of Bhutan (Lo et al., 2016). In 1907, the civil and monastic body crowned King Ugyen Wangchuck as the first King of Bhutan. He was then succeeded by his son Jigme Dorji Wangchuck as the second king of Bhutan. King Jigme Dorji Wangchuck also known as the father of modernization reigned from 1929 to 1972. The fourth Druk Gyalpo King Jigme Singye Wangchuck received a modern education and became a king at the age of 17 in 1974. During his reign, he introduced the concept of Gross National Happiness. The country is currently a constitutional monarchy with the reigning monarch Druk Gyalpo Jigme Khesar Namgyel Wangchuck, the fifth king.

The great fourth propounded the concept of Gross National Happiness (GNH) and declared 'Gross National Happiness is more important than the Gross Domestic Product'. GNH is a unique development philosophy that guides its development plans, emphasizing a holistic and inclusive approach to sustainable development. The four pillars of GNH are good governance, sustainable socio-economic development, preservation, promotion of culture, and conserving the environment. GNH is measured with help of four indicators and nine domains (Verma, & Ura, 2018). One of the vital aspects of promoting the legacy of GNH is by preserving and promoting our unique culture & tradition such as festivals, arts and crafts, stories, and songs.

Bhutan is known as one of the popular tourist destinations due to the country's well-preserved culture and natural heritage. As per the annual report of Bhutan Tourism Monitor Bhutan (2019), Bhutan received a total of 315,599 visitors in 2019, up 15.14% from the previous year. There were 72,199 international visitors, a 0.55% increase over 2018. 99.39% of the tourists were known to have visited Bhutan for cultural sightseeing. Revenue from the tourism industry had increased from over USD 2 million in the late 1980s to over USD 85.41 million in 2018 (Bureau, 2020).

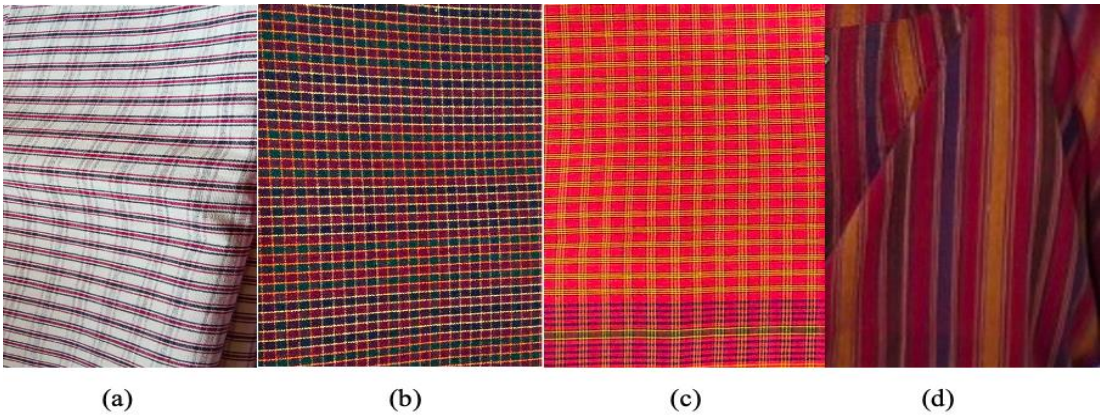


Figure 1 Types of Karchang Pattern

Figure 2 Types of Metochem Pattern

Bhutanese textiles represent the country's rich and sophisticated culture and tradition. They are broadly categorized into two groups: *karchang* and *metochem*, which can be further divided into various types as shown in Figure 1 and Figure 2 respectively. *Karchang* is a traditional garment with simple stripes and patterns that can be worn at home or work. *Metochem* is a densely patterned and costly fabric that is worn on special occasions such as festivals, weddings, and new year celebrations. (a) *Pangtse*, *(b)mathra*, *(c) sethra* and *(d) adang mathra* are classified as *karchang* while *(a) aikapur*, *(b) shinglochem* and (c) *kushuthara* are classified as *metochem*.

(a) (b) (c) (d)

Figure 3 Sample of Significant Bhutanese Textile pattern

Every textile item is given a name based on the colour, style, pattern, and fibre used while weaving. The patterns have moral, spiritual, and religious significance attached to them. The weavers generally draw inspiration from their surroundings and religion when creating patterns (Yaganegi, 2014). Figure 3 depicts some of the patterns as well as their significance. The pattern *Karma* (a) or a star resembles a butterfly. *Yunrung* (b) or Swastika is a symbol of total stability found in both Hinduism and Buddhism. *Shinglo* (c) or a life tree resembles a tree that represents longevity. The *Dorji Jadram* (d) or a double thunderbolt is a powerful symbol of peace, self-awareness, and the four elements. Similarly, all the patterns found on Bhutanese textiles have their meaning and significance.

The national dress of Bhutan is *gho* and *kira* worn by men and women respectively. It is made of a variety of materials and features a variety of designs and motifs. All Bhutanese must wear *gho* and *kira* in offices, monasteries, schools, festivals, and on other special occasions. A *gho* is a knee-length robe that is wrapped around the body and tied at the waist with a belt-like woven cloth called *kera*. Men often wear a long-sleeved or short-sleeved white jacket called *tego* beneath the gho.

Kira is an apron-like dress with various fabric designs and motifs wrapped around the body that is long and ankle-length. It is draped over the shoulder and fastens with a pair of *koma* on both sides. *Koma* is a compilation of two silver and silver-gilt clips with turquoise insets as shown in Figure 4. The pair are joined by a ring known as *jabtha*. The *kira* is then tied well at the waist with the *kera* as shown in Figure 4 (left). *Tego* is a long-sleeved jacket-like dress worn outside of *kira*. *Wonju* is a long-sleeved blouse, thinner than *tego* worn under the *kira*. The *half kira* is another type of *kira* that is worn without the *koma*, robed, and wrapped around the waist with the *kera*. Some men also wear a half *gho*, which is a shortened piece that is worn from the waist to the knees without the *tego*. During the summer season, *half gho* and *half kira* are commonly worn; however, *half gho* is not considered formal attire. The sample Bhutanese traditional attire for men and women is depicted in Figure 5.

Figure 4 Kera (left) and Koma (right)

Figure 5 Sample image of Bhutanese traditional attire

The rich tradition and cultural aesthetics are some of the reasons for tourism attraction in the country (Kaewkhunok, 2018). Bhutan gives priority to promoting and preserving the country's culture and heritage. The promotion of arts and crafts happened after the arrival of Zhabdrung Ngawang Namgyal in 1616 AD. Bhutanese art is a unique cultural tradition that is categorized into thirteen different traditional arts known as *Zorig Chusum*. It is comprised of thirteen different traditional arts such as *dzazo* (pottery), *jimzo* (sculpture), *chakzo* (blacksmithing), *lugzo* (metal casting), *shingzo* (carpentry), *troezo* (gold- and silversmithing), *tshemzo* (tailoring and tapestry), *thagzo* (weaving), *tsharzo* (bamboo and container work), *lhazo* (painting), *shagzo* (woodturning and lacquering), *yigzo* (calligraphy), and *dozo* (masonry) (Tshering, & Wangchuk, 2008).

A textile is a piece of fabric produced by hand or machine knitting or weaving. Weaving (*Thagzo*) is one of the oldest and most commonly used skills in the production of textiles that have been practised for ages in Bhutan. Bhutanese textiles have a long history, dating back to the 17th century, and it is gaining increasing recognition and visibility in this century. The royal patronage has been playing a vital role in promoting and promulgating Bhutanese arts and crafts (Wangchuk, 2016). The Royal Textile Academy and Textile Museum were set up under the patronage of Her Majesty Ashi Sangay Choden Wangchuck in Thimphu for demonstration, collection and transmission of information on textiles (Yaganegi, 2014). Bhutanese hand-woven textiles are famous for their rich tone, refined and assortments of examples including the many-sided fibre readiness, colouring, and weaving procedures. Bhutanese take pride in this rich and complex craft of weaving. Each intricate pattern has a name based on its tone, pattern, and fibre with certain strict significance.

Bhutanese weavers weave in two broad categories, *karchang* and *metochem* which can be further classified into different categories. The patterned woven textile is known as *methochem* and plain-woven textiles as *Karchang*. *Karchang* contains simple stripes and patterns and is worn at home or during daily work. *Metochem* is densely patterned and expensive and is worn on special occasions like festivals, marriages, and celebrations. Fibres such as raw cotton, wool, nettle fibre, yak hair and bura (raw silk) are used in weaving the Bhutanese textiles. The name of the textile is based on its colour, design, pattern, and fibre. The art of weaving is associated with religion and textile represents prestige, artistic skills, discipline, familial sentiments, and expression of devotion towards the dharma.

Bhutan is committed to preserving and promoting the country's identity by maintaining and promoting its heritage and culture. The National Museum in Paro Ta Dzong was established in 1965 to conserve, protect, create, and promote cultural heritage (Bhutan). It conducts exhibitions, symposiums, and publications to conserve ancient artefacts. In 2001 and 2005, the Textile Museum and Royal Textile Academy (RTA) respectively opened. Its mission is to conserve, encourage, and educate the public about Bhutanese textiles. RTA provides professional training in Bhutanese textile weaving, design, and development (Academy, 2016). Many more textile and handicraft shops are now open and accessible to local people as well as to tourists. However, the Bhutanese hand-woven textiles are hardly seen.

1.2 Purpose of study

This research discusses the following purposes:

1.2.1 Preparation of datasets, Bhutanese hand-woven textile patterns (dataset 1) and Bhutanese Textile Types (dataset 2).

1.2.2 Apply image processing and Machine Learning techniques to recognize and classify the Bhutanese hand-woven textile patterns and Bhutanese textile types.

1.2.3 Find a suitable algorithm for identification, recognition, and classification of Bhutanese textile patterns into various pattern categories and classification of Bhutanese textiles into different types.

1.2.4 Deploy the best model for Bhutanese hand-woven textile pattern recognition and classification.

1.3 Statement of Problem

Bhutanese textile is known for its vibrant colours, complex fibre preparation, dyeing, weaving techniques, and variety of patterns. However, machine-made clothes from India with Bhutanese textile designs have gained popularity in Bhutan in recent years (Yudon, 2015). This has led to duplication and replication of textile designs putting the country's distinctive identity at risk.

Some of the ancient designs are no longer woven, they are found only in museums and on antique textiles. The antique patterns such as the lungta (horse) and other animal figures, and torma figures are rare nowadays (Yaganegi, 2014). This is due to a lack of expertise and interest in weaving.

As per Yudon (2015), weaving is predominantly a women-centric skill and only a few Bhutanese men are seen weaving. Most of the weavers are women from the eastern region of the country known as *Sharchops*. Weaving skills and experience have been passed down through the generations, from mother to daughter and teacher to student. However, with the advent of modern schooling, most youths are enrolled in schools, universities, or employed in organizations, making it impossible for them to learn how to weave from their parents or relatives. As a result, the youth are ignorant of the basics of weaving and its various patterns. There are not many interested weavers participating in the course in the textile industry.

The problems and challenges encountered by the Bhutanese youth are probably the most concerning for the country's national identity (Walcott, 2011). With the spread of globalization, westernization, and modernization in the country, there is a great deal of concern for the continuity of this appealing and enduring tradition of weaving. Technology such as television, mobile phones, computers, and the internet are important tools for transformation, regeneration, and preservation. Bhutanese youth's interests have changed from traditional to western clothing due to the rapid influence of westernization. These factors are eroding and challenging Bhutan's traditional culture, which is regarded as a valuable asset.

Bhutanese Textiles are integral in promoting the tourism industry and every Bhutanese should know their importance. Tourists could also learn about the existence of such textiles in Bhutan and appreciate their beautiful and delicate art of them. As textile production transcends from the art of weaving clothes to the creative expression of individuals and communities, tourists gravitate to such intuitive textile design. And with the increasing number of tourists every year, it has become crucial that tourist guides have adequate knowledge on textile's origin, importance, and name.

Bhutanese textile designs and their importance have been the subject of numerous studies. However, no one has come up with a way to make it easier to recognize different patterns and designs. Thus, this study is being conducted to recognize and familiarize the Bhutanese textiles to conserve our culture and heritage. This study is emphasized using an Artificial Neural Network to detect and recognize Bhutanese textile patterns.

1.4 The scope of study

This research covers the following scopes:

1.4.1 Develop a Bhutanese Hand-Woven Textile Pattern (BHTP) and Bhutanese Textile Types (BTT) datasets containing 10 and 7 different classes respectively. The patterns were selected based on their significance and the data availability.

1.4.2 Collect data (images) from Facebook pages and Instagram accounts, textile shops in Thimphu, and the wardrobes of random participants.

1.4.3 Apply image position augmentation and colour augmentation techniques. The augmentation techniques include image inverting, adding light, adding saturation, average blurring, image sharpening, grayscale, median blur, bilateral blur, erosion, dilation, padding, vertical and horizontal flip, rotation (90 clockwise and 90 anti-clockwise), salt and pepper noise, Gaussian noise.

1.4.4 Train neural network algorithms (AlexNet, VGG16 and traditional CNN model) and Machine Learning models (K-Nearest Neighbour and Support Vector Machine) on the datasets.

1.4.5 Identify the best learning model for recognizing and classifying the 10 hand-woven Bhutanese textile patterns.

1.4.6 Identify the best learning model for recognizing and classifying the Bhutanese textile into 7 different types.

1.4.7 For training and testing the model for the BHTP dataset, full pattern images should be taken close to the camera with appropriate lighting without any disorientations or folding.

1.4.8 Deploy best model (PatternNet) for classification of BHTP on a web application.

1.4.9 Test the deployed model.

CHAPTER II

LITERATURE REVIEW

2.1 Related studies

Various studies have been being conducted on textile pattern recognition. Regular band analysis was used by Chan et al. (2017) to detect floating yarns in photographs of ancient Chinese textiles. Machine learning methods were used in the classification of Batik textile patterns, including Scale Invariant Feature Transform (SIFT) by Nurhaida et al. (2015), Multi Texton Co-occurrence Descriptor (MTCD) with Support Vector Machine (SVM) by Minarno, Azhar, et al. (2020) and Gray Level Co-occurrence Matrix (GLCM) with K-Nearest Neighbour (KNN) by Andrian et al. (2019).

2.2 Pattern Recognition

An occurrence of a group of objects, events, or concepts where the components of the group are almost identical to one another in certain ways or aspects is called a pattern. A pattern can be a human face, a speech signal, a fingerprint image, a barcode, a design on garments or a human gait (Sharma & Kaur, 2013). In the textile context, a pattern can be defined as frequently repeated orders, particularly a design from repeated lines, shapes, or colours on a fabric surface. Pattern recognition is a branch of Machine Learning that focuses on recognizing, analysing, and classifying the patterns and regularities in the data using machine learning algorithms. Pattern recognition is applicable in fields such as multimedia data retrieval, data mining, bioinformatics, remote sensing, biometric recognition, and document classification (Kaur, 2014). The general pattern recognition process involves three stages: data acquisition, pattern extraction and pattern classification (Hu et al., 2021). The four best approaches for pattern recognition are template matching, statistical approach, syntactic/structural approach, and neural networks.

2.2.1 Template Matching

Template matching is one of the earliest and simplest approaches for pattern recognition (Jain et al., 2000). It is widely used to find the resemblance between two objects or entities; it may be points, curves or pixels to localize and identify the patterns. The pattern to be recognized or identified is available as a template or prototype. Each pixel of the template is compared against the stored images to recognize the pattern. The main shortcoming of this approach is that it will go wrong if the patterns are distorted due to image processing and natural stretches, highly oriented or skewed patterns and large intra-class variation among the patterns (Vasantapan, & Chouvatut, 2017).

2.2.2 Statistical Approach

In the statistical approach, the categories of patterns are predefined. It is based on statistics and probabilities. Each pattern is represented as a point in a *d*-dimensional space and is described in terms of '*d*' attributes or measurements. A feature differentiates one pattern from another, each feature is different in each dimensional space. The goal is to select those unique features that let the patterns in different categories capture compact and disjoint regions in a *d*-dimensional feature space. A decision boundary is established in feature space to distinguish patterns belonging to various classes or categories. For the statistical approach, the decision boundaries are determined by a probability distribution. The efficiency of the feature set is evaluated by the ability to recognize patterns from distinct categories (Jain et al., 2000).

2.2.3 Syntactic Approach

This approach is also known as the structural approach for pattern recognition and is based on the relationship between features. Patterns are represented by a hierarchical structure that consists of sub-patterns that consider more complex relations among features. Sub-patterns to be recognized are known as *primitives* that will appear as sentences of language (Kaur, 2014). Patterns are considered as language sentences, primitives are viewed as language alphabets, and sentences are formed according to grammar (Jain et al., 2000). A small number of primitives and grammatical descriptions can be used to define patterns, with the grammars of each pattern drawn from the available training samples.

2.2.4 Neural Network

The fundamental characteristics of neural networks are the ability to understand sophisticated nonlinear relationships between input and output, use sequential training techniques, and adapt to the data (Sharma & Kaur, 2013). A neural network, also known as an artificial neural network (ANN), is a self-versatile teachable cycle capable of learning and resolving complicated problems based on available data. An ANN-based framework works like the human brain functions; it is made from interconnected handling components that reproduce neurons. A neural network is made up of three layers: an input layer, a hidden layer, and an output layer. It has neurons or nodes which are connected by weighted edges. Every neuron can communicate with another by utilizing the interconnection. ANN models attempt to organize weighted coordinated charts using hierarchical standards such as learning, speculation, adaptivity, adaptation to internal failure and appropriated portrayal, and calculation, in which artificial neurons frame the model's hubs and the coordinated edges (with weights) are associations between neuron yields and neuron inputs. The weights applied to the associations result from the learning cycle and demonstrate the significance of the commitment of the first neuron in the data being passed to the accompanying neuron.

2.3 Stages involved in Pattern Recognition

The entire process of pattern recognition is divided into four stages: data acquisition, data pre-processing, feature extraction and pattern recognition.

2.3.1 Data acquisition

The data for the model training is gathered using cameras in the environment. If CNN is used, the quality and quantity of the dataset affect the accuracy of the models. Mobile cameras and other digital cameras are commonly used to capture textile images. Fabric images come from a variety of places, including internet crawling, online stores, and textile stores (Rasyidi & Bariyah, 2020). The woven fabric texture was captured with a Nikon D5600 digital camera and a micro NIKKOR lens with a focal length of 45 mm (Iqbal Hussain et al., 2020). For their study, (Meng et al., 2021) introduced a new portable wireless device that captures fabric images in RGB. For capturing the images of goat leather, (Sousa et al., 2020) used a structure with additional lighting and fluorescent lights, as well as a Canon camera. For image acquisition, authors use digital cameras with various specifications depending on the study's needs.

2.3.2 Data pre-processing

Data pre-processing includes conversion to a suitable format, resizing and pre-processing the images. Data augmentation includes the addition of pixels and colours, top and black hat, performing the morphological transformation, blurring, saturation, dilation, erosion, flipping and rotating images. Data pre-processing is normally performed for image transformations and expansions of datasets (Tang et al., 2020). Lemley et al. (2017) used smart augmentation to boost the accuracy of their gender classification task. Smart augmentation used a network that learned to produce augmented data during the training. The output of one experiment, however, was worsened after using the smart augmentation technique. To solve the overfitting problem in pattern recognition and classification, (Feng et al., 2018; Rasyidi & Bariyah, 2020) performed data augmentation and achieved 94.8% and 99% accuracy respectively. Similarly, Iqbal Hussain et al. (2020) applied data augmentation techniques on a dataset to train a CNN model for woven fabric pattern recognition and classification. A total of 11328 images were generated from 708 original images by applying augmentation techniques and achieved a model accuracy of 99.3%. Data augmentation can not only expand the datasets to overcome overfitting problems, but it can also help improve classification model accuracy (Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019; Umer et al., 2021).

2.3.3 Feature Extraction

Feature extraction is a systematic process that identifies the related structure details found in a pattern to make the task of classifying the pattern simpler. Features of an image include contour, shape, rotation, angle, coordinates, motions, and background data. Feature extraction is a way of dimensionality reduction used in pattern recognition and image processing (Liang et al., 2017). The primary aim of feature extraction is to extract the most significant information from the original data and express it in a lower-dimensional space. Template matching, Deformable models, Unitary Image transformations, Graph definition, Projection Histograms, Contour profiles, Zoning, Geometric moment invariants, Zernike Moments, Spline curve approximation, Fourier descriptors, Gradient function, and Gabor feature are some of the most frequently used feature extraction methods (Kumar & Bhatia, 2014).

Many more feature extraction methods are being used for textile pattern recognition. For Batik pattern recognition, Nurhaida et al. (2015) used Scale Invariant Feature Transform (SIFT) with an error rate of 8.47% and Andrian et al. (2019) used Gray Level Co-occurrence Matrix (GLCM) with 97.96% accuracy. The application of GLCM outperformed prior works in a batik classification experiment performed by (Minarno, Sumadi, et al., 2020). Multi Texton Co-occurrence Descriptor (MTCD) by Minarno, Azhar, et al. (2020) and achieved an accuracy of 99%. Meng et al. (2020) achieved 92.97% of accuracy to recognize yarn location and weaving patterns in textiles by using Multi-Task Multi-Scale Neural Network (MTMSnet). da Silva BarrosM et al. (2020) used CNN as a feature extractor in the identification of textile fabrics. Puarungroj and Boonsirisumpun (2019) also used CNN for hand-woven fabric motif recognition and achieved the highest accuracy of 98.223% in classification.

2.3.4 Pattern Recognition

Pattern recognition is the final stage of image processing. After analysing the features, the class detection and recognition decision are made by assigning a label. Textile patterns are either recognized by using the extracted features or by using the machine learning classifiers.

Photometric differential analysis was designed and used by Li et al. (2020) and achieved an accuracy of 97.22% in recognizing the weave patterns. Histogram equalization and adaptive wiener filtering were used to highlight the yarn profile, followed by gradient pyramid fusion of images captured under four different lighting conditions. The template matching method was implemented with an accuracy of 95.20% (Vasantapan & Chouvatut, 2017). To recognize a similar pattern, a template pattern was chosen and then matched with the rest of the input images.

Machine learning methods are being used in the classifications of textile patterns, including K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Artificial Neural Network (ANN), Bayes Classifier. (Minarno, Sumadi, et al., 2020) compared K-nearest neighbour (KNN) and Support Vector Method (SVM) in the classification of the batik fabric pattern. The results revealed that SVM is effective and reliable with an accuracy of 92.3% obtained from cross-validation testing. For the classification of textile fabrics from mobile images, Naive Bayes, Random Forest (RF), SVM, and Multi-layer Perceptron (MLP) were used, with SVM outperforming the others with 94% accuracy (da Silva BarrosM et al., 2020).

In recent times, image processing and pattern recognition applications have been developed using machine learning classifiers such as Convolutional Neural Network and Hidden Markov Model.

2.4 Textile Pattern Recognition Algorithms

Textile pattern recognition algorithms have advanced tremendously. The shift from supervised to unsupervised recognition has begun. The pattern recognition algorithms in Machine Learning can be generally classified into supervised and unsupervised algorithms.

2.4.1 Unsupervised approach

The unsupervised algorithms make use of unlabelled training and testing sets. To make predictions, they look for patterns in the data and group them based on how similar their characteristics are like dimensions. For unsupervised pattern recognition, K-means clustering approaches namely fuzzy logic and crisp logic are used (Strączkiewicz et al., 2016). Clustering is a pattern recognition technique that uses an unsupervised approach.

An algorithm for automatic unsupervised patterned fabric defect detection was presented by (Hamdi et al., 2018). To classify defected blocks in a pattern fabric image, this algorithm employed standard deviation filtering and K-means clustering. The simulation results confirmed that the algorithm successfully defines the blocks containing the defects with an accuracy of 95%.

Using texture features of GLCM, Jing et al. (2015) used Kernel fuzzy c-means Clustering (KFCM) algorithm to classify weave patterns. The weave repeat of the weave diagram was obtained using the improved distance matching function (IDMF), which is then used to correct error floats and increase the accuracy of the identification result. The study concluded with the highest error rate of 2.8% in recognition of weave points.

Meng et al. (2021) clustered the colours of fabric yarns by introducing a density-based colour clustering algorithm (DBCCA). Every float was located, and its colour information was extracted by using a new multi-task and multi-scale convoluted neural network (MTMSnet). Then a DBCCA was used to cluster the colour of yarns. Finally, by combining the weave pattern and the colour clustering results, the layout of the coloured yarns and the colour effect was identified. The fabric density and weave pattern were also obtained during the process.

The Automatic Feedback Error-Correcting Colour-Weave Pattern Recognition Algorithm (AFEC) was presented by Di et al. (2013) to detect and automatically analyse original fabric images. To detect the skew angles of warp and weft threads, the Hough Transform was applied. A steering filter was then used to improve the original fabric in the direction of the obtained skew angles. The boundary of warp and weft yarns was obtained after processing with Canny edge detection and a local optimal removal filter. Meanwhile, an X-means algorithm was used to automatically classify the fabric's colours. To identify the logic embedded in the incomplete weave pattern, the Infill algorithm was used. A Rectification algorithm was then used to fix errors in the recognition of colour and weave patterns that may have occurred in previous components by contrasting the colour and weave patterns of the fabric.

However, in a limited number of textile pattern recognition trials, unsupervised methods were used, and the findings were less reliable.

2.4.2 Supervised approach

A supervised learning algorithm can predict the outcomes of unlabelled data by learning from labelled training data (Strączkiewicz et al., 2016). Classification is a supervised approach to pattern recognition. To train the mapping function, the labels of the training dataset are properly assigned. Then the mapping function is fed the input to generate the desired output. Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Artificial Neural Network (ANN) are some of the supervised algorithms used for textile pattern recognition.

Salem, & Abdelkrim (2020) used SVM to classify woven fabric defects. A dataset of 500 images was used, with five classes: *Hole*, *Kink*, *Missing Weft, Oil Satin*, and *no defect*. The classification accuracies were 98.25% when using GLCM as a feature extractor and 97.25% when using LBP as a feature extractor.

Khamket, & Surinta (2020) performed a classification of Thai silk pattern image recognition. Texture features and local features were created and then applied to SVM and KNN classifiers for the classification. SVM with LBP performed better in the Silk-Diff-Pattern dataset whereas the KNN classifier with Local Binary Pattern (LBP) performed better in the Silk-Pattern dataset. However, the authors concluded that deep learning architecture achieves higher accuracy in pattern recognition.

To solve Thai Loei woven fabric pattern recognition, Boonsirisumpun and Puarungroj (2018) used a deep neural network. It was intended to assist the machine in automatically recognizing patterns and for visitors to gain a better understanding of Thai culture. The authors demonstrated the classification of patterns using GoogleNet or Inception v1, v2, v3, and v4. The study focused on four different types of patterns. A dataset of 720 images belonging to 4 classes is used for the research. The test on Inception-v4 yielded the best results, with a 93.06% accuracy. According to the authors, the model's accuracy was influenced by inter-class similarity.

The same authors Puarungroj and Boonsirisumpun (2019) increased the number of categories in their training pattern image dataset from 4 to 25 to cover more fabric designs available in the Loei local woven group. To identify the patterns, researchers used Inception-v3 and MobileNets. With an accuracy rate of 98.223% and 93.208%, respectively, the results indicated that MobileNets outperformed Inception-v3. The authors concluded that their sample data could be further expanded to improve the model's results.

Studies on textile pattern recognition that used supervised learning methods yielded better results, but the textile diversity could still be increased.

2.5 Convolutional Neural Networks (CNN)

Deep learning is currently the fastest-growing area in the field of machine learning (ML) and (DNN). Convolutional Neural Networks (CNN) are currently the most widely used method for image analysis and classification among many DNN structures (Mikołajczyk & Grochowski, 2018). Deep neural networks and their corresponding learning algorithms face a few significant challenges, despite their accomplishments and prospects.

The convolutional neural network is a multilayer neural network, which made is up of learnable neurons and learnable weights. The nodes or neurons take certain output/s of previous layers as an input. The weights can be considered as *n* x *n* filters where n must be greater than the input size. These filters are often called kernels and it is convolved with each input. In recent years, in the fields of object recognition, tracking, and, in particular, image processing, convolutional neural networks (CNN), which can recognize patterns in images, have achieved remarkable efficiency (Iqbal Hussain et al., 2020). Convolutional Neural Networks (CNN) models have been used to identify and classify hand-woven textile patterns with high performances.

Boonsirisumpun, & Puarungroj (2018) used a deep neural network to automatically recognize Thai textile patterns. The authors compared GoogleNet or Inception v1, v2, v3, and v4 for pattern classification. The analysis looked at four different kinds of patterns. With a 93.06% accuracy, the tests on Inception-v4 provided the best performance.

To cover more fabric designs available in the Loei local woven category, the same authors Puarungroj and Boonsirisumpun (2019) increased the number of pattern categories in their training pattern image dataset, which now has 25 categories. The pattern recognition was done with Inception-v3 and MobileNet. MobileNets outperformed Inception-v3 with a 98.223 % accuracy score. The authors concluded that their sample data could be extended to boost the model's accuracy after discovering that inter-class similarity influenced the model's accuracy.

Multi-Task Multi-Scale Neural Network (MTMSnet) was used in a study by Meng et al. (2020) to recognize yarn location and weaving patterns in textiles. Ohi et al. (2021) suggested a FabricNet neural network model for fibre recognition, but the research accuracy necessitated a greater dataset size. However, only a few categories of textiles were used for the research.

2.6 Recent Trends in Textile Pattern Recognition

The application of image recognition has had a huge effect on people's lives. Image recognition can convert physical images into information that can be used as a source for automated development (Fu, 2020). Researchers were able to recognize different patterns and colours due to the evolution of algorithms and the ability to extract spatiotemporal information from textile images. Researchers have been focusing on garment texture design classification (Islam et al., 2016), fabric pattern recognition and classification (Boonsirisumpun & Puarungroj, 2018), fabric defect detection (Wei et al., 2018) and fabric colour clustering (Meng et al., 2021) of the various textiles.

Defect identification, pattern classification, and recognition of textile fabrics have been the focus of research in recent years. Sousa et al. (2020) focused on defect detection and quality classification of blue goatskin textiles. The accuracy rate for detecting defects in wet blue goatskin was 95.9%, and 93.3% for assessing the quality standard. CNN was used by Rasyidi and Bariyah (2020) to identify and recognize six distinct Batik patterns. To increase the variations in the dataset and prevent overfitting, 994 images were collected from six categories and image augmentation was performed. The dataset was used to train the DenseNet network architecture, which achieved a 94% accuracy and a top-2 accuracy of 99%.

Zhang et al. (2020) applied the Hyperspectral imaging system (HIS) for colour measurement of printed fabrics. The key clusters were established using the self-organizing map (SOM) algorithm, and the optimal number of colours and cluster merging were calculated using the density peaks clustering (DPC) algorithm with Silhouette Index. This algorithm not only determined the optimal number of colours for printed cloth and achieved proper colour segmentation, but it also proved to be time-efficient.

Meng et al. (2020) used the Multi-Task Multi-Scale Neural Network (MTMSnet) to identify yarn position and weaving patterns in textiles. Later, the ability to identify the colour layout of textiles was introduced, and it was discovered that it can automatically distinguish the basic structure parameters with high effectiveness and robustness.

The research also looked at how to identify printed fabric colour patterns to generate new fabric designs (Zheng, 2021). For motif unit tilting and detection, the local statistical corner features and Markov random field model were applied. The motif unit elements were segmented, detected, localized, and saved into a database, from which new fabric patterns were created. Peng et al. (2021) used deep neural networks to identify fabric categories from small videos with an accuracy of 98.75%.

CHAPTER III

METHODOLOGY

3.1 Introduction

This chapter discusses the thorough working of the neural network architecture for classification and recognition of Bhutanese Textile Patterns. Figure 6 depicts the four major phases of pattern recognition: data acquisition, data pre-processing, feature extraction, and pattern recognition. For reliable pattern detection and prediction systems, each step is critical. The framework requirement is illustrated first, followed by a flowchart of the entire system. The parts that follow include a comprehensive block diagram for each step.

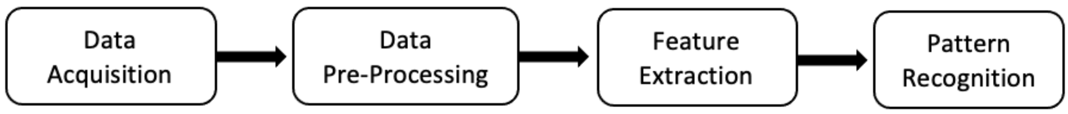


Figure 6 Phases involved in Pattern Recognition

3.2 System overview

The system overflow begins with the data collection step, as shown in Figure 7. Photos and data were gathered from Bhutanese textile shops as well as random participants' wardrobes. The additional photos were also gathered from the internet. The collected data were augmented to increase the dataset's variation. Google Collab was used to train the machine learning algorithm Convolutional Neural Network on the dataset. The model was then saved and used to test its output on the webcam. The following section delves into the phases involved in recognizing Bhutanese textile patterns.

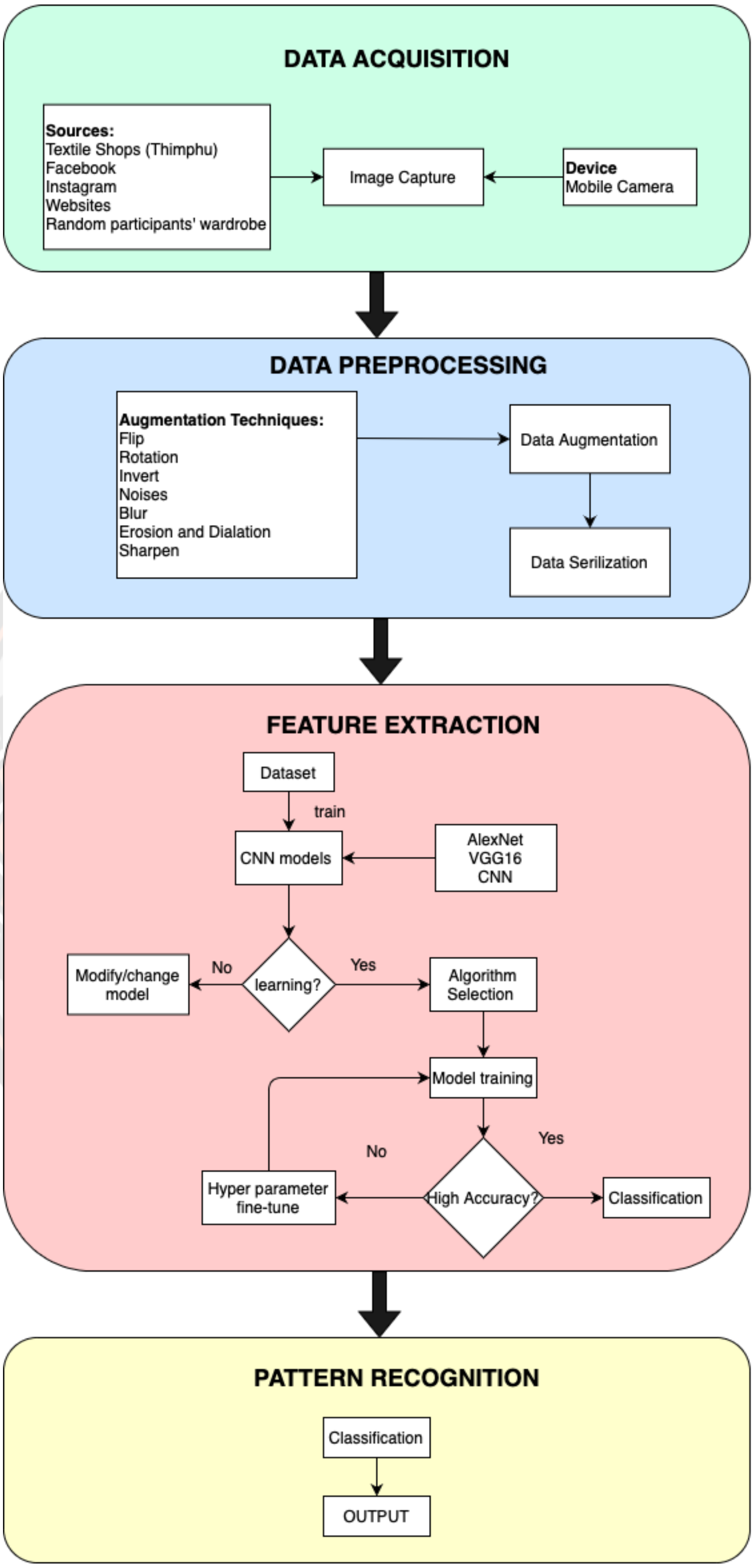


Figure 7 System Overflow

3.3 Data acquisition

The BHTP and BTT datasets were compiled from textile shops in Thimphu, Bhutan, as well as the selected participants' wardrobes. The images from textile shops were captured by walking into the textile shops. For at least one class of the pattern, ten random participants had contributed their textile photographs. Additional images were collected from Facebook pages and Instagram accounts such as CDK, Kelzang Handicraft, Bhutan Textiles, Chen-Ray Textiles, Tashi Yoedbar Handicraft, Norbooz Buray and Textiles, Kencho Couture, Dungsam Textile and Wangpestreetstyle. Most of the data (80%) are gathered through web crawling, Facebook pages, Instagram pages, and other websites. Mobile captures from Thimphu's textile stores have acquired less than 10% of the overall data. Less than 10% of the data is gathered from random participants' wardrobes.

The ten different patterns for BHTP (dataset 1) and seven different types of textiles for BTT (dataset 2) used in this study were chosen based on their availability and common use. The 10 most common and significant patterns were selected for the BHTP dataset as shown in Figure 8. 50 images were collected for each pattern class, totalling 500 images, which was then augmented to increase the dataset size.

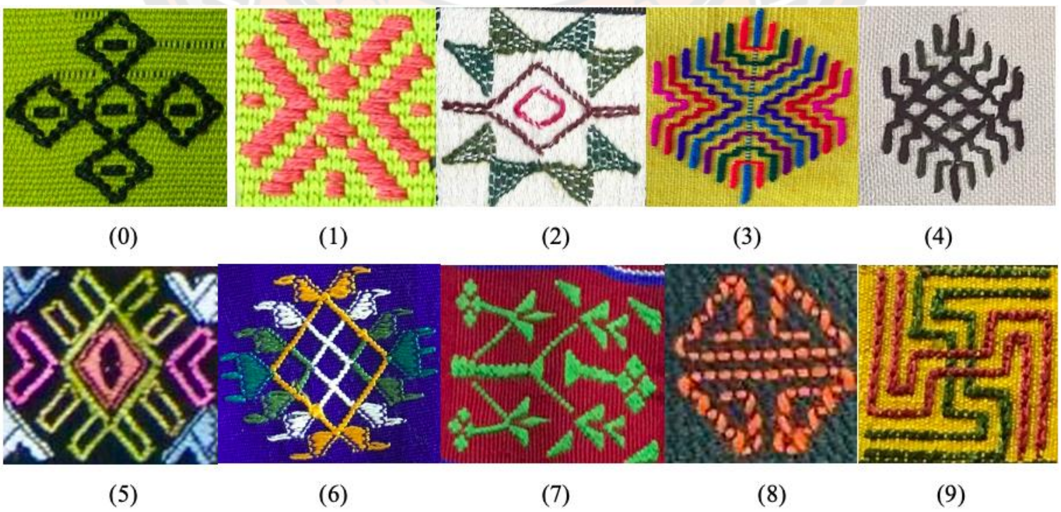


Figure 8 Ten selected patterns for the BHTP dataset

Dorji (0) or double thunderbolt is a ritual weapon in Hinduism, Buddhism, and Jainism. It represents the properties of indestructibility and irresistible force. It is thought to protect against all impediments, illnesses, and misfortunes. *Gemse (1)* is based on a peanut that absorbs the Earth's energy, resulting in continuous prosperity and wealth growth. Peanut represents a time of fruition and balance when everything is working together or will soon be working together at their perfect timing. *Karma (2)* is a symbol of a butterfly that represents resurrection, change, renewal, hope, endurance, and the courage to embrace transformation to improve one's life. It is interpreted as a symbol of spiritual transformation in Buddhism. Patterns such as *karsi chi (3)* and *mehub tima (4)* are varieties of *tsenden rema* or *trima.* It is the Bhutanese word for the shape of a cypress tree leaf, and it represents the cypress. Aside from all other attributes of a tree, the cypress represents immortality and hope, growth, and ambition. *Pagoda (5)* symbolizes the Buddhist architecture of temples found in Bhutan, Thailand, Nepal, India, Sri Lanka. The term "pagoda" is derived from the Sinhalese word "dagoba" which means "relic chamber" in Sri Lanka. The pagoda evolved from ancient Nepalese stupas, and its name reflects the stupa's function as a reliquary – a place to store holy relics. *Phyemali Tren (6)* is a bee web that represents fertility, wisdom, chastity, love, success, wealth, hard work, and altruism. *Shinglo (7)* represents the tree of life. In general, leaves represent fertility and growth, though different leaves represent different symbols. Buddhists associate leaves with the Bodhi Fig tree, where the Buddha attained enlightenment through meditation. This textile design symbol combines temporal and spiritual significance. *Thala (8)* pattern is a coin which is also known as the *Khorlo* or the wheel by some weavers. The pattern is associated with longevity and can be found on a wide range of fabrics, both old and modern. *Yunrung (9)* or *Swastika* is derived from the Sanskrit word svastika, which means "conducive to well-being." It is a symbol of good fortune and prosperity in Buddhism, Hinduism, and Jainism. Because of the lovely patterns and their significance, people have adapted them for use as a textile design.

Similarly, for the BTT dataset, the images were collected from the same sources as that of the BHTP dataset. However, the BTT dataset has 7 classes as depicted in Figure 9. Seven Bhutanese textile types that are commonly worn by Bhutanese were collected for this dataset. 100 images were collected for each class totalling 700 images in the dataset. The rest of the images were regenerated using augmentation techniques. Textile type names are as follows: *(0) Hor, (1) Jadrima, (2) Kishuthara, (3) Marthra, (4) Pangtse, (5) Serthra, (6) Shinglo.* Hor has horizontal line patterns with some simple patterns, but Jadrima simply contains horizontal patterns in a variety of colours. Kishuthara has a wide range of horizontal and complex patterns. Marthra has checked patterns that are both horizontal and vertical. In Marthra, the dominant colours are red, blue, and green. Pangtse has checked patterns with white or ash dominant colours, while Serthra has orange (sometimes red) and yellow checked patterns. Shinglo has complicated patterns known as Shinglo with horizontal patterns, as the name implies.

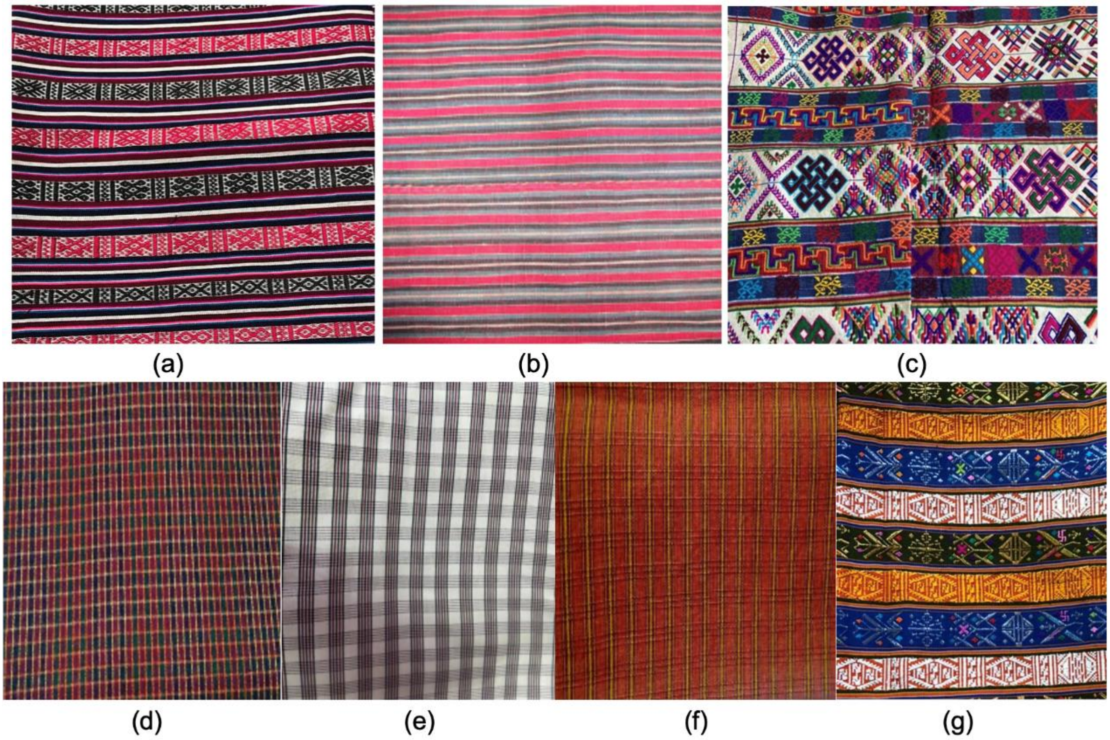


Figure 9 Sample images of 7 selected textile types for BTT

3.4 Data Pre-processing

Data pre-processing is a significant aspect of machine learning because it converts images into a machine-readable format. As per Obaid et al. (2019), missing values, noises, discrepancies, and redundancies are common characteristics of raw data, and their presence affects the output of the subsequence learning phase. As a result, a proper pre-processing phase is often used to restrict the effects of data anomalies (if any) on subsequent steps' output (quality and reliability). Most issues with training Deep Neural Network models are caused by a lack of sufficient training data or an uneven class balance within datasets. One way to get around this is to use data augmentation on the dataset (Mikołajczyk & Grochowski, 2018). As per Iqbal Hussain et al. (2020), data augmentation also solves the overfitting problem and improve the training effectiveness.

The BHTP dataset contains 30000 images for training and validation. 500 images were collected from Bhutanese Textile shops in Thimphu, random participants' wardrobes, Facebook, and Instagram pages. The rest of the were generated using image augmentation techniques. The images were read from the folder using the Python in Visual Studio Code (VS code) to perform data augmentation.

Data augmentation is required to increase the size of data needed to train a model. Deep learning models frequently need a large amount of training data, which is not always available. As a result, existing data is augmented to produce a more comprehensive model. Both positions, as well as colour image augmentation, are performed on the datasets. Each image augmentation methods are described as follows. Scaling is when the image is scaled to the specified size when scaling or resizing, for example, the width of the image can be doubled or reduced.

Image inversion is the process of reversing the colours on an image by subtracting the original image pixel value from the maximum pixel value (255). Saturation: The depth or intensity of colour contained in an image is referred to as saturation. Saturation is also known as chroma. The more saturated an image is, the more colourful and brilliant it will appear; the less saturated an image is, the more subdued or muted it will appear.

Greyscale is when the image pixel values are grey shades ranging from white to black. Sharpening an image brings out features or edges by increasing the contrast between bright and dark areas. An image is blurred when the edges or features in the image are smoothed off between bright and dark areas. Based on the behaviour of the pixels being selected and replaced, blurring can be of type median blur, average blur, and bilateral blur.

Image noise is the random variation of brightness or colour information in captured images. It is caused by external sources to degrade the image signal. The noises such as Gaussian, salt and pepper can be added manually to an image to alter the image features. Gaussian Noise, also known as Gaussian Distribution, is statistical noise having a normal probability density function. To generate this noise, a Random Gaussian function is applied to the Image function. It is also known as electronic noise since it occurs in amplifiers or detectors. Salt noise is created by adding bright (255-pixel value) pixels throughout an image. Pepper noise is created by adding random dark pixels (with a pixel value of 0) across an image. Salt and Pepper noise is created by adding both bright and dark pixels throughout an image. These noises are also known as impulse noises.

Morphological operations include dilation and erosion. Dilation adds pixels on the edges of objects in an image, whereas erosion removes pixels from the image. The top-hat and black-hat transforms are morphological operations used to extract small elements and details from images. The difference between the input image and its opening by some structuring element is defined as the top-hat transform. The difference between the closing and input images is defined as the black-hat transform. Padding in deep learning is the addition of extra pixels on image boundaries for more accurate image analysis. Flipping: The vertical flip will reverse the entire pixels column-by-column, while the horizontal flip will reverse the entire pixels row-by-row. Rotations: Images can be rotated in any direction, such as 45 degrees or 90 degrees clockwise or anti-clockwise. The original information on the image is preserved in the rotated image.

The augmentation techniques performed are shown in Figure 10; (a) Original image, (b) Invert-Gamma Correction, (c) Saturation, (d)Sharpen, (e) Multiply, (f) Erosion, (g) Dilation, (h) Padding, (i) Average blur, (j) Bilateral blur, (k) Median blur, (l) Horizontal flip, (m) Vertical flip, (n) Rotation (o) Salt noise, (p)Pepper noise, (q)Salt and Pepper noise, (r) Gaussian Noise, (s) Greyscale, (t) Top hat transform.

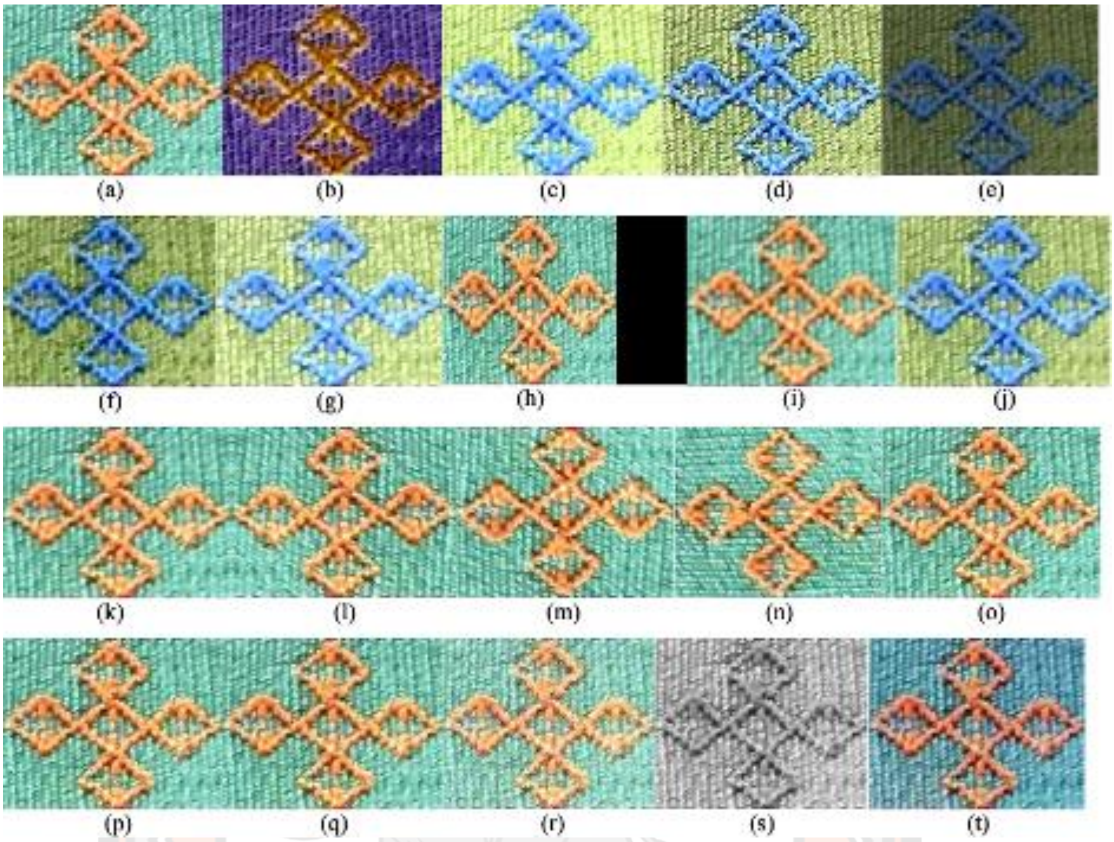


Figure 10 Data Augmentation on BHTP dataset

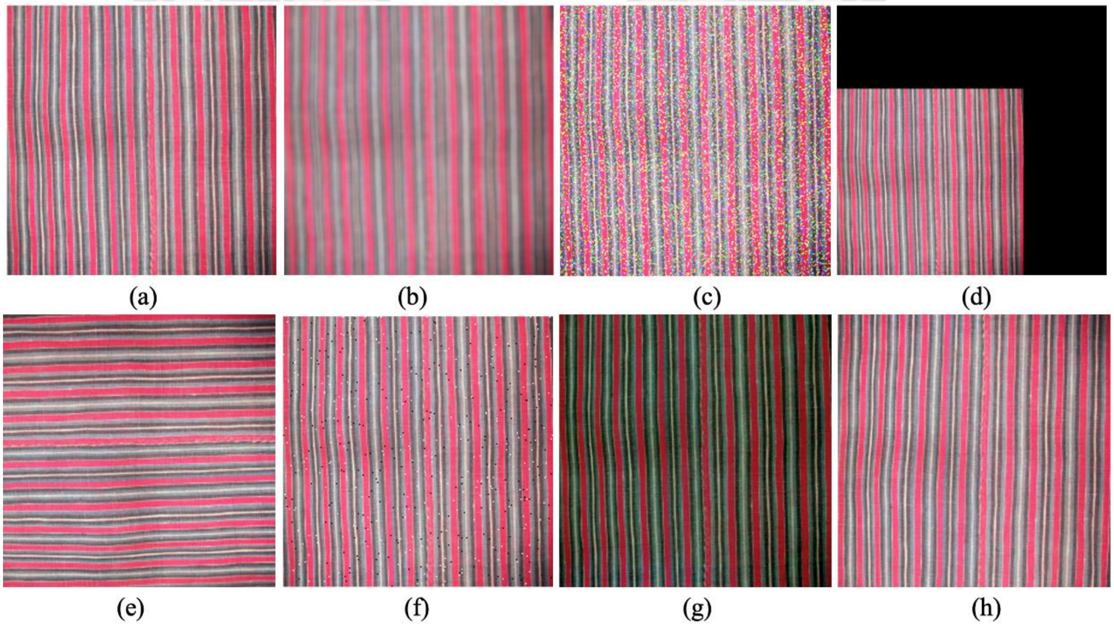


Figure 11 Data Augmentation on BTT dataset

As shown in Figure 11, similar data augmentation techniques were performed on the BTT dataset to increase the dataset size. BTT dataset size was increased from 700 to 8400 images.

The datasets were then serialized and uploaded on Google drive to train the model using Google Collab. The process of converting a data structure or object state into a format that can be saved and revived in the same or another computer environment is known as serialization. Python pickle module was used in VS Code to perform the serialization. These pickled data were uploaded to Google Drive and data was read for training the model using Google Collab. In the following segment, we will go over feature extraction and model preparation. Before training the model, the datasets were divided into 80% training and 20% testing sets as shown in Table 1. For the BHTP dataset, each pattern class had 2400 images for training and 600 for validation after the augmentation. BTT dataset has a total of 8400 images which is split into 6720 images for training and 1680 images for the validation set. BHTP dataset also has a testing set containing 550 images gathered from the same sources to test the selected model on the web. The images for both datasets were rescaled to 64x64 pixels and fed into the proposed training model.

Table 1 Bhutanese Textile datasets

Dataset	Class	Training	Validation	Total
BHTP	10	24000	6000	30,000
BTT	7	6720	1680	8,400

3.5 Feature extraction

3.5.1 Introduction

Feature extraction is a dimensionality reduction method that reduces a large collection of raw data into smaller groups for processing. Feature extraction refers to methods for selecting and merging variables into features to decrease the volume of data that needs to be processed while still accurately and completely representing the original data set. The primary objective of feature extraction is to extract the most important information from the original data and display it in a lower-dimensional space (Kumar & Bhatia, 2014). Features can be the colour, shape, or texture of the image. Global, block-based, and region-based features are the three types of feature representation methods. The authors described local and global features as two types of feature extraction. Geometry such as concave/convex, endpoints, branches, joints, and shapes are used to create local features. Topological characteristics, on the other hand, are determined by lines, corners, curves, connectivity, and the number of holes. Deep learning algorithms such as Convolutional Neural Network (CNN) and Visual Geometry Group (VGG) extract the features automatically from the image during the training. As per Liu (2018), convolution is an effective method of feature extraction in image processing, capable of reducing data dimension and generating a less redundant data collection, also known as a feature map. Each kernel acts as a feature identifier, removing out areas of the original image where the feature resides. It eventually generates a map whose altitude shows the distribution of these features.

3.5.2 Convolutional Neural Network

Deep Learning or Deep Neural Network means an Artificial Neural Network (ANN) with multilayers. Deep Learning imitates the human brain's processing of data and the building patterns for decision-making. Deep learning is often referred to as deep structured learning or hierarchical learning. It is a subcategory of machine learning that used an artificial neural network to perform the process of machine learning. The artificial neural networks are designed similarly as the human brain, with neuron nodes connected in a web-like pattern.

Convolutional Neural Network is one of the common deep neural networks with excellent performance in dealing with image classification, computer vision and natural language processing (NLP) (Albawi et al., 2017). In general, CNN comprises three different types of layers: input layer, hidden layer, and output layer. The input layer receives input signals and forwards them onto the hidden layers between the input and output layers. Signals are processed and activated under certain conditions and passed on through the hidden layers. Finally, the output layer generalizes output signals. A CNN's hidden layers are typically made up of convolutional layers that convolve with multiplication or other dot product.

3.5.3 Feed-Forward Artificial Neural Network (ANN)

Feed-Forward Artificial Neural Network is a biologically inspired classification algorithm that is also known as a multi-layer perceptron (MLP). It is the first and simplest neural network invented, with no cycles created by connections between units or neurons. The data travels only in a forward direction in the network, from the input to the hidden layer, and then to the output nodes. The network is called a feed-forward network because it does not have feedback. If there is more than one hidden layer in the network, it is called a deep neural network.

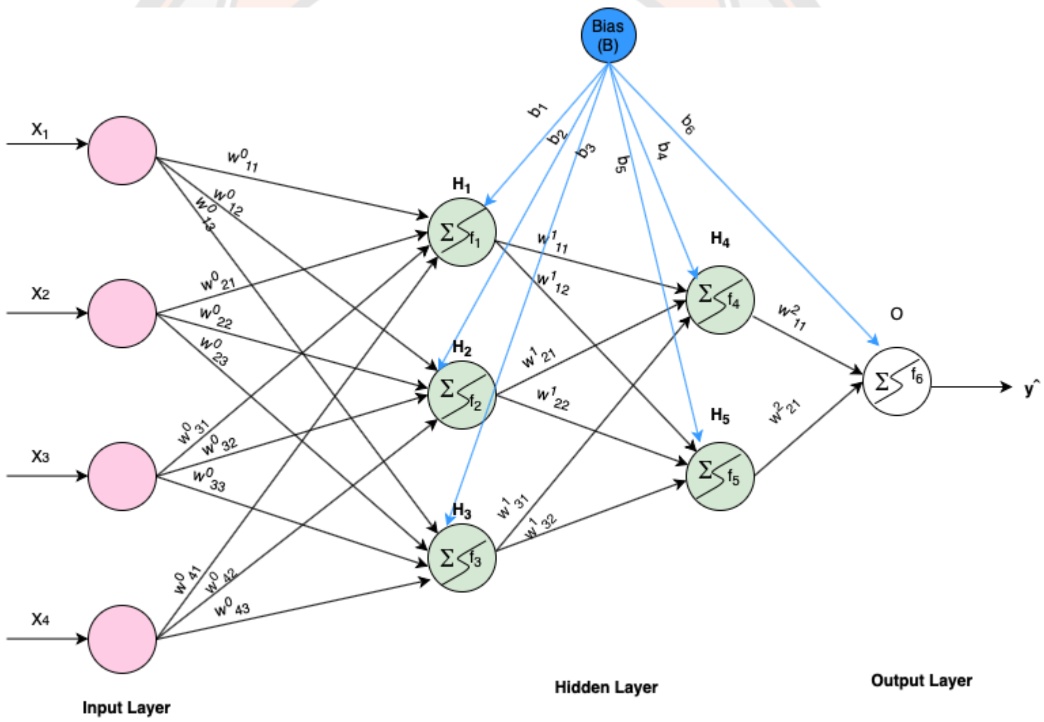


Figure 12 Feedforward Neural Network

Figure 12 shows a feed-forward neural network with two hidden layers. The weights are calculated in each neuron by the sum of element-wise products and are passed to the activation function, which is a threshold function. The predicted value from the input data is the final output. The features of input x_1, x_2, x_3 and x_4 with their weights from their inputs are fed into the first hidden layer as inputs.

Weights are randomly initialised by using weight initialization techniques such as uniform distribution, Xavier/Gorat distribution and He init. Each neuron performs the weight calculation and activation functions. The weight is calculated by the sum of the dot product of the input feature and its weight. The general formula for weight calculation is as follow:

$$H_n = \sum_{i=1}^{n} (w_i . x_i) + bias$$

The weights of neurons H_1, H_2, and H_3 are calculated as follows:

$$H_1 = w_{11}^0 . x_1 + w_{21}^0 . x_2 + w_{31}^0 . x_3 + w_{41}^0 . x_4 + b_1$$
$$H_2 = w_{12}^0 . x_1 + w_{22}^0 . x_2 + w_{32}^0 . x_3 + w_{42}^0 . x_4 + b_2$$
$$H_3 = w_{13}^0 . x_1 + w_{23}^0 . x_2 + w_{33}^0 . x_3 + w_{43}^0 . x_4 + b_3$$

To activate the neuron, the calculated weight is then passed through an activation function. The activation function selection depends on the individual researcher. The sigmoid activation function is defined by the following equation:

$$f(x) = \frac{1}{1 + e^{-x}}$$

As shown below, the activation sigmoid is applied to neurons H_1, H_2 and H_3:

$$f_1 = Activationfunction(H_1)$$
$$f_1 = \frac{1}{1 + e^{-H_1}}$$
$$f_2 = \frac{1}{1 + e^{-H_2}}$$
$$f_3 = \frac{1}{1 + e^{-H_3}}$$

The neurons f_1, f_2 and f_3 are fed as inputs to the next hidden layer nodes H_4 and H_5, along with their weights. The activation function is then applied to H_4 and H_5 to produce f_4 and f_5, as shown below.

$$H_4 = w_{11}^1 . f_1 + w_{21}^1 . f_2 + w_{31}^1 . f_3 + b_4$$
$$H_5 = w_{12}^1 . f_1 + w_{22}^1 . f_2 + w_{32}^1 . f_3 + b_5$$
$$f_4 = \frac{1}{1 + e^{-H_4}}$$
$$f_5 = \frac{1}{1 + e^{-H_5}}$$

Similarly, the final neuron in the output layer is activated and its weight is calculated and the f_6 is the predicted output.

$$O = w_{11}^2 . f_4 + w_{22}^2 . f_2 + w_{32}^1 . f_5 + b_6$$
$$f_6 = \frac{1}{1 + e^{-0}}$$

1) Loss Functions

The loss function compares the prediction and the true value to assess the algorithm's performance on the dataset. Depending on whether the model's learning is poor or good, the difference between predicted and actual value will be greater or lesser. Mean Absolute Error (MAE), also known as L1 loss, Hinge, Huber, and Kullback-Leibler are examples of loss functions. Mean Squared Error (MSE) and Cross-Entropy are the most used loss functions in image classification. MSE is also referred to as L2 loss.

For the illustration, the error is measured using Mean Square Error (MSE), but the loss function is chosen by the individual researcher. The general formula for classification error is shown below.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y})^2$$

Where ‘n' represents the number of data points, y_i represents the actual observed values, and$\hat{y}$ represents the predicted value. On the first forward pass, the error will most likely be greater. This error can, however, be reduced by employing the backpropagation method, which iteratively updates the weights and trains the model for improved performance.

3.5.4 Back Propagation Algorithm

In backpropagation, we go back and adjust the parameters (weights and biases) to optimize the cost function or to improve the performance of the network. It is used to train the neural network through a method called chain rule.

1) Optimizers

An optimizer is an algorithm that computes the function's minimum value. It reduces loss by fine-tuning neural network parameters such as weights, biases, and learning rates during backpropagation. Optimizers include Gradient Descent, Stochastic Gradient Descent (SGD), Adagrad, RMSprop, Adam, Adadelta, Adamax, Nadam, and Ftrl. ADAM Optimizer was used in all the models to change the attributes of the model such as weights and learning rate (Bera & Shrivastava, 2020). Gradient Descent has been dubbed "the mother of machine learning" by many researchers. It is the main workhorse of learning features. Figure 12 and Figure 13 depict the gradient descent loss calculation and weights update, respectively.

The general equation for backpropagation is as shown below, where $w_{(t+1)}$ is a new weight, $w_{(t)}$ is the old weight, η is the learning rate that determines the rate of convergence to the global minima, and $\frac{\partial L}{\partial w_t}$ is the derivative of loss w(t) concerning the old weight that determines the direction of descent to the global minima.

$$w_{(t+1)} = w_t - \eta \frac{\partial L}{\partial w_t}$$

To update the weight of node w^3_{11} , the illustrations are as below,

$$w^2_{11(new)} = w^2_{11} - \eta \frac{\partial L}{\partial w^3_{11}}$$

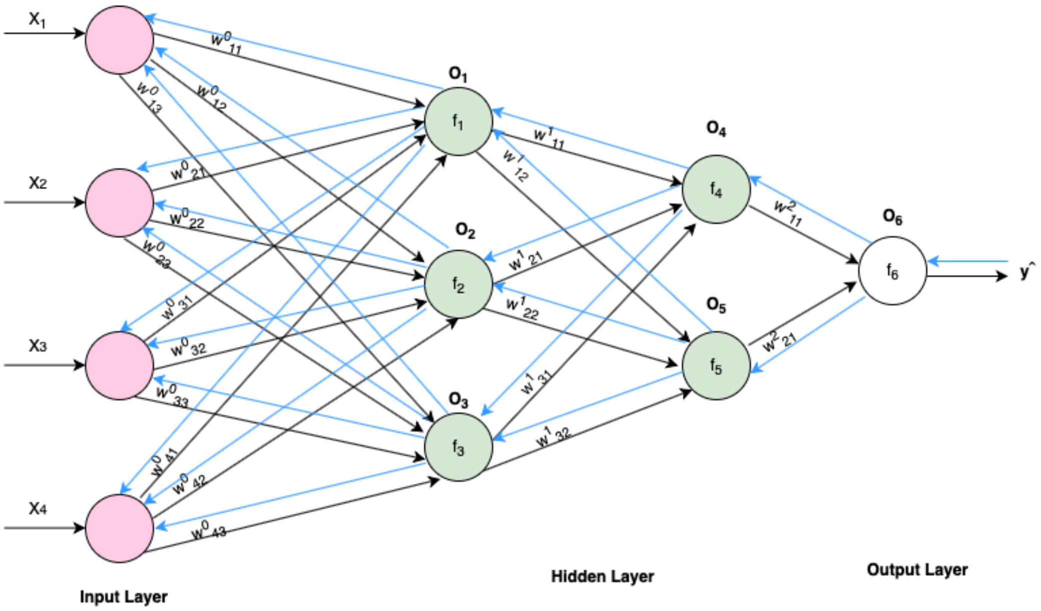


Figure 13 Back Propagation

The values of w_{41} and η are known in the feedforward training. Now, using the chain rule, $\frac{\partial L}{\partial w^{3}_{11}}$ is computed as below:

$$\frac{\partial L}{\partial w^{2}_{11}} = \frac{\partial L}{\partial O_6} \cdot \frac{\partial O_6}{\partial w^{2}_{11}}$$

Similarly, the weights for w^{1}_{11} is updated as below:

$$w^{1}_{11(new)} = w^{1}_{11} - \eta \frac{\partial L}{\partial w^{1}_{11}}$$

$$\frac{\partial L}{\partial w^{1}_{11}} = \frac{\partial L}{\partial O_6} \cdot \frac{\partial O_4}{\partial O_6} \cdot \frac{\partial O_6}{\partial w^{1}_{11}}$$

As a result, the other weights are updated in the same way. The second iteration begins once all the weights have been updated. The number of weight updates is the same as the number of training epochs. The loss is reduced with each epoch. However, the loss could remain the same or increase. The vanishing gradient (VG) problem makes the network difficult to learn and error reduction remains nearly the same or slightly different, whereas the exploding gradient (EG) problem causes a large error by updating large weights. The VG problem can be solved by using activation functions other than sigmoid or tanh, and the EG problem can be solved by properly initializing the weights during training.

3.4.5 Components of CNN

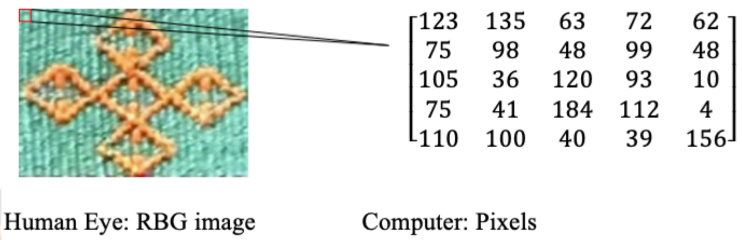


Figure 14 Different perceptions of images by humans and computers

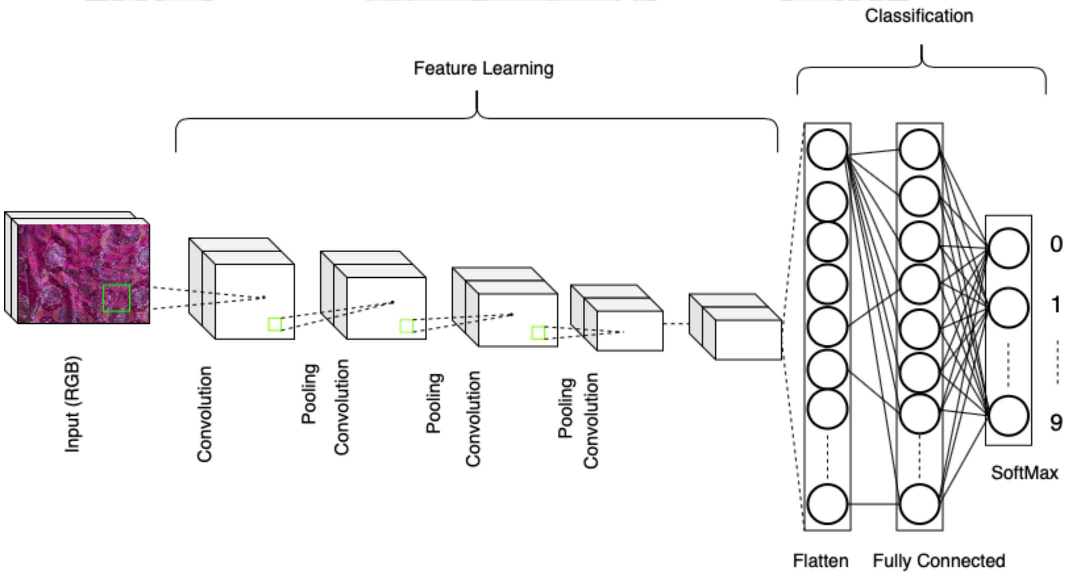


Figure 15 Components of CNN

Computers see images in pixels, unlike human eyes. Colour shades ranging from 0–255-pixel values are used to divide the images. A grayscale picture has 256 different colours, with 0 being black and 255 being white. The RGB picture, on the other hand, has three channels, each with a colour range of 0-255 pixels. Human eyes perceive the images as RGB, while computers perceive them as pixels as shown in Figure 14.

These pixels are then flattened into a vector and fed into the CNN layer as an input. A typical CNN has layers such as a convolutional layer, pooling layer, flatten layer, fully connected layer and loss layer for the classification as shown in Figure 15.

1) Convolutional Layer

CNN's first part is the convolution layer. It uses a variety of filters to extract features (lines, edges, corners, and curves) from the input image (Yamashita et al., 2018). The image matrix is multiplied or convolved with the filter (kernel) to produce a feature map as shown in Figure 16.

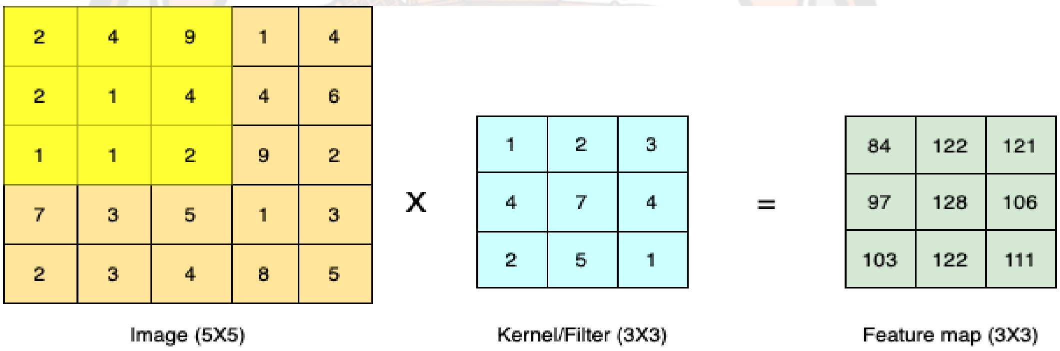


Figure 16 Convolution operation

The yellow section is an input image of size 5x5, which convolves with the filter size 3x3, represented in blue. Following the convolution operations, the final feature map in 3x3 size is produced, which is represented in green colour. The kernel moves from the top-left of the input image to the bottom-right, convolving with each pixel. The dimensions of the image are reduced during the convolution operation, and the filters do not exactly fit the image. Padding could be used to solve these issues. The same dimensions are maintained by using zero paddings around the images. the general formula for convolution operation is

$$g[x,y] = (f * k)(x,y) = \sum_i \sum_i f[x-i, y-i] * k[]i,j$$

Where f and k represent the input image and filter, respectively. X and y represent the rows and columns, respectively. The feature map is the output of convolution and is denoted by the letter g. The indexes I and j are used to convolve the image and filter. In image content analysis, filters are extremely essential. While convolving an image, various filters extract various features such as edge detection, sharpening, and blurring.

2) Activation Functions

Activation functions activate neurons based on relevant information from input data to assess the performance of the neural network. If the neuron does not have features that can predict desired outputs, it will not be activated. The sum of inputs and weights are determined at each neuron, and then the activation function is applied to obtain the output, which is then supplied to the next layer. Ten activation functions squash the output to a finite value: binary phase function, linear, sigmoid, tanh, ReLU, leaky ReLU, parametrized ReLU, exponential linear unit, swish, and SoftMax. Sigmoid is particularly useful in models where the probability must be predicted as an output. Since the chance of something only exists between 0 and 1, a sigmoid is the best option. The benefit of using tanh is that the negative inputs will be mapped strongly negative, and zero inputs will be mapped near zero. Rectified Linear Unit (ReLU) is efficient in increasing the expression ability of neural networks and it is the most widely used activation function in deep learning Wang et al. (2020).

The vanishing gradient problem affects the sigmoid and Tanh functions, in which neurons either do not change their weights or do so very slowly. As shown in Figure 17, the sigmoid (a) and Tanh (b) take real numbers and squash them between 0 and 1 and -1 to 1 respectively. The sigmoid and tanh functions are defined by the formulas below:

$$\text{Sigmoid: } f(x) = \frac{1}{1+e^{-x}}$$

$$\text{Tanh: } f(x) = \frac{2}{1+e^{-2x}} - 1$$

The ReLU(c) takes real-valued numbers and squashes them between 0 and the maximum number. It is defined by,

$$f(x) = max(0, x)$$

$$f(x) = \begin{cases} 1, if\ x > 0 \\ 0, if\ x < 0 \end{cases}$$

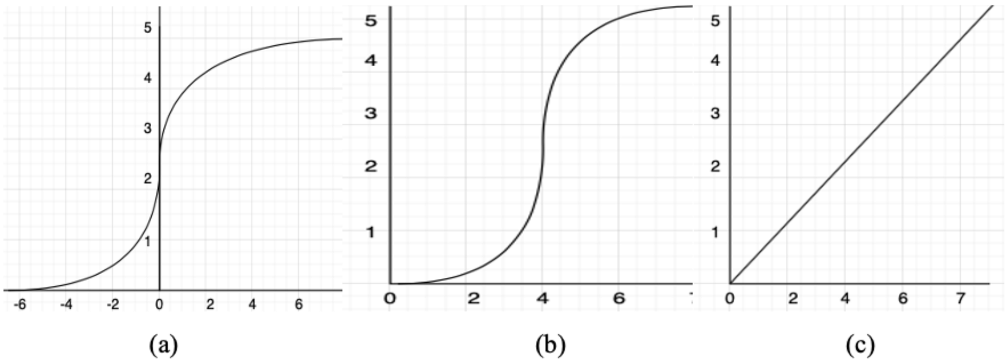


Figure 17 Activation Functions

3.4.6 Pooling Layer

The pooling layer is another building block of CNN architecture that down-samples the input images' spatial dimensions and works on each function map independently. It reduces the number of parameters and makes processing more computationally efficient (Yu et al., 2014). By preserving important information while discarding others, pooling aims to combine the most prominent features of a picture. Pooling can be average pooling, maximum pooling, and maximum pooling. Figure 18 illustrates an example of max pooling which chooses the single maximum value from the neighbourhood pixels.

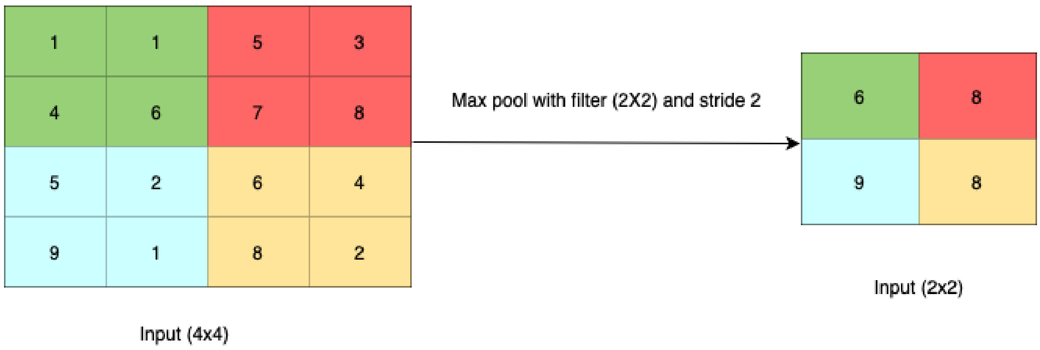


Figure 18 Max Pooling

To perform max-pooling on this input channel, the input size is 4x4 and the filter size is 2x2 with a stride of 2. The formula for max pooling is,

$$f(x) = max_i . x_i.$$

The first region is orange, with a maximum value of 6, so 6 is saved in the output channel. After that, we slide over 2 pixels and see that the red region's maximum value is 8. As a result, we can store 8 on the output channel. It will move back to the left and shift down by 2 pixels now that it has reached the edge. The maximum value in the cyan region is 9 in this case, and it is thus saved in the production. Finally, it is shifted 2 pixels to the right, and the full value of 8 is saved in the output channel. Thus, the max-pooling has been completed for this sample input with a size of 4x4 and a filter size of 2x2. Similarly, the average max-pooling outputs the average of all the neighbouring pixels in each region. The formula for average pooling is,

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

where n is the total number of pixels on the local pooling region. Global pooling, on the other hand, uses only one value from the function map. It dramatically reduces the size of the object. As a result, the N_h x N_w Xn_c dimension of the function map is reduced to $1x1Xn_c$ dimension. Global max pooling or global averaging pooling are two options for global pooling.

1) Batch Normalization

Normalization is a pre-processing technique used to standardize data in preparation for training. It transforms the data to fit on the same scale while preserving value range contrasts. Batch normalization is a method of accelerating training that is important in obtaining state-of-the-art results on benchmark problems in many studies. Ioffe and Szegedy (2015) proposed the normalization layer which is used to normalize the previous layers' output and speed up the training process, stabilize the network and reduce the covariance. Each part of a layer in a neural network is normalized to zero mean and unit variance using batch normalization, which is based on statistics from a mini-batch. Batch normalization has several advantages in the deep neural network training process, including avoiding the gradient exploding and disappearing problem, speeding up the training process, reducing the model's initialization effect, and making the output distribution more evenly distributed (Chen et al., 2020). Learning becomes more efficient when batch normalization is used, and it can also be used as regularization to avoid model overfitting. The expressions for mean, standard deviation and normalization are as shown below respectively,

$$\mu = \frac{1}{n}\sum_i z^{(i)}$$

$$\sigma^2 = \frac{1}{n}\sum_i (z_i - \mu)^2$$

$$z^i_{norm} = \frac{(z^i - \mu)}{\sqrt{(\sigma^2 + \varepsilon)}}$$

μ represents mean andσ^2 represents standard deviation. z_i is the hidden unit value of the hidden layer in the network. ε is included to ensure numerical consistency ifσ^2 becomes zero during normalization. Any part of z has a 0 mean and standard unit variance after normalization. Hidden units, on the other hand, do not want to have a 0 mean and a single variance; instead, hidden units prefer to have multiple distributions. The different distribution is expressed as,

$$\tilde{z}^i = \gamma z^i_{norm} + \beta$$

γ and β are the batch normalization learnable parameters that allow hidden units to choose different mean and variance values.

2) Fully connected layer

Every neuron in one layer is linked to every neuron in another layer in fully connected layers. It works in the same way as a conventional multi-layer perceptron neural network in theory (MLP) (Albawi et al., 2017). To identify the images, the flattened matrix passes through a completely connected layer. As shown in Figure 19.a, each node in a fully connected layer is directly connected to any node in the previous and subsequent layers. A fully connected layer's main disadvantage is that it contains many parameters that require complex computation in training examples. As a result, we attempt to reduce the number of nodes and connections by applying the dropout strategy.

3.4.7 Dropout Layer

Dropout is a regularization technique that drops units or neurons from a neural network to combat overfitting (Wu & Gu, 2015). The overfitting problem occurs when a model performs well on training data but poorly on test data (unseen data), resulting in a higher generalization error. Dropout is the most commonly used regularization technique for deep neural networks, and many methods have been suggested, including weight decay, data augmentation, and batch normalization (Cai et al., 2019). It helps in reducing interdependent learning among neurons in neural networks.

During training, some neurons are temporarily dropped or ignored from the network in the dropout layer. During backpropagation, the neurons that were dropped during forwarding training would be overlooked. However, these neurons may or may not be dropped out of the neural network in the next iteration of forwarding training. Dropouts will result in decreased parameters, a fast trained network, and there will be a regularization effect since all the data points are not learned in a single epoch to avoid overfitting. Figure 19 illustrates the dropouts in the feed-forward neural network. Figure 19.a shows the fully connected feedforward neural network and Figure 19.b shows the implementation of dropouts where the black neurons are the dropout neurons. Although various dropout rates are used, 0.5% to 0.8% is a good range. Dropouts should not be used after the convolution layers, but rather after the deep layers of the network.

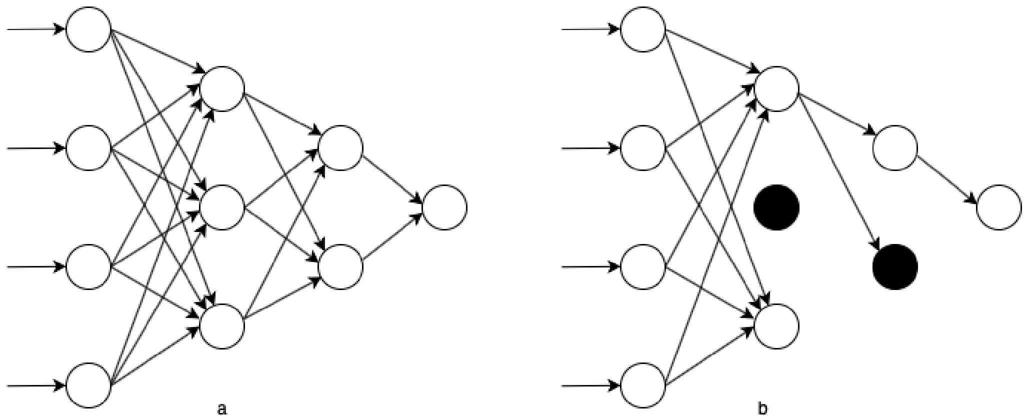


Figure 19 Dropout in neural network

3.4.8 AlexNet

AlexNet Convolutional Neural Network has 8 learning layers, 5 convolutional layers, and 3 fully connected layers (Krizhevsky et al., 2012). The network's final fully linked layer connects to 1000 classes, and the rest of the network is a feature extractor. For each image, AlexNet can generate a 4096-dimensional feature vector, which contains the activations of the hidden layer just before the output layer. Against all classic machine learning and computer vision approaches, AlexNet obtained state-of-the-art recognition accuracy (Alom et al., 2018). It was a significant moment in the history of machine learning and computer vision for visual recognition and classification tasks, and it marked the beginning of a surge in interest in deep learning. The architecture for AlexNet is as shown in Figure 20.

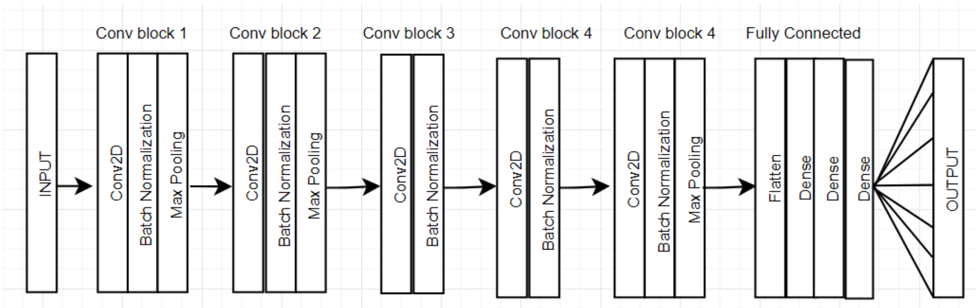


Figure 20 Alex Net Architecture

3.4.9 Visual Geometry Group

The VGG is a state-of-the-art network for visual identification developed by the Visual Geometry Group at Oxford University (VGG). It performs well on the ImageNet dataset as well as other images not included in ImageNet. VGG was the ILSVRC2014 image classification competition's first runner-up (Simonyan & Zisserman, 2014). It is one of the most used image classification and recognition architectures.

The performance of six different VGG networks by stacking layers. The RGB images of 224 x 224 pixels were fed into the VGG network. The network consists of convolution layers with 3x3 stride 1 filters and the padding, then the max-pooling layers with 2x2 filters of stride 2. The convolution layers have twice as many filters as the previous layer, such as 64, 128, 256, and 512. Throughout the architecture, these layers are arranged in the same way. Nonetheless, the number of layers in different VGG network varies.

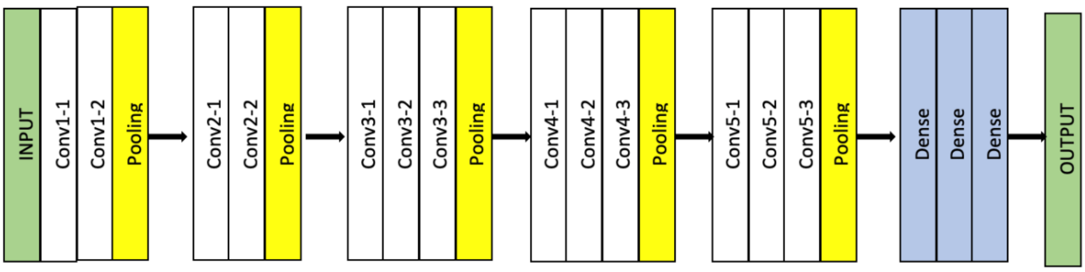


Figure 21 VGG16 Architecture

Figure 21 illustrates the basic architecture of VGG16 with 13 convolutional layers and 3 fully connected layers.

3.4.10 K-Nearest Neighbor

K-nearest neighbour (KNN) is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data (Peterson, 2009). The k-nearest neighbour algorithm (Bremner et al., 2005; Cover & Hart, 1967) is a method for classifying objects based on the feature space's closest training instances. KNN is used for regression, classification, and the imputation of missing values. It is a case-based learning method, which keeps all the training data for classification. This algorithm's training phase consists solely of storing the training images' feature vectors and labels.

The KNN algorithm assumes that similar classes are close together and it searches for similarity points between new data points and the classes. The shortest distance class is assigned to the new data point data points to classify the similarity between the classes. Euclidean distance is used to calculate the shortest distance between the new data point and neighbouring data points. It classifies the new data points based on how the neighbour data points are classified. A parameter 'k' refers to the number of nearest neighbours to include in the majority voting process. A data point is classified by the class which has a maximum number of neighbours from its 'k' nearest neighbours.

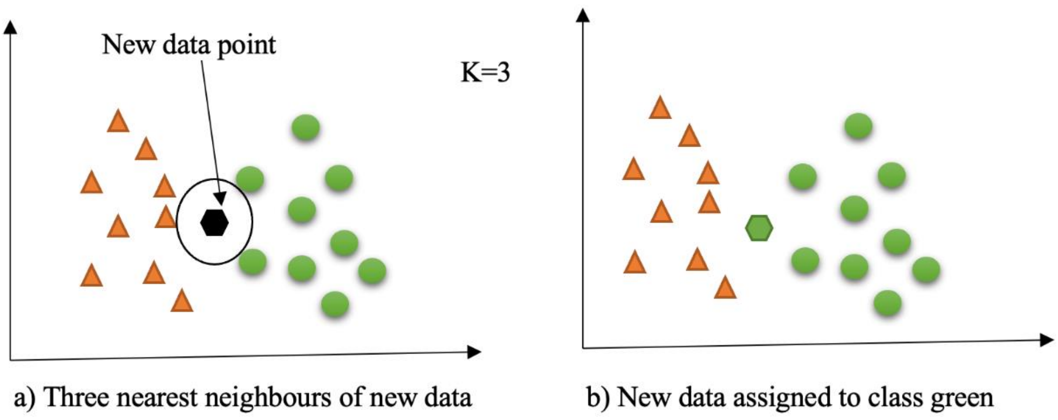


Figure 22 KNN data points

Figure 22 illustrates how the KNN method is used to assign new data points. The new data point (black) is to be classified as either of these two classes (orange and green). The nearest neighbour distance will be calculated from the new data point to each neighbouring data point. The shortest distance is calculated using Euclidean distance as shown in the formula below.

$$\boldsymbol{Euclidean\ distance} = \sqrt{(\boldsymbol{x}-\boldsymbol{a})^2 + (\boldsymbol{y}-\boldsymbol{b})^2}$$

Where, *(a,b)* is the coordinates of the new data point and *(x,y)* is the coordinates of the neighbouring data points. *K* in KNN is a parameter that refers to the number of nearest neighbours to include in the majority voting process. The *k* value is 3 in Figure 22, and the three shortest neighbours are shown by a circle. Two of the three nearest neighbours are in the class green; thus, the new data point is classified as well. KNN is simple to implement however, it performs slower with the dataset containing higher categories.

3.4.11 Support Vector Machine

Support Vector Machine (SVM) is a machine learning model used in classification and regression problems. It is a supervised model which works only with the labelled data. The algorithm generates a line or hyperplane that divides the data into categories. The hyperplane is a point in one-dimensional space, a line in two-dimensional space, and a plane surface in three-dimensional space that splits the data points into categories. SVM performs classification tasks on linearly separable and non-linearly separable data.

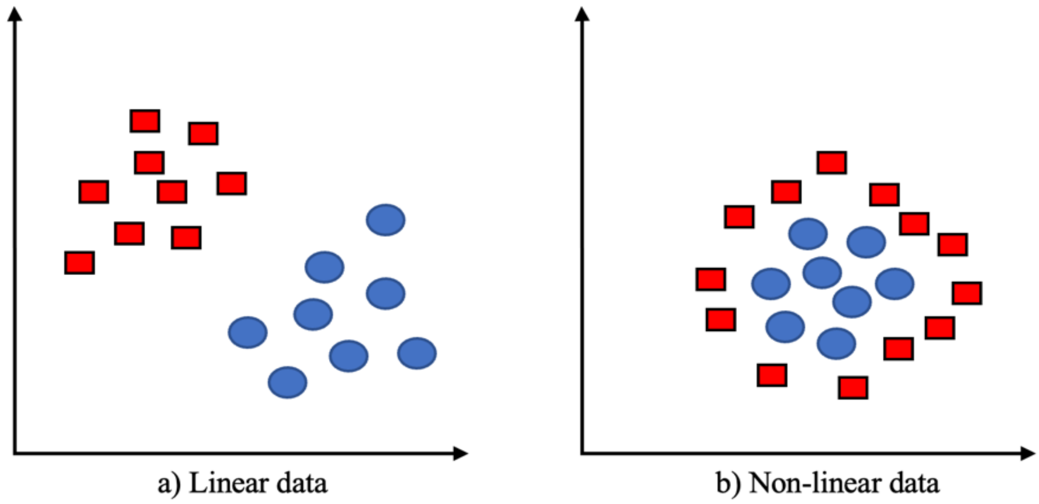


Figure 23 linearly separable and non-linearly separable data

Figure 23 shows linearly separable and non-linearly separable data. Linearly separable data can be separated by finding a line, but we cannot find a line to separate nonlinearly separable data. The linear SVM uses hyperplane, marginal distance, and support vector to linearly separate data. Hyperplanes are thresholds or decision boundaries that are used to classify the data points. The data point is better classified by the hyperplane with the largest marginal distance. SVM chooses the best hyperplane that maximizes the margin. SVM is sometimes called a Maximum Margin Classifier.

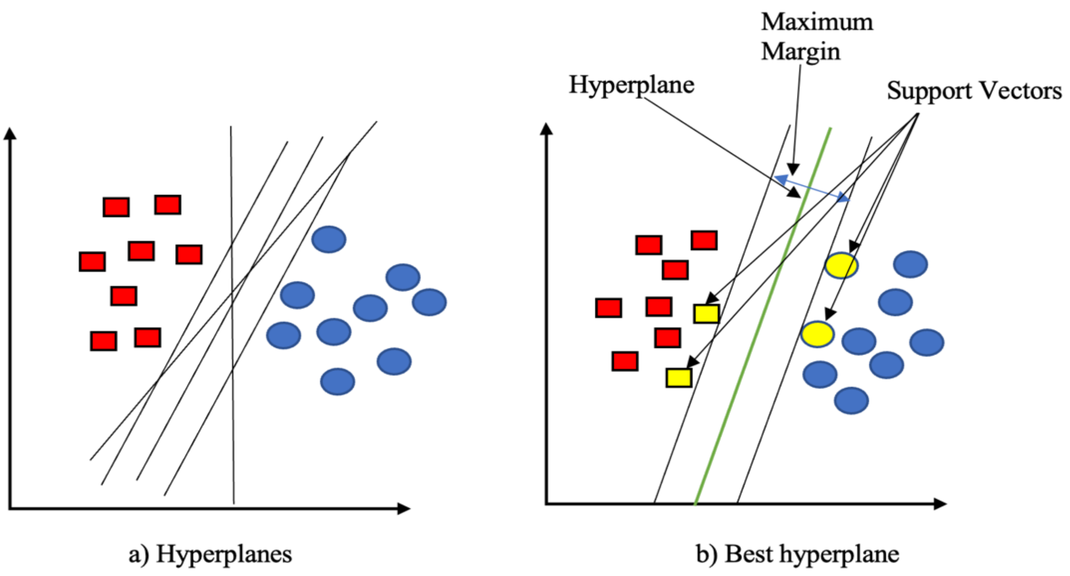


Figure 24 SVM data points: Hyperplanes and best hyperplane

The data points (yellow box and circles) are called support vectors as shown in Figure 24 (b). Support vectors are most difficult to classify, that lie closer to the hyperplane and influence the position and orientation of the hyperplane. To convert non-linearity data to linearity, kernel functions such as polynomial kernel, radial basis function (RBF), and sigmoid kernel are employed to translate input vectors into higher-dimensional space, as shown in Figure 25.

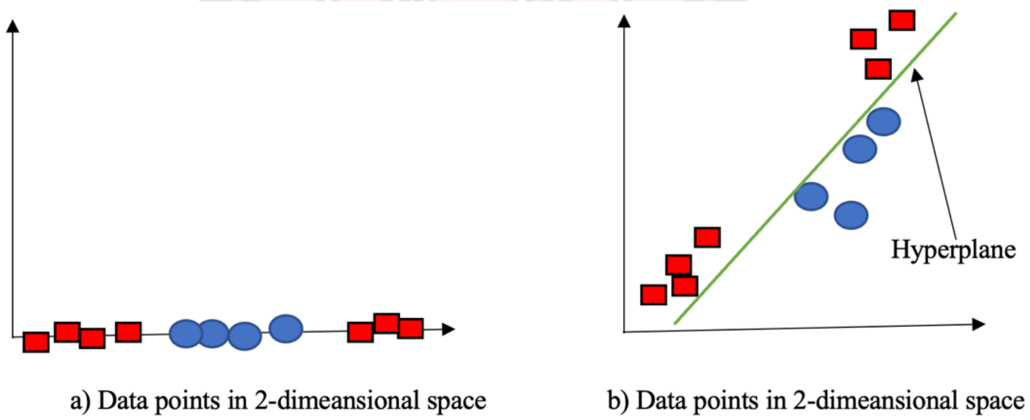


Figure 25 Data points in 1D and 2D spaces

SVM performs well on smaller datasets as it takes a longer time to process. It can be used in a variety of applications, including bioinformatics, face identification, text, and hypertext categorization, handwriting recognition, medical image classification, and many others.

3.6 Confusion Matrix

A confusion matrix (or Error Matrix) is a table that is frequently used to describe a classification model's performance on a set of test data for which the true values are known. As shown in Figure 26, the confusion matrix has parameters such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) to calculate the accuracy, precision, recall and F1-score of the model.

Predicted Values

Actual Values		Positive	Negative
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 26 Confusion Matrix

True Positive (TP) is when the actual value was positive, and the model predicted a positive value. True Negative (TN) is when the actual value was negative, and the model predicted a negative value. False Positive (FP) is also known as type 1 error. The projected value was incorrectly predicted. Although the actual number was negative, the model projected that it would be positive. False Negative (FN) is also known as type 2 error. The projected value was incorrectly predicted. Although the actual number was positive, the model predicted that it would be negative.

The simplest intuitive performance metric is accuracy, which is just the ratio of properly predicted observations to all observations. Precision is the number of positive class predictions that belong to the positive class. A recall is the number of positive class predictions made from all positive examples in the dataset. F-Measure generates a single score that accounts for both precisions and recalls concerns in a single number. The following equations show the formula for accuracy, precision, recall and F1-score,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FP}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 * precision * recall}{precision + recall}$$

Where TP is True Positive, TN is True Negative, and FP is False Positive.

CHAPTER IV

RESULTS

4.1 Introduction

The results of the proposed study are discussed in detail in this chapter. The main goal of the study was to create a machine learning model for the recognition and classification of Bhutanese handwoven textile patterns and textile types. In addition, the study focused on the collection and preparation of the first Bhutanese textiles and patterns image datasets. As discussed in Chapters II and III, the datasets are tested using various textile classification models.

In this study, two different textile datasets were created namely the Bhutanese Handwoven Textile Pattern (BHTP) dataset and the Bhutanese Textile Types (BTT) dataset. (BHTP) dataset focused on patterns that are comprised of 30,000 images from 10 classes. The Bhutanese Textile Types (BTT) dataset focused on types of textiles that consist of 8400 images for 7 types of textiles. Various machine learning models were trained on the datasets and an appropriate model each was selected for classifying Bhutanese handwoven textile patterns and classifying types of Bhutanese textiles.

The next sections present a more in-depth description of the findings. The recognition and classification of Bhutanese handwoven textile patterns are presented first, followed by the recognition and classification of Bhutanese textile types.

4.2 Bhutanese Handwoven Textile Pattern Recognition and Classification

The PatternNet was designed considering the past experiments gathered from literature reviews. The serialized RGB images were read from Google Drive and performed augmentation once more before being fed as inputs to the proposed model using the Keras class called *ImageDataGenerator*. The augmentations include rotation, height and width change, shear and zoom, and horizontal flip. Model parameters were fine-tuned until they achieved the desired accuracy. The model with the highest accuracy was saved and used to test unseen images or to analyse the model in real-time. Figure 27 illustrates the proposed CNN architecture (PatternNet) for the Bhutanese hand-woven textile pattern recognition model.

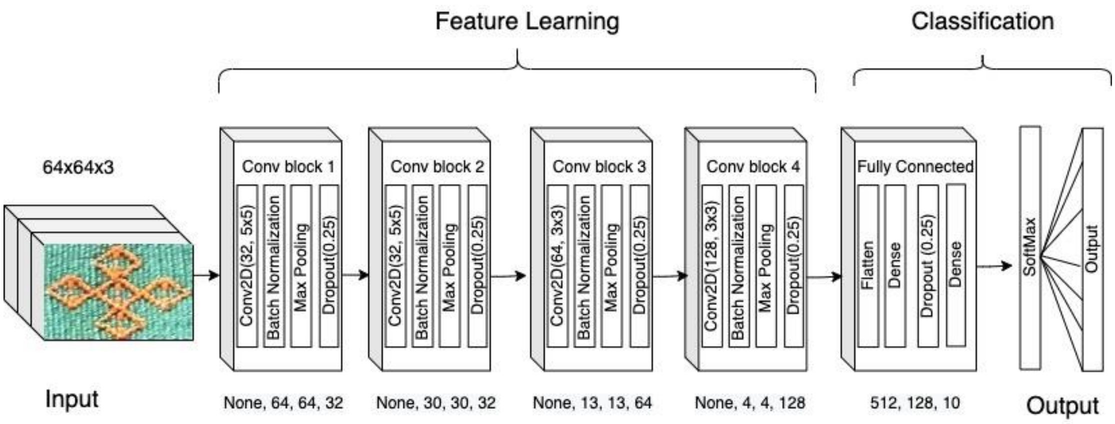


Figure 27 CNN Architecture of PatternNet Model

In layer 1, a convolution layer was stacked together using 32 filters of size 5x5 with a stride and the same padding followed by batch normalization, a max-pooling, and a dropout. The second, third and fourth layers (convolution blocks) have the same layer structure as the first, however, several filters and strides vary depending on the valid padding. To speed up training convergence and mitigate overfitting problems, the batch normalization and dropout layers were used. In layer 5, a flattening layer was stacked with two dense layers and a dropout layer with a rate of 0.25. The model was trained with the ReLU activation function, and the 10 patterns were classified using the SoftMax classifier.

Table 2 shows the configuration of the CNN model for PatternNet. There were 24000 images for training and 6000 for validation. Using Google Collaboratory, the PatternNet model was trained for 500 epochs with a batch size of 32 using Keras and TensorFlow. The best performance of the model had accuracies of 99.75% and 99.25% for training and validation respectively.

Table 2 Configuration of PatternNet Model for BHTP dataset

Type of Layer	Number of Filters	Filter size	Output
Convolution	32	5x5	64x64
Batch Normalization	-	-	-
Max Pooling	-	2x2	32x32
Dropout	-	-	-
Convolution	32	3x3	30x30
Batch Normalization	-	-	-
Max Pooling	-	2x2	15x15
Dropout	-	-	-
Convolution	64	3x3	13x13
Batch Normalization	-	-	-
Max Pooling	-	2x2	6x6
Dropout	-	-	-
Convolution	128	3x3	4x4
Batch Normalization	-	-	-
Max Pooling	-	2x2	2x2
Dropout	-	-	-
Flatten 512			
Dense 128			
Dropout 0.25			
Dense 10			

Various models such as SVM, KNN, AlexNet, VGG16, ResNet34 and ResNet50 were trained on the BHTP dataset to evaluate the performances as shown in Table 3. The PatternNet model with 4 convolutional layers was found to have the highest accuracy of 99.25%. It was also discovered that the PatternNet has the best precision, recall, and f1-score compared to other trained models. It recorded the training time (8 minutes 26 seconds). The graphical representation of all the trained models performance based on accuracy is shown in Figure 28.

Table 3 Comparisons of different model performances

Model	Accuracy	
	Training	Validation
SVM	86.53%	
KNN	83.33%	
VGG16	99.65%	98.37%
AlexNet	99.52%	97.57%
ResNet34	99.78%	87.63%
ResNet50	99.51%	97.52%
PatternNet [proposed]	**99.75%**	**99.25%**

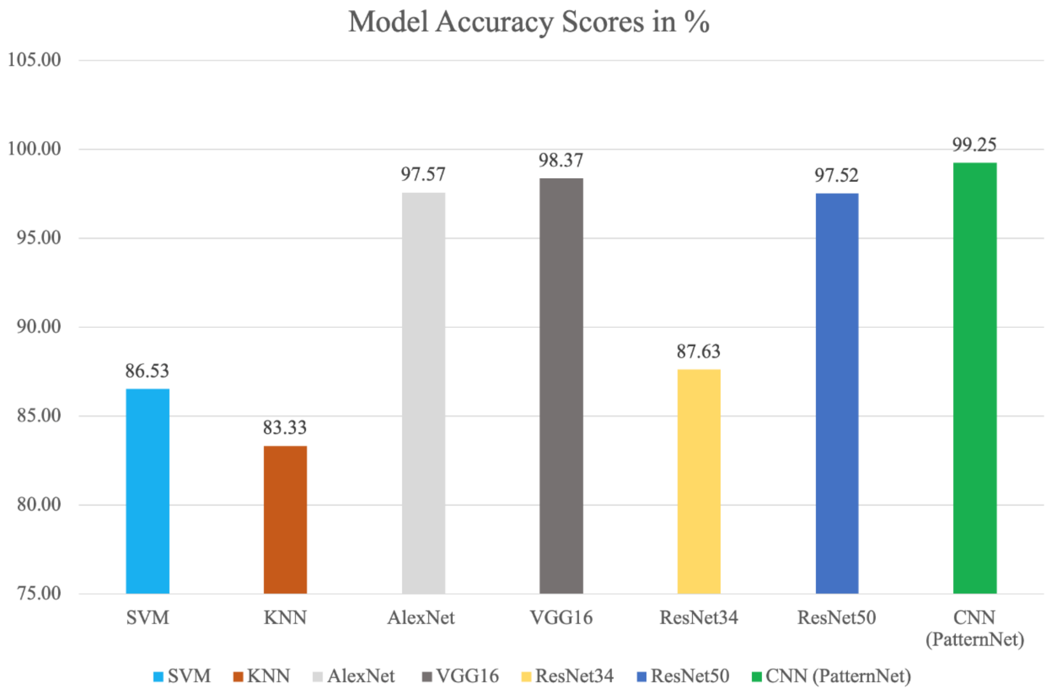


Figure 28 Comparisons of different model performances

Figure 29 illustrates the comparisons of accuracies and losses during training and testing the model. The model learned until the 31st epoch, after which it stopped without further learning.

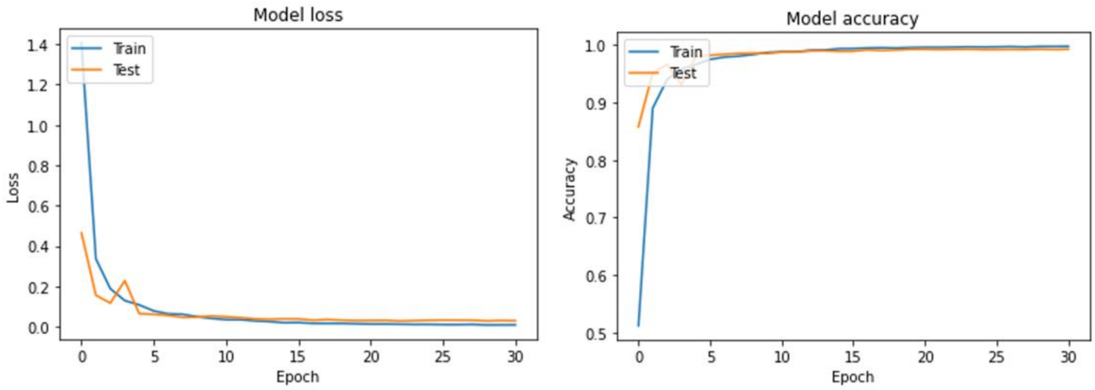


Figure 29 Training and Testing Performances: (Right) Accuracy Vs. Epochs, (Left) Loss Vs. Epoch

The confusion matrix for the PatternNet model validation result is shown in Figure 30. There were 45 misclassifications out of 6000 validation images. The class 4 (Mehub Tima) has the highest misclassification among 10 classes of patterns. Class was 10 times misclassified as class 6 (Phyemali Tren) and 2 times misclassified as class 8 (Thala). Class 6 (Phyemali) has the second-highest misclassifications. It was 6 times misclassified as class 4 (Mehub Tima) and 1 time each as class 1 and class 8. Followed by class 5 (Pagoda) with 7 misclassifications and the rest of the pattern classes had less than 4 misclassifications.

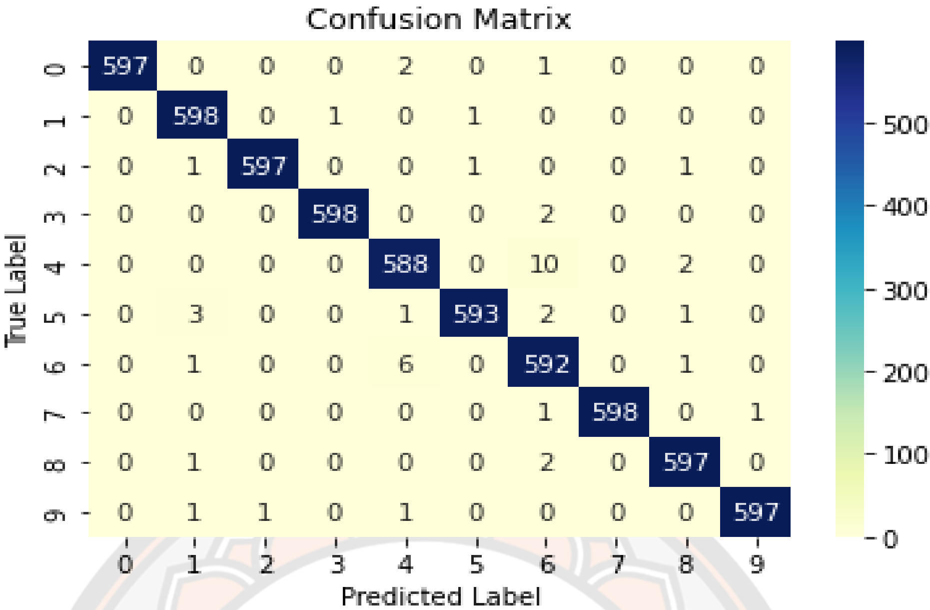


Figure 30 Confusion Matrix for BHTP Classification

The reasons for these misclassifications are due to inter-class pattern similarities, intra-class pattern variations, pattern backdrops, and incorrect augmentation methods. For instance, Figure 31 shows the intra-class pattern variation where pattern/class 4 (*mehub)* has variation within the same class. The patterns fall under the same category, but they are not identical, hence the model predicts that they are different patterns.

Figure 31 Sample images for intra-class feature variation of Pattern 4 (Mehub)

Misclassifications are also caused by similarities among inter-class patterns, as shown in Figure 32. Aside from the point of the edges, (a) pattern 4 (*mehub*) and (b) pattern 6 (*phyemali*) have comparable designs especially at the centre of the pattern. Thus, these two patterns were often misclassified as each other.

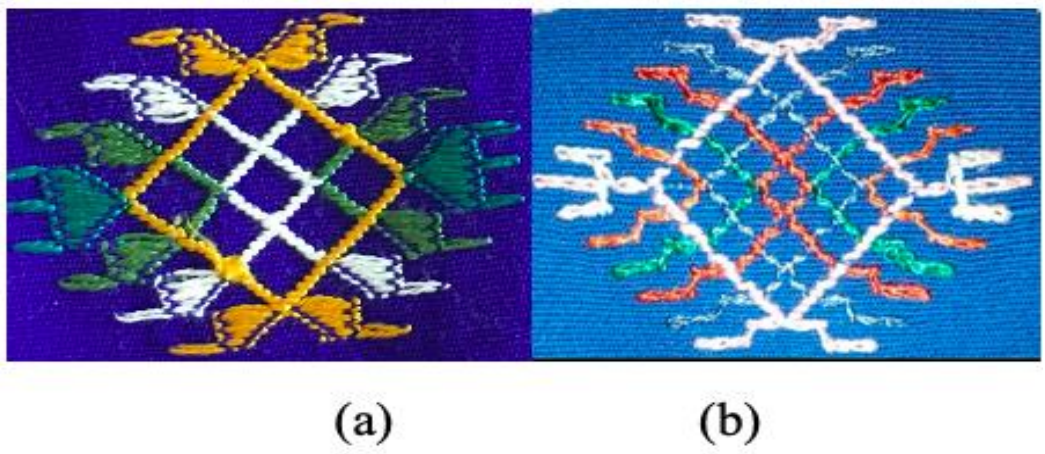


Figure 32 Sample images for inter-class feature similarities

The model also misclassified some patterns because of the redundant patterns in the background. In Figure 33, the background of the first two patterns contains other irrelevant patterns, causing the model to anticipate them as other patterns. Another reason for pattern misclassification is due to improper image augmentation during data processing. Some portions of the pattern are cut off while the image is rotated as shown in Figure 33, causing it to be misclassified as other patterns.

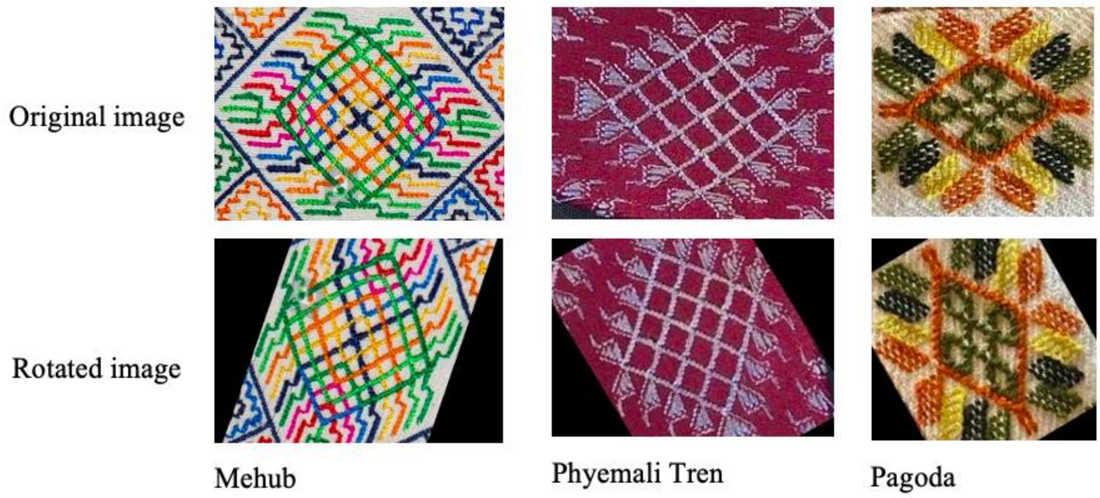


Figure 33 Sample images showing effects of rotated images

The classification report of accuracy, recall, and F1-score percentage for each class is shown in Figure 34. It was discovered that the lowest precision is 97% for class 7 and minimum recall is 98% for class 4. The minimum f1-score was 98% for classes 4 and 6. The weighted average, on the other hand, increased to 99%. The model's weight was saved and loaded in Visual Studio Code using Python and TensorFlow with an OpenCV. The model was used to classify unseen Bhutanese Textile Patterns by deploying the model on a web application.

Classification Report

	precision	recall	f1-score	support
Dorji	1.00	0.99	1.00	600
Gemse	0.99	1.00	0.99	600
Karma	1.00	0.99	1.00	600
Karsi	1.00	1.00	1.00	600
Mehub	0.98	0.98	0.98	600
Pagoda	1.00	0.99	0.99	600
Phyemali	0.97	0.99	0.98	600
Shinglo	1.00	1.00	1.00	600
Thala	0.99	0.99	0.99	600
Yunrung	1.00	0.99	1.00	600
accuracy			0.99	6000
macro avg	0.99	0.99	0.99	6000
weighted avg	0.99	0.99	0.99	6000

Figure 34 Accuracy, recall, and F1-score

The training results obtained reveal that the PatternNet is the best model to recognize or classify the Bhutanese handwoven textile patterns. Figure 30 displays the correctly classified and misclassified patterns in a confusion matrix. During validation, 5955 patterns out of 6000 images were correctly recognized as their respective classes. However, 45 patterns were misclassified. Other models, such as SVM, KNN, AlexNet, ResNet34, ResNet50 and VGG-16, were also trained and the results were compared, with the standard proposed network coming out on top.

4.3 Bhutanese Textile Type classification and Recognition

To train models to recognize and classify Bhutanese textile types, convolutional neural networks (CNN) and machine learning approaches were applied. A dataset of 7 Bhutanese textiles was created for the training. BTT dataset consisted of 8400 images which were later split into 80% and 20% for training and validation respectively. The dataset started with 700 images (100 for each class) and was later enlarged by applying image processing techniques. The training set had 6720 images, and the validation set had 1680 images. The images were resized to 64x64, formatted, and then serialized using the Python pickle module. The serialized data were uploaded on Google Drive for the training. The images were again rescaled into 64x64x3 at the time of training on Google Collaboratory. It takes significantly less time to upload a serialized dataset to Google Drive than it does to submit a raw dataset. To classify the different types of Bhutanese textiles, a VGG16 architecture was trained. To compare performance, AlexNet, SVM, and KNN were trained on the same dataset. VGG16 network outperformed the other models with an accuracy of 99% and 98.88% for training and validation respectively. The VGG16 architecture is shown in Figure 35.

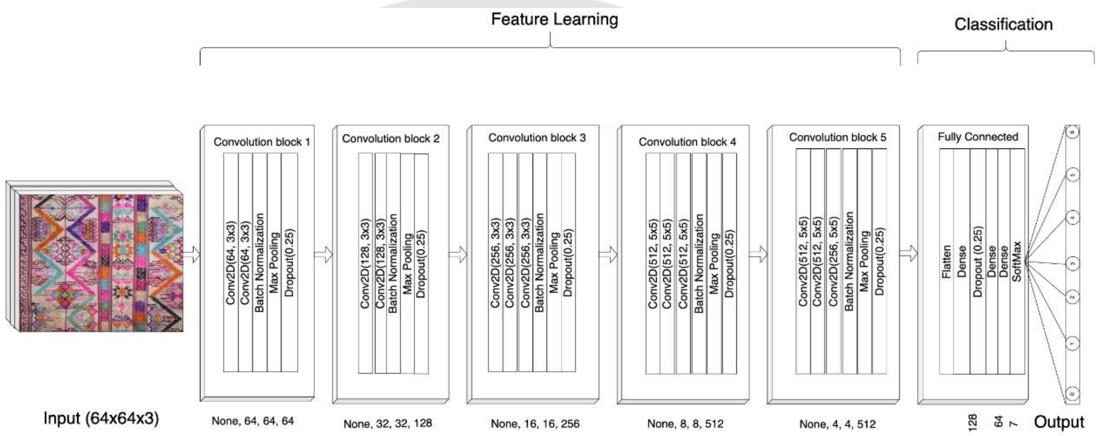


Figure 35 VGG16 Architecture for BTT dataset

VGG16 works well on both the ImageNet dataset and photos that aren't in ImageNet. The colour images were fed into the input layer as inputs. In the hidden layer, the model is comprised of 5 convolutional blocks with 13 convolutional layers and 3 fully connected layers. Batch normalization, max pooling and dropout layers were implemented to speed up the training as well as to avoid the overfitting problem. The configuration for VGG16 is shown in Table 4.

Table 4 Configuration of VGG16 model for BTT dataset

Type of layer	Number of filters	Filter size/stride	Output
Convolution	64	3x3/1	64x64
Convolution	64	3x3/1	64x64
Batch Normalization	-	-	-
Max Pooling	-	2x2/2	32x32
Dropout	-	-	-
Convolution	128	3x3/1	32x32
Convolution	128	3x3/1	32x32
Batch Normalization	-	-	-
Max Pooling	-	2x2/2	16x16
Dropout	-	-	-
Convolution	256	3x3/1	16x16
Convolution	256	3x3/1	16x16
Convolution	256	3x3/1	16x16
Batch Normalization	-	-	10x10
Max Pooling	-	2x2/2	8x8
Dropout	-	-	-
Convolution	512	5x5/1	8x8
Convolution	512	5x5/1	8x8
Convolution	512	5x5/1	8x8
Batch Normalization	-	-	-
Max Pooling	-	2x2/2	4
Dropout	-	-	-
Convolution	512	5x5/1	4x4
Convolution	512	5x5/1	4x4
Convolution	512	5x5/1	4x4
Batch Normalization	-		-
Max Pooling	-	2x2/2	2
Dropout	-	-	-

Flatten (2048)
Dense (128)
Dropout
Dense (64)
Dense (7)

To extract the features, the RGB images are fed into hidden layers as input images and convolved with the filters passed in this layer. Learning becomes more efficient when batch normalization is applied, and it can also be used to prevent model overfitting. The pooling layer shrinks the input images' spatial dimensions and operates on each function map independently. Pooling aims to combine the most noticeable features of a photograph by maintaining the most important information while discarding the rest.

The kernel extracts the highest value from the area it convolves using a max-pooling technique. A dropout is a regularization technique that prevents overfitting in a neural network by eliminating units or neurons. It is the most often used regularization technique for deep neural networks. In neural networks, it aids in the decrease of interdependent learning. The network's final layers receive the output of the pooling layer and apply it by categorizing the images with separate labels and flattening them into probability values that reflect a class.

The VGG16 model's parameters were fine-tuned and trained until they reached the highest level of accuracy. The SoftMax classifier was then used to classify Bhutanese textiles into seven different categories. The output layer displays the final probability values or scores, which aid in image classification.

The performance of Bhutanese textile type recognition was examined and compared to four models. The comparisons of trained models on the BTT dataset are shown in Table 5 as well as in Figure 36. The comparison is based on models' accuracy, precision, recall, and F1-score.

Table 5 Performance comparisons of different models on the BTT dataset

Model	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
SVM	79	80	79	79
KNN	76	80	76	76
Alex Net	95	96	96	96
VGG16	**98.33**	**98**	**98**	**98**

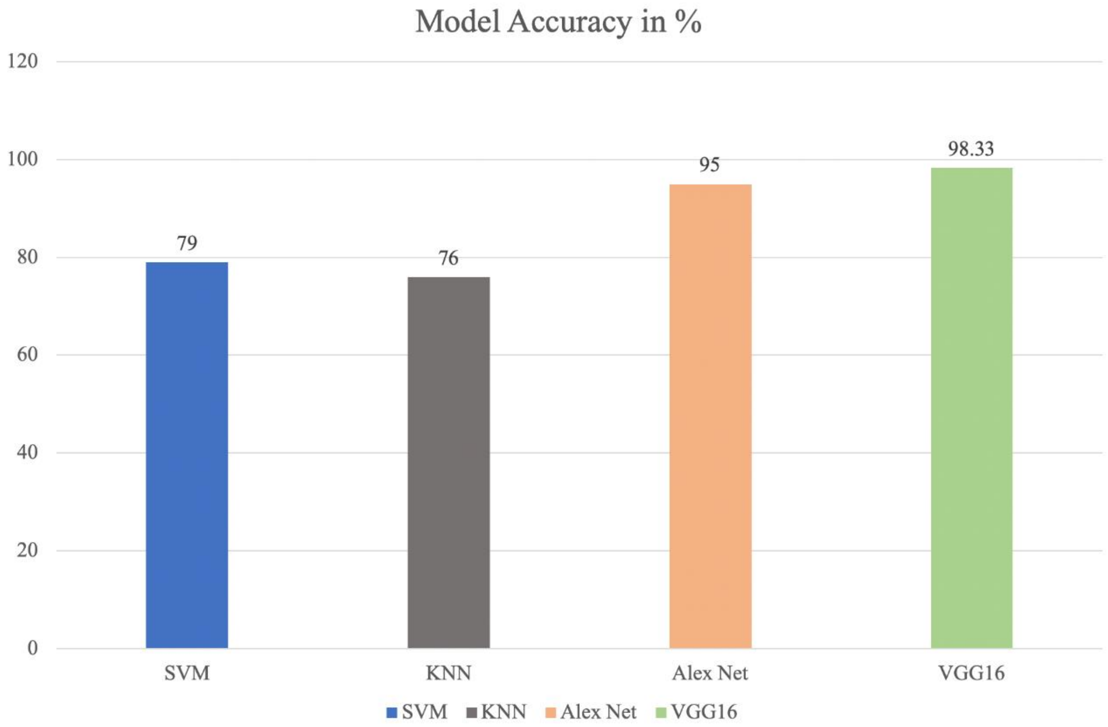


Figure 36 Comparisons of different model performances

VGG16 model was configured to train for 100 epochs and a batch size of 32. The VGG16 model outperformed the rest of the models with a 98.33% accuracy rate. The model learned until the 35th epoch, after which it stopped learning because there was no progress. Early stopping was adopted to avoid difficulties with overfitting during training. Figure 37 shows the accuracy and loss of training and testing between 100 epochs.

The accuracy of both the train and the test grew dramatically in the first 5 epochs. However, after 17 epochs, learning ceased, and accuracy remained constant. The loss decreased at first, but after 5 epochs, it plateaued. Overall, accuracy improved while loss decreased, with neither overfitting nor underfitting evident.

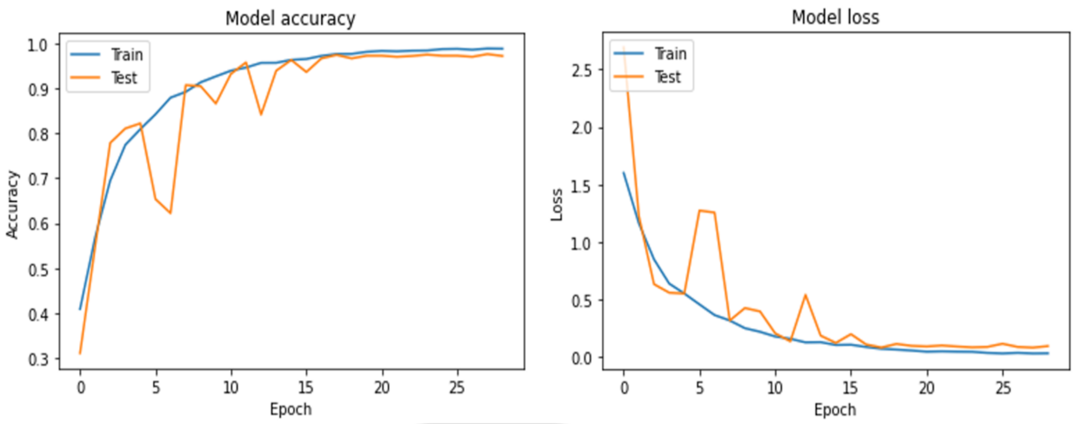
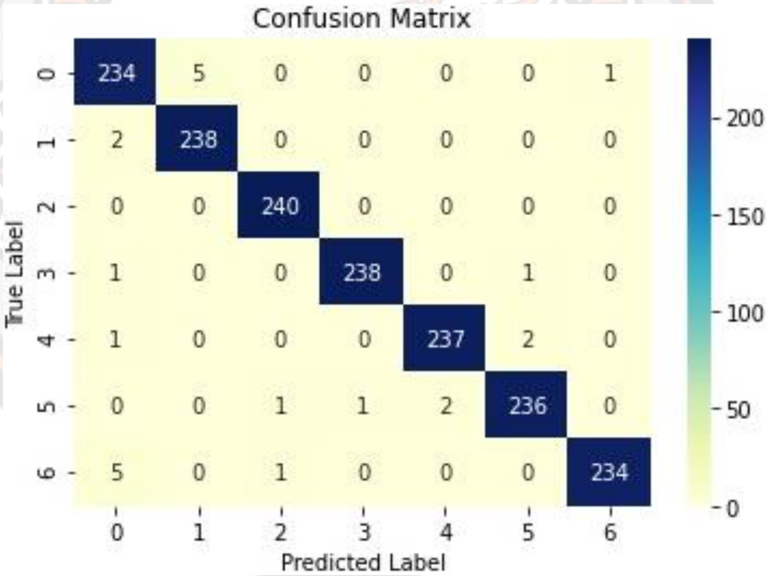


Figure 37 Training and Testing Performances: (Right) Accuracy Vs. Epochs, (Left) Loss Vs. Epoch



	0	1	2	3	4	5	6
0	234	5	0	0	0	0	1
1	2	238	0	0	0	0	0
2	0	0	240	0	0	0	0
3	1	0	0	238	0	1	0
4	1	0	0	0	237	2	0
5	0	0	1	1	2	236	0
6	5	0	1	0	0	0	234

Figure 38 Confusion Matrix for BTT classification

The confusion matrix in Figure 38 shows the performance of the Bhutanese textile type recognition model. There were 1680 images in the validation set, with 240 images from each class. There were 1651 images accurately predicted into their appropriate classes, while 29 images were incorrectly predicted. Classes 0 (Hor) and 6 (Shinglo) had the highest misclassification results. Class 0 (Hor) was 5 times misclassified as class 1 (Jadrima) because these two classes exhibit similar horizontal patterns on the textiles. Class 0 (Hor) was once misclassified as class 6 (Shinglo) and class 6 (Shinglo) was incorrectly predicted 5 times as class 0 (Hor). This is because of the similar horizontal lines and the motifs present in both the textiles. Class 2 (Marthra) has no misclassification.

Figure 39 Misclassifications (vertical & horizontal lines and similar motifs (patterns)

Generally, the misclassifications are mainly caused due to common features such as vertical & horizontal lines and similar motifs (patterns) on the textiles as shown in Figure 39.

Furthermore, the undesirable backgrounds and multiple textile types/patterns on the textiles lead to more misclassifications. Similar patterns and unwanted backgrounds added while capturing the images were the reasons for the misclassification of the textiles as shown in Figure 40.

Figure 40 Misclassification (Similar patterns and unwanted backgrounds)

Table 6 shows the scores for precision, recall, and f1-score of each class. The precision score was highest for classes 1, 3, 4 and 5 with 99% and lowest for classes 0 and 2 with 97%. Classes 2, 4, and 6 had the highest recall score of 99% and classes 1 and 5 had the lowest recall score of 97%. Classes 3, 4, and 6 had 99% for F1-score and the rest of the classes had 98%. The weighted average for all precision, recall, and the f1-score was 98 %.

Table 6 Tabulation of Precision, Recall, and F1-score for each class

Class	Precision (%)	Recall (%)	F1-Score (%)
0 (Hor)	97	98	98
1 (Jadrima)	99	97	98
2 (Kishuthara)	97	99	98
3 (Marthra)	99	98	99
4 (Pangtse)	99	99	99
5 (Serthra)	99	97	98
6 (Shinglo)	98	99	99
Weighted Avg	98	98	98

4.4 Deployment BHTP Recognition Web application

PatternNet model is the best model for recognizing and classifying Bhutanese Textile Patterns. The model was deployed on the web using the Flask framework for the prediction of new Bhutanese textile patterns. The model was hosted on the local server to test model.

A web application framework is a set of libraries and modules that enable web application developers to construct applications without having to worry about protocol and thread management.

A micro-framework is a basic framework that gives developers the liberty to create the web application layer. A developer would not need to set up many things in a micro-framework compared to an enterprise framework to get the web app hosted and functioning. This is very effective for small web app development where the needs aren't the same as an enterprise-level framework, as it saves a lot of time and money in development and maintenance.

Flask is a Python-based micro-framework for web application development. It was created by Armin Ronacher, who is the founder of Pocco, an international association of Python fans. The Werkzeug WSGI toolkit and the Jinja2 template engine are the foundations of Flask.

The Web Server Gateway Interface (WSGI) is the industry standard for developing Python web applications. The Web Server Gateway Interface (WSGI) is a specification for a universal interface between the web server and web applications. Werkzeug is one of the most advanced WSGI modules, containing several tools and utilities that aid in the creation of online applications. Werkzeug is implemented in Flask. Jinja2 is a popular Python templating engine. To render dynamic web pages, a web templating system combines a template with a specific data source.

Flask's goal is to keep an application's core basic but expandable. Flask lacks a built-in abstraction layer for database management, as well as functionality for form validation. Instead, Flask allows you to use extensions to add this functionality to your program. Flask has two significant advantages: it is simple to set up and operate, and it gives you complete control over the web application's structure.

4.4.1 BHTP Recognition Web application

The application will read the input image provided by the user and predict the pattern type for it. The application can classify the images into 10 labelled classes such as Dorji, Gemse, Karma, Karsi Chi, Mehub Tima, Pagoda, Phyemali Tren, Shinglo, Thala, and Yunrung. The predicted pattern type, as well as the top three patterns with their probability scores, are displayed on the output page. If the highest probability score is less than 50%, the system does not decide the prediction result, rather it suggests users try another input image with the message "*Prediction is <50%, try another image*". Figure 41 shows the overall flow of the BHTP Recognition web application.

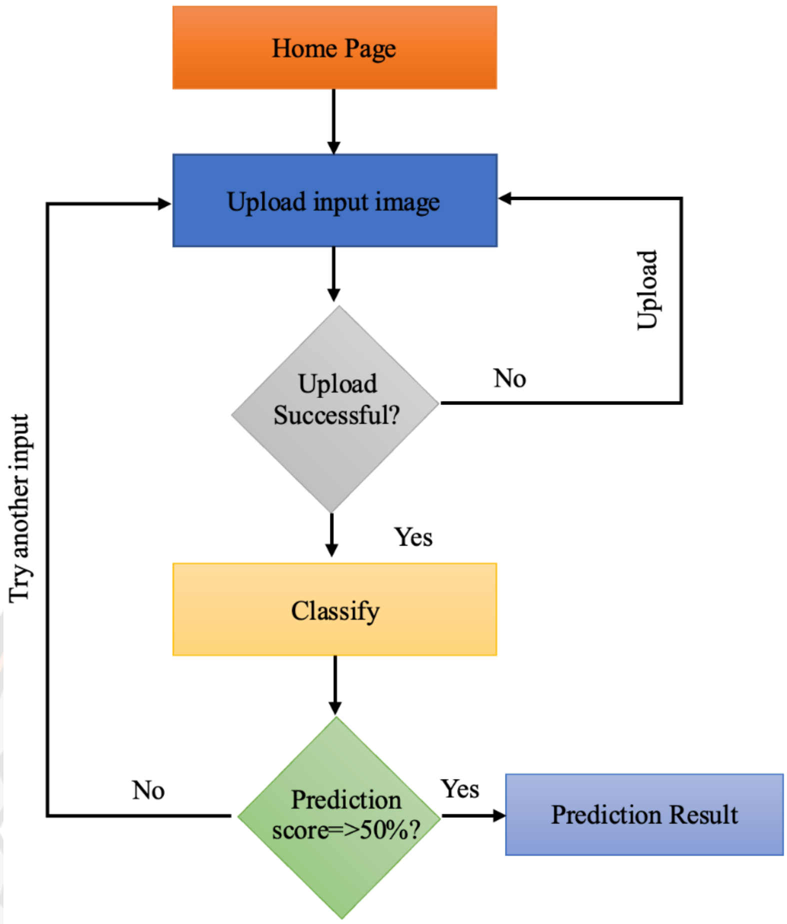


Figure 41 Overflow of the BHTP Recognition web application

The working of the application is explained in the following section. On the homepage (Figure 42), it has an option to upload the input image for the classification. The user should choose and upload the images from their local devices and submit them to the application to identify the pattern category (Figure 43). If no image is selected or an unsupported file format is selected for the classification, an error message is displayed with an error message "*Please upload images of jpg, jpeg, and png format only*" (Figure 44). The name and extension of the selected or uploaded input image will be then displayed (Figure 45). The input image can be changed by clicking on the '*upload*' button. Clicking the '*Classify*' button will take you to the result page. In the prediction result, the input image and the top three pattern probability scores are displayed (Figure 46). The pattern class with the highest probability score is the prediction result and it is displayed at the end of the result page. For example, in (Figure 46) the app correctly recognized the input pattern as Dorji with 100% probability, and thus the result is Dorji. However, if the prediction result (probability score) is less than 50%, an error message will be displayed saying "*Prediction is <50%, try another image*" (Figure 47).

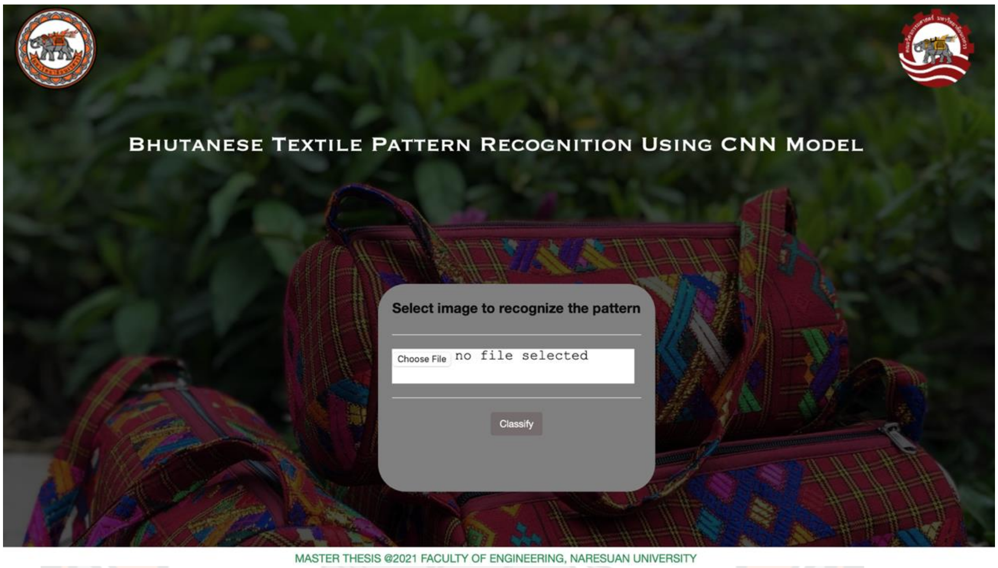


Figure 42 Home Page

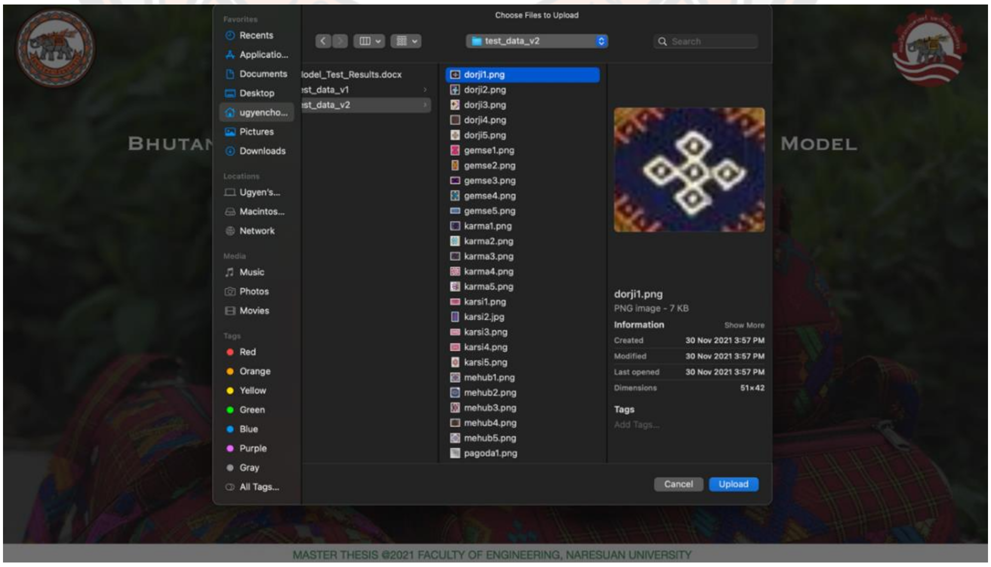


Figure 43 Input Image Upload Page

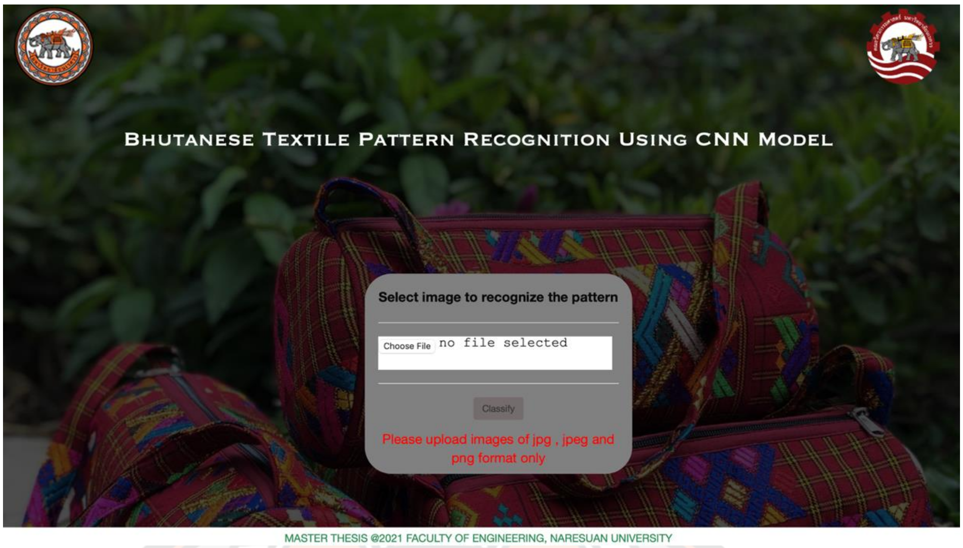


Figure 44. Error Page

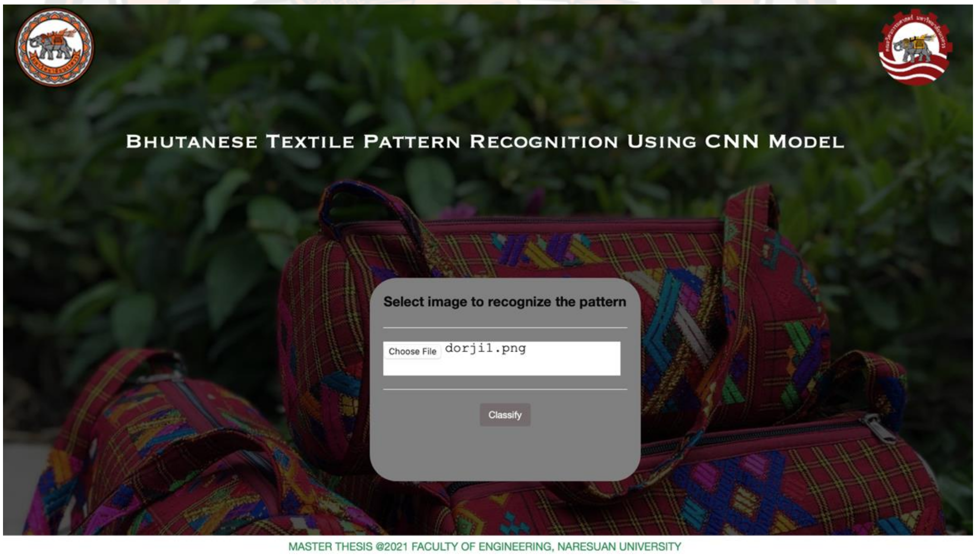


Figure 45. Uploaded Image details

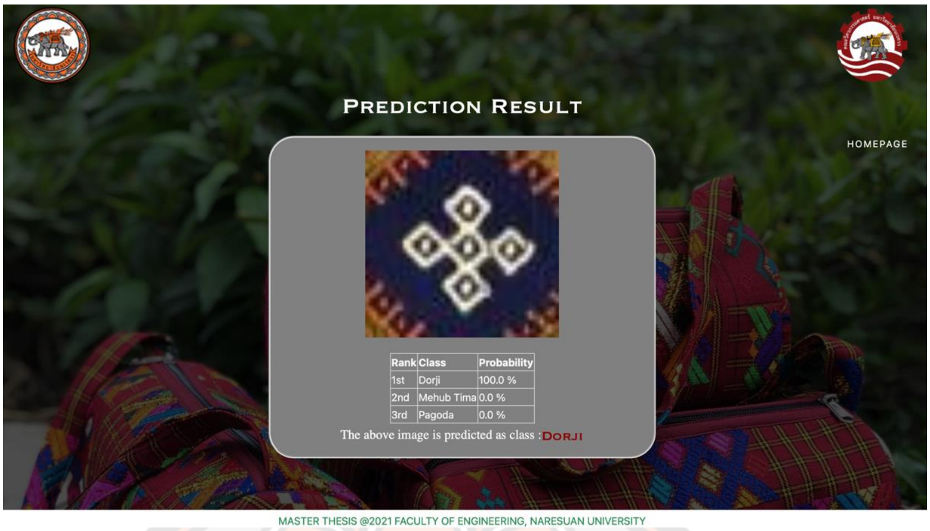


Figure 46 Classification Result Page case1: Correct Prediction

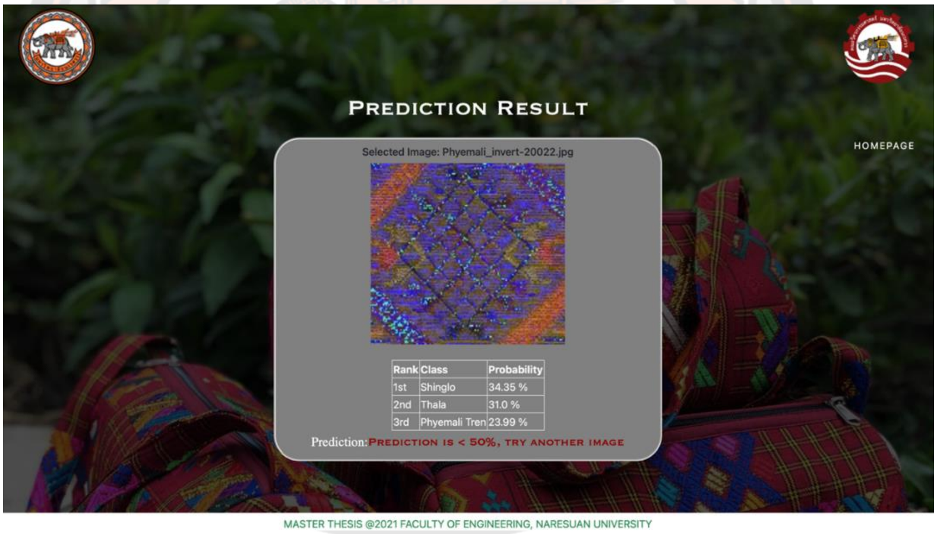


Figure 47 Classification Result Page case2: Incorrect Prediction

4.4.2 Testing Result: BHTP Recognition Web application

There were 55 testing images for each class, for a total of 550 images. Out of 550 images, only 8 images were incorrectly predicted, and the rest are all correctly predicted. The test data contains similar images like that of the images in the training and validation data because the images are pre-processed, augmented, and then split into the train, validate, and test sets.

Figure 48 shows the misclassifications during testing the model deployed on a web application.

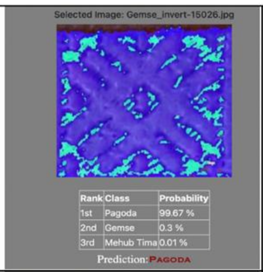
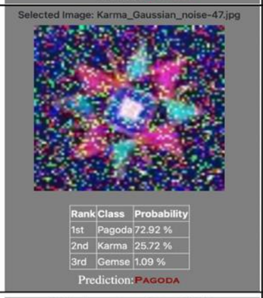
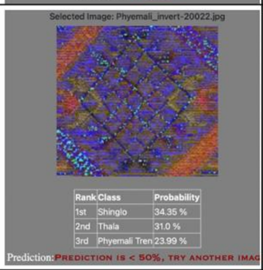
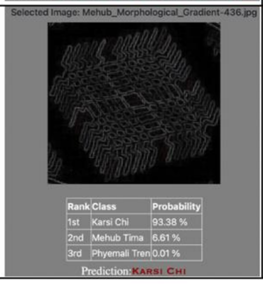
<table>
<tr><th>Input pattern</th><th colspan="2">Predicted as</th><th></th><th>Input pattern</th><th colspan="2">Predicted as</th></tr>
<tr><td>Gemse</td><td>
</td><td>Pagoda</td><td></td><td>Dorji</td><td>
</td><td>Mehub Tima</td></tr>
<tr><td>Karma</td><td>
</td><td>Pagoda</td><td></td><td rowspan="3">Pagoda</td><td>
</td><td rowspan="2">Phyemali Tren</td></tr>
<tr><td>Phyemali Tren</td><td>
</td><td>Shinglo</td><td></td><td>
</td></tr>
<tr><td>Mehub Tima</td><td>
</td><td>Karsi Chi</td><td></td><td>
</td><td>Mehub Tima</td></tr>
</table>

Figure 48 Misclassifications during testing

Table 7 shows the confusion matrix for testing the PatternNet model on a web application.

Table 7 Confusion Matrix for BHTP Recognition web application

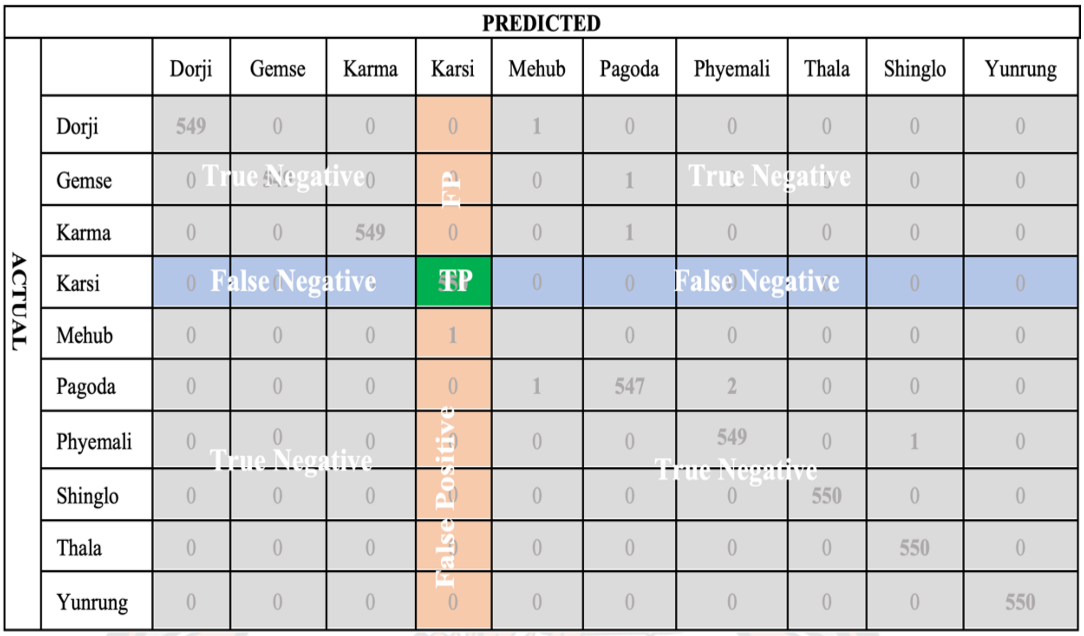
ACTUAL \ PREDICTED	Dorji	Gemse	Karma	Karsi	Mehub	Pagoda	Phyemali	Shinglo	Thala	Yunrung
Dorji	54				1					
Gemse		54				1				
Karma			54			1				
Karsi				55						
Mehub				1	54					
Pagoda					1	52	2			
Phyemali							54		1	
Shinglo								55		
Thala									55	
Yunrung										55

Pagoda has the highest misclassification where 3 images out of 550 testing images were incorrectly predicted as Mehub and Phyemali. This is because the pagoda has similar patterns as that of Mehub and Phyemali. The testing accuracy is for the PatternNet model application is 98.54%. It is calculated as below.

$$Accuracy = \frac{\text{Total number of correct prediction}}{\text{Total testing data}} * 100$$

$$Accuracy = \frac{542}{542} * 100 = 98.54\%$$

Figure 49 shows the FP, FN, TP, and TN of the multi-class classification result. *False Positive Rate [Type I Error]* is the number of items wrongly identified as positive out of the total actual negatives *[FP/(FP+TN)]*. *False Negative Rate [Type II Error]* is the number of items wrongly identified as negative out of the total actual positives *[FN/(FN+TP)]*.



PREDICTED										
ACTUAL	Dorji	Gemse	Karma	Karsi	Mehub	Pagoda	Phyemali	Thala	Shinglo	Yunrung
Dorji	549	0	0	0	1	0	0	0	0	0
Gemse	0	[illegible]	0	[illegible]	0	1	[illegible]	[illegible]	0	0
Karma	0	0	549	0	0	1	0	0	0	0
Karsi	0	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	0	0
Mehub	0	0	0	1		0	0	0	0	0
Pagoda	0	0	0	0	1	547	2	0	0	0
Phyemali	0	0	0	[illegible]	0	0	549	0	1	0
Shinglo	0	0	0	[illegible]	0	0	0	550	0	0
Thala	0	0	0	[illegible]	0	0	0	0	550	0
Yunrung	0	0	0	0	0	0	0	0	0	550

Figure 49 Sample Confusion Matrix

Table 8 Individual class performances for BHTP Recognition web application testing

Class 0 (Dorji)	Class 1 (Gemse)	Class 2 (Karma)	Class 3 (Karsi)	Class 4 (Mehub)
TP: 549 FN: 1 FP: 0 TN: 4950	TP: 549 FN:1 FP:0 TN: 4950	TP: 549 FN:1 FP:0 TN: 4950	TP: 550 FN: 0 FP: 1 TN: 4949	TP:549 FN:1 FP:1 TN: 4949
Class 5 (Pagoda)	**Class 6 (Phyemali)**	**Class 7 (Shinglo)**	**Class 8 (Thala)**	**Class 9 (Yunrung)**
TP: 547 FN: 3 FP:2 TN:4948	TP: 549 FN:1 FP:2 TN:4948	TP:550 FN:0 FP:0 TN: 4950	TP:550 FN:0 FP:1 TN:4949	TP:550 FN:0 FP:0 TN:4950

Class 5 (Pagoda) has 3 False Negatives. Dorji, Karsi, Phyemali, Gemse, Mehub and karma has 1 FN each. Class 6 (Phyemali) and class 5 (Pagoda) have the highest False Positive (FP) value 2. Classes 3, 4 and 6 have one FP each however, classes 0,1,2,5,7 and 9 do not have any FP.

4.4.3 BHTP Recognition Web application (PatternNet) Limitations

The PatternNet model can predict the pattern correctly if an input image contains only the required pattern. As shown in Figure 50 (left), the prediction of the full image with more than one pattern gives an incorrect prediction whereas Figure 50 (right) shows the correct prediction when the input image contains the single pattern. This is because the input images were all cropped and featured a single pattern during training the model.

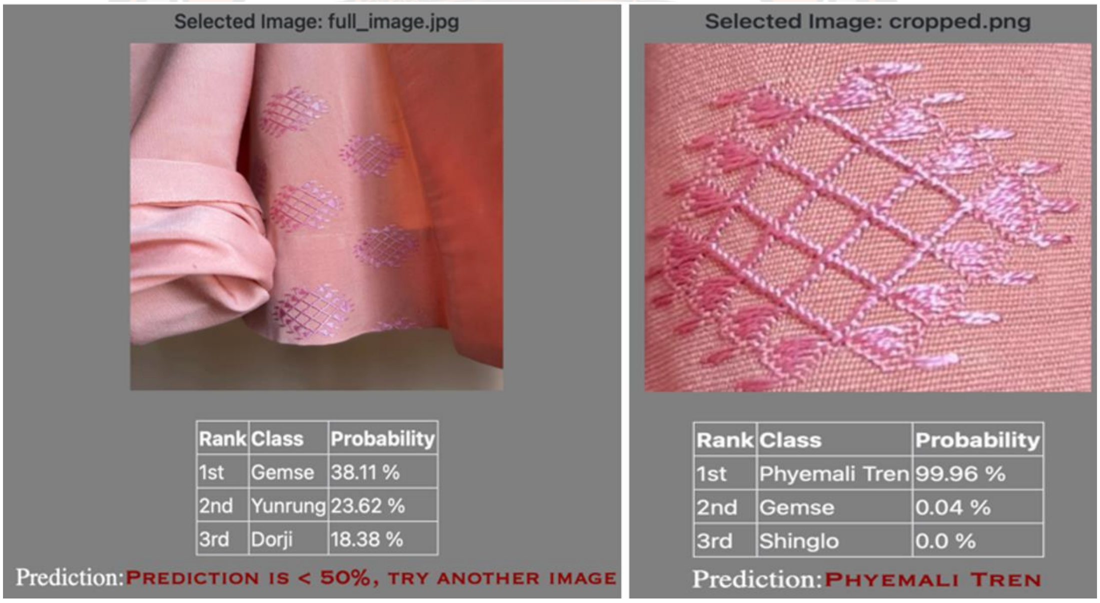


Figure 50 Left (Correct Prediction) and right (Incorrect Prediction)

During the prediction, the image size also matters. As shown in Figure 51, the left image size is 1050x962, and the right image size is reduced to 186x171. When the input size is large (1050x962), the input pattern is wrongly predicted as Karsi Chi. However, the input pattern is correctly predicted as Mehub Tima when the input image is downsized to 185x171. This is because the size of all the training images was 64x64 pixels. Furthermore, before predicting, the classification application will convert the image to 64x64, which results in the loss of features.

The images in the training data do not have a diversity of angles or orientations. Figure 52 is class 5 (Pagoda) on the left and its rotated version on the right. The model predicted the pattern incorrectly as Mehub Tima, but when it is rotated 90 degrees, the model correctly predicts it as Pagoda. This is due to a lack of angle and orientation diversity in the training data/images which can be addressed in future training.

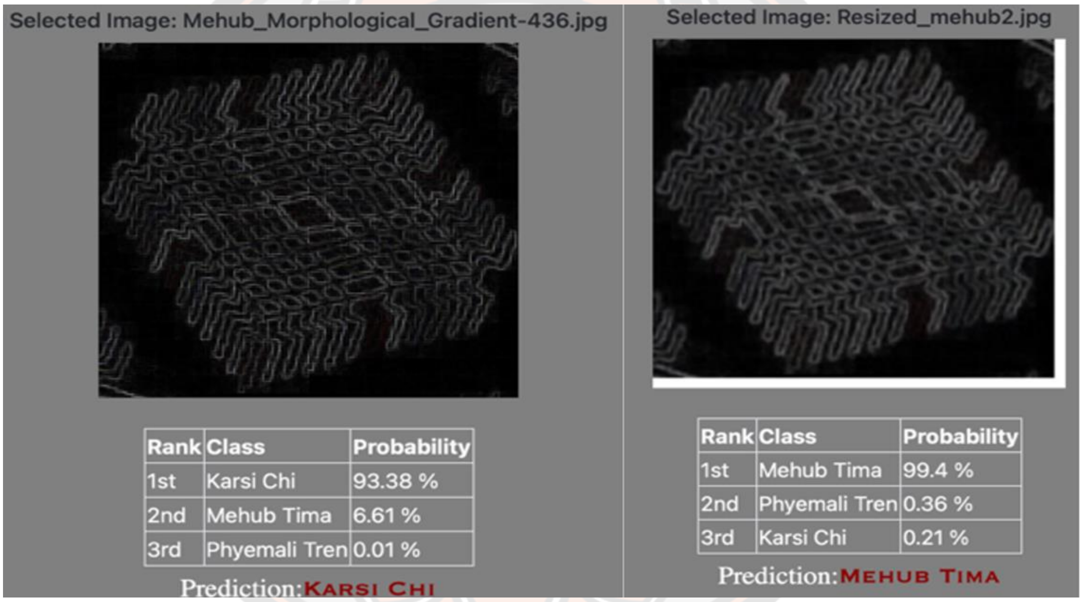


Figure 51 Mehub Tima; left (1050x962) and right (185x171)

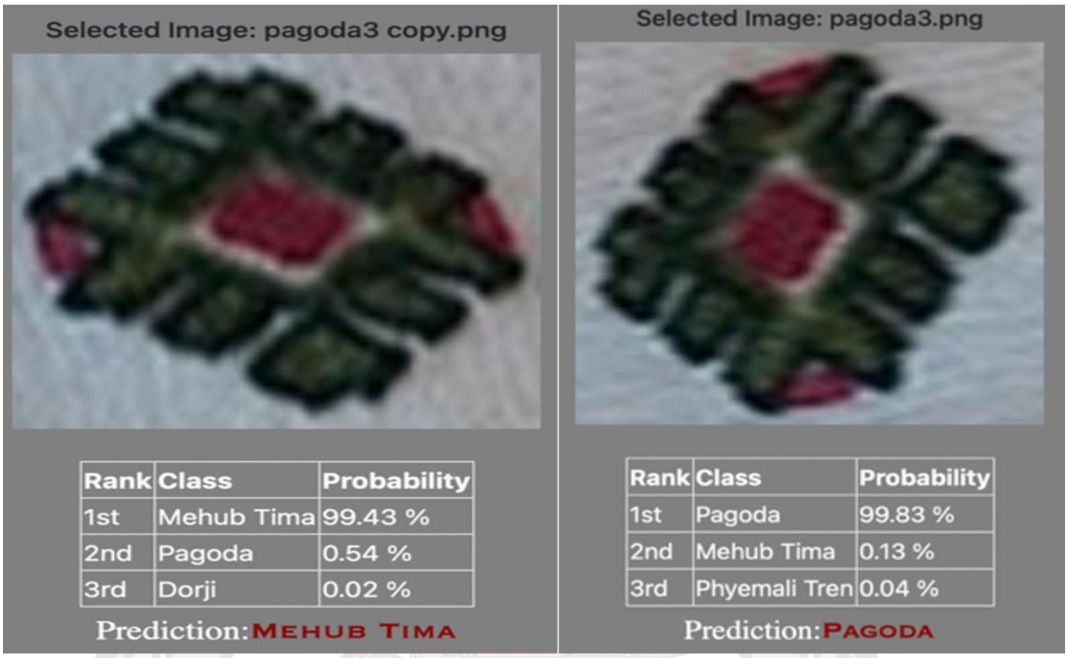


Figure 52 left (Pagoda) and right (Pagoda rotated)

CHAPTER V

CONCLUSION

5.1 Introduction

In Chapter I, the study's background was discussed, followed by literature reviews in Chapter II. The methodology was discussed in depth in Chapter III, with a discussion of the results in Chapter IV and a conclusion in Chapter V. A summary of the thesis is offered first in the conclusion section, followed by its limitations and future study

5.2 Study Summary

Chapter I delves into the background and significance of the study. The study's purpose, problem statements, and scopes were all thoroughly discussed in this chapter. The main goal of this study is to prepare Bhutanese textile datasets and train various machine learning and neural network models to recognize the textile types and their patterns. The study was prompted by causes such as the duplication of Bhutanese textiles, the vanishing ancient textile patterns, modernization and globalization, and the lack of appropriate technology for textile recognition and storage. In general, the study's scope includes dataset preparation, model training on the datasets, and deployment of the selected model for testing.

Chapter II discusses the various approaches of pattern recognition such as template matching, statistical approach, syntactic approach, and neural networks. The study explored machine learning algorithms for diverse textile pattern recognition which describes the present state of the art. It was found that the Bhutanese textiles can be seen and purchased in a few textile shops, as well as a few museums and textile academies in Bhutan. However, no technology exists that can save the archive of these fabrics for future use. Furthermore, Bhutanese textile knowledge is being shared through verbal communication and some written documentation. Furthermore, no study has been done on building technology to recognize and classify Bhutanese textile patterns automatically.

To address the problems identified in Chapter I led to this study emphasized preparing Bhutanese textile datasets and training various machine learning and neural network models to recognize the textile types and their patterns. The first-ever Bhutanese Textile datasets called BHTP and BTT were also curated and designed. Two datasets were created: 30,000 images for the BHTP dataset and 8400 images for the BTT dataset. These two datasets are used to evaluate the suitable machine learning models.

Chapter III illustrates the methodology of this study. Bhutanese textile datasets were analysed using different textile recognition models based on the literature reviews in Chapter II. To extract features from the input images, the CNN algorithm was employed, with parameters fine-tuned to suit the datasets. PatternNet model with six convolutional layers, batch normalization, and dropout outperformed SVM, KNN, VGG16, ResNet34, ResNet50, and AlexNet for handwoven textile pattern recognition with 99.75% and 99.25% accuracy for training and validation respectively. Similarly, the VGG16 of 13 convolutional layers and 3 dense layers with batch normalization and dropout outperformed SVM, KNN, and AlexNet with 99% training accuracy and 98.33 % validation accuracy for textile type recognition.

Furthermore, the viability of Computer Vision applications with Bhutanese textiles was studied in this study. This is the first time that Computer Vision has been used with Bhutanese textiles. The Bhutanese textile dataset was used to test and evaluate various textile recognition algorithms. It was discovered that multiclass models with many classes take longer to train. However, utilizing pickle to serialize images (byte streams) lowered the training time. Machine learning algorithms may be used to build Bhutanese textile recognition apps.

5.3 Limitation of study

As described in Chapter IV, the Bhutanese textile datasets exhibit remarkable accuracy in training the models. However, data collection and model training has some limitations as given below:

1. The dataset size has a high impact on model training. For example, the patterns with similar features are more frequently misclassified. Which could have been solved with the addition of more images to the dataset.

2. Misclassifications resulted from the image's limited angles and perspectives during capture. The image should have been taken from every angle possible.

3. Data augmentation adds diversity to a dataset. However, the dataset's width and height shifts resulted in inaccurate class classification.

4. Only the required single pattern image must be captured for the BHTP dataset. Multiple patterns on a single image led to misclassifications during training and testing.

5. The input images for testing must be 64x64 or similar size because the model was trained using 64x64 images. The image will be incorrectly predicted if large image size is used for testing. For example, an input image for class Mehub Tima with size 1050x962 was incorrectly predicted as Karsi Chi. However, when the same image was resized to 186x171, it was correctly predicted as Mehub Tima with a probability score of 99.4%.

5.4 Future Work

In the future, the dataset size can be expanded by capturing high-resolution images with varied angles. This would help in more accurate classification of similarly featured patterns. The number of textile patterns and type classes can be increased to include the rest of Bhutanese textiles. In addition, the study can be expanded to textile defect detection quality assessment. Furthermore, Generative adversarial networks (GAN) can be studied to create new textile patterns from the existing patterns which could be valuable in the textile manufacturing industries in the future.

REFERENCES

REFERENCES

Academy, R. T. (2016). *Royal Textile Academy*. Retrieved 11 from https://www.rtabhutan.org/about-us/

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET),

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

Andrian, R., Naufal, M., Hermanto, B., Junaidi, A., & Lumbanraja, F. R. (2019). k-Nearest Neighbor (k-NN) Classification for Recognition of the Batik Lampung Motifs. Journal of Physics: Conference Series,

Bera, S., & Shrivastava, V. K. (2020). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing*, *41*(7), 2664-2683.

Bhutan, N. M. o. Retrieved 11 from https://www.nationalmuseum.gov.bt/index.php/about-us/

Bhutan, T. C. o. (2019). *BHUTAN TOURISM MONITOR 2019* (Thimphu: Bhutan: Kuensel Corporation Ltd, Issue. https://www.tourism.gov.bt/uploads/attachment_files/tcb_K11a_BTM%202019.pdf

Boonsirisumpun, N., & Puarungroj, W. (2018). Loei fabric weaving pattern recognition using deep neural network. 2018 15th International joint conference on computer science and software engineering (JCSSE),

Bremner, D., Demaine, E., Erickson, J., Iacono, J., Langerman, S., Morin, P., & Toussaint, G. (2005). Output-sensitive algorithms for computing nearest-neighbour decision boundaries. *Discrete & Computational Geometry*, *33*(4), 593-604.

Bureau, N. S. (2020). *Statistical Yearbook of Bhutan 2020*.

Cai, S., Shu, Y., Chen, G., Ooi, B. C., Wang, W., & Zhang, M. (2019). Effective and efficient dropout for deep convolutional neural networks. *arXiv preprint arXiv:1904.03392*.

Chan, C. C., Li, K., & Ngan, H. Y. (2017). Weaving pattern recognition of ancient chinese textiles by regular bands analysis. *Electronic Imaging*, *2017*(9), 31-36.

Chen, H., Wang, Y.-H., & Fan, C.-H. (2020). A convolutional autoencoder-based approach with batch normalization for energy disaggregation. *The Journal of Supercomputing*, 1-18.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

da Silva BarrosM, A. C., Ohata, E. F., da Silva, S. P. P., Almeida, J. S., & Rebouças Filho, P. P. (2020). An Innovative Approach of Textile Fabrics Identification from Mobile Images using Computer Vision based on Deep Transfer Learning. 2020 International Joint Conference on Neural Networks (IJCNN), Di, Z., Luoqing, Z., & Jun, S. (2013). A novel feedback error-correcting algorithm for automatic recognition of the color and weave pattern of yarn-dyed fabrics. *Textile Research Journal*, *83*(16), 1673-1689.

Feng, M., Chen, D., & Wang, X. (2018). Classification of Fabric Patterns Image Based on Improved Log-AlexNet. 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI),

Fu, S. (2020). The Application of Computer Image Recognition Technology in Textile and Clothing. Journal of Physics: Conference Series,

Hamdi, A. A., Sayed, M. S., Fouad, M. M., & Hadhoud, M. M. (2018). Unsupervised patterned fabric defect detection using texture filtering and K-means clustering. 2018 International Conference on Innovative Trends in Computer Engineering (ITCE),

Hu, M., Qiu, R. T., Wu, D. C., & Song, H. (2021). Hierarchical pattern recognition for tourism demand forecasting. *Tourism Management*, *84*, 104263.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning,

Iqbal Hussain, M. A., Khan, B., Wang, Z., & Ding, S. (2020). Woven Fabric Pattern Recognition and Classification Based on Deep Convolutional Neural Networks. *Electronics*, *9*(6), 1048.

Islam, S. S., Dey, E. K., Tawhid, M. N. A., & Hossain, B. M. (2016). Garments Texture Design Class Identification Using Deep Convolutional Neural Network.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(1), 4-37.

Jing, J., Xu, M., & Li, P. (2015). Automatic recognition of weave pattern and repeat for yarn-dyed fabric based on KFCM and IDMF. *Optik*, *126*(21), 2876-2883.

Kaewkhunok, S. (2018). The Commodification of Culture: Bhutan’s Tourism in Globalisation Context. *Thammasat Review*, *21*(1), 152-164.

Kaur, R. (2014). An Overview of Pattern Recognition Applications and Its Approaches: A Review. *International Journal of Science and Research*, *3*(11), 3179-3181.

Khamket, T., & Surinta, O. (2020). Comparative Study between Texture Feature and Local Feature Descriptors for Silk Fabric Pattern Image Recognition. Proceedings of the 2020 The 3rd International Conference on Information Science and System,

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.

Kumar, G., & Bhatia, P. K. (2014). A detailed review of feature extraction in image processing systems. 2014 Fourth international conference on advanced computing & communication technologies,

Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, *5*, 5858-5869.

Li, J., Wang, W., Deng, N., & Xin, B. (2020). A novel digital method for weave pattern recognition based on photometric differential analysis. *Measurement*, *152*, 107336.

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP journal on wireless communications and networking*, *2017*(1), 1-12.

Liu, Y. H. (2018). Feature extraction and image recognition with convolutional neural networks. Journal of Physics: Conference Series,

Lo, J., Macintyre, L., & Kalkreuter, B. (2016). Investigating Markers of Authenticity: The Weavers' Perspective Insights from a Study on Bhutanese Hand-woven Kira Textiles. *Textile*, *14*(3), 306-325.

Meng, S., Pan, R., Gao, W., Zhou, J., Wang, J., & He, W. (2020). A multi-task and multi-scale convolutional neural network for automatic recognition of woven fabric pattern. *Journal of Intelligent Manufacturing*, 1-15.

Meng, S., Wang, J., Pan, R., Gao, W., Zhou, J., & He, W. (2021). Recognition of the layout of colored yarns in yarn-dyed fabrics. *Textile Research Journal*, 0040517520932830.

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. 2018 international interdisciplinary PhD workshop (IIPhDW),

Minarno, A. E., Azhar, Y., Sumadi, F. D. S., & Munarko, Y. (2020). A Robust Batik Image Classification using Multi Texton Co-Occurrence Descriptor and Support Vector Machine. 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS),

Minarno, A. E., Sumadi, F. D. S., Wibowo, H., & Munarko, Y. (2020). Classification of batik patterns using K-Nearest neighbor and support vector machine. *Bulletin of Electrical Engineering and Informatics*, *9*(3).

Nurhaida, I., Noviyanto, A., Manurung, R., & Arymurthy, A. M. (2015). Automatic Indonesian's batik pattern recognition using SIFT approach. *Procedia Computer Science*, *59*, 567-576.

Obaid, H. S., Dheyab, S. A., & Sabry, S. S. (2019). The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON),

Ohi, A. Q., Mridha, M., Hamid, M. A., Monowar, M. M., & Kateb, F. A. (2021). FabricNet: A Fiber Recognition Architecture Using Ensemble ConvNets. *IEEE Access*, *9*, 13224-13236.

Peng, T., Zhou, X., Liu, J., Hu, X., Chen, C., Wu, Z., & Peng, D. (2021). A textile fabric classification framework through small motions in videos. *Multimedia Tools and Applications*, *80*(5), 7567-7580.

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883.

Puarungroj, W., & Boonsirisumpun, N. (2019). Convolutional Neural Network Models for HandWoven Fabric Motif Recognition. 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON),

Rasyidi, M. A., & Bariyah, T. (2020). Batik pattern recognition using convolutional neural network. *Bulletin of Electrical Engineering and Informatics, 9*(4), 1430-1437.

Salem, Y. B., & Abdelkrim, M. N. (2020). Texture classification of fabric defects using machine learning. *International Journal of Electrical and Computer Engineering*, *10*(4), 4390.

Sharma, P., & Kaur, M. (2013). Classification in pattern recognition: A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(4).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1-48.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sousa, C. E., Medeiros, C. M., Pereira, R. F., Neto, M. A., & Neto, A. A. (2020). Defect Detection and Quality Level Assignment in Wet Blue Goatskin. Proceedings of the 11th International Conference on Advances in Information Technology,

Strączkiewicz, M., Czop, P., & Barszcz, T. (2016). Supervised and unsupervised learning process in damage classification of rolling element bearings. *Diagnostyka*, *17*.

Tang, S., Yuan, S., & Zhu, Y. (2020). Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery. *IEEE Access*, *8*, 149487-149496.

Tshering, D., & Wangchuk, T. (2008). Zorig Chusum: Bhutan’s Living Arts and Crafts. *Bhutan: The Thunder Dragon Comes to Washington*, 30-31.

Umer, S., Rout, R. K., Pero, C., & Nappi, M. (2021). Facial expression recognition with trade-offs between data augmentation and deep learning features. *Journal of Ambient Intelligence and Humanized Computing*, 1-15.

Vasantapan, N., & Chouvatut, V. (2017). Pattern extraction from northern Thai fabrics using flexibly matching segments: Sarong Teenjok and Lanna textiles. 2017 9th International Conference on Knowledge and Smart Technology (KST),

Verma, R., & Ura, K. (2018). Gender differences in Gross National Happiness in Bhutan: abridged analysis of the GNH surveys. *GNH: From Philosophy to Praxis*, 196-247.

Walcott, S. M. (2011). One of a kind: Bhutan and the modernity challenge. *National Identities*, *13*(3), 253-265.

Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, *10*(5), 1897.

Wangchuk, D. (2016). *Foundations of Bhutan History: a concise guide*. 365 Days Book.

Wei, B., Hao, K., Tang, X.-s., & Ren, L. (2018). Fabric defect detection based on faster RCNN. international conference on artificial intelligence on textile and apparel,

Wu, H., & Gu, X. (2015). Towards dropout training for convolutional neural networks. *Neural Networks*, *71*, 1-10.

Yaganegi, K. A. (2014). Druk Thagzo -The Bhutanese Art of Weaving. *CATALOG 2014*, . http://www.apic.org.bt/wp-content/uploads/2014/10/Textile-Catalogue-Druk-ThagZo.pdf

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, *9*(4), 611-629.

Yu, D., Wang, H., Chen, P., & Wei, Z. (2014). Mixed pooling for convolutional neural networks. International conference on rough sets and knowledge technology,

Yudon, S. (2015). Overview of Traditional Weaving (Thagzo) in Bhutan. *Traditional Knowledge and Traditional Cultural Expressions of South Asia*, 232.

Zhang, J., Wu, J., Hu, X., & Zhang, X. (2020). Multi-color measurement of printed fabric using the hyperspectral imaging system. *Textile Research Journal*, *90*(9-10), 1024-1037.

Zheng, D. (2021). Pattern-driven color pattern recognition for printed fabric motif design. *Color Research & Application*, *46*(1), 207-221.