MYANMAR SIGN LANGUAGE RECOGNITION SYSTEM USING SUPPORT VECTOR MACHINE(SVM) AND KERNEL PRINCIPAL COMPONENT ANALYSIS(KPCA)

EAINT THU THU KHAING

M.C.Sc. **DECEMBER 2022**

MYANMAR SIGN LANGUAGE RECOGNITION SYSTEM USING SUPPORT VECTOR MACHINE(SVM) AND KERNEL PRINCIPAL COMPONENT ANALYSIS(KPCA)

By

EAINT THU THU KHAING

B.C.Sc.

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

M.C.Sc.

University of Computer Studies, Yangon

DECEMBER 2022

ACKNOWLEDGEMENTS

To complete this thesis, many things are needed like my hard work and the supporting of many people who gave a lot of idea to me to complete this thesis.

First of all, I would like to thank the Union Minister, the Ministry of Science and Technology, for allowing me the advanced study and providing full facilities during the master degree course at the University of Computer Studies, Yangon.

I would like to express very special thanks to **Dr. Mie Mie Khin**, Rector of the University of Computer Studies, Yangon, for her kind permission to submit this dissertation.

I would like to express very special thanks to **Dr. Khin Sandar Aung**, Pro-Rector of the University of Computer Studies, Taunggyi, for allowing me to develop this thesis and giving me general guidance during the period of my study.

I deeply and specially thank the external examiner, **Dr. Aung Nway Oo,** Professor and Head of Faculty of Computer Science, University of Information Technology, for his patience in critical reading, valuable suggestions and comments in the preparation of thesis.

I would like to thank course coordinators, **Dr. Si Si Mar Win**, Professor, University of Computer Studies, Yangon and **Dr. Tin Zar Thaw,** Professor, University of Computer Studies, Yangon for their superior suggestions and administrative supports during my academic study.

I would like to express my deepest gratitude to my thesis supervisor, **Dr. Yi Mon Shwe Sin**, Lecturer, Natural Language Processing Lab., Faculty of Compute Science, University of Computer Studies, Yangon, for her close supervision, proper guidance, valuable suggestions, advice and encouragement during the course of this work.

I would like to express my respectful gratitude to **Daw Aye Aye Khin,** Lecturer of English Department, for her valuable supports from the language point of view and pointed out the correct usage in my dissertation.

I also like to acknowledge my thanks to **all my teachers** who taught me throughout the master's degree course and **my friends** for providing support and friendship that I needed. And I wish to express my gratitude to **my beloved parents, and my younger sister** for their endless love, invaluable support and encouragement to fulfill my wish.

Finally, I wish to express my thankful teachers from University of Computer Studies, Taunggyi. If not for the above-mentioned people, my thesis would never have been completed in such a successfull manner. I once again extend my sincere thanks to all of them.

ABSTRACT

Sign Language is the essential language for deaf and dumb people. It is a non-verbal language used by people with speech and hearing disabilities for communication. Only a few people can use sign language proficiently and most of the people do not know how to communicate the deaf people. Therefore, sign language became a boon for the physically challenged people to express their thoughts and emotion. A system that can translate is needed when a normal person wants to talk with a dumb or deaf person. Our proposed system was built to classify 11 static number signs and 30 consonant signs expect 3 dynamic signs for Myanmar Language.

In our proposed system, there are three main processes, namely, preprocessing, features extraction and classification for those extracted features. In the preprocessing stage, the input images are cropped manually for only hand regions and resized them. And then, they are converted into grayscale images. One of the feature extraction methods in image processing, namely, Kernel Principal Component Analysis (KPCA) is combined with Supportive Vector Machine (SVM) to implement a Myanmar sign language recognition system. The main concept of Kernel Principal Component Analysis is to extract non-linear features. Conventional Principal Component Analysis (PCA) is effective if data are in the form of linear structure. But it can fail to reduction data dimensions if data belong to a nonlinear low-dimensional manifold.

For classification of the extracted features, SVM is used. Its goal is to determine the best decision boundary that can separate classes with less error. Among many kernels, Gaussian Kernel Radial Basis Function (RBF) is used together with SVM to classify non-linear data in higher dimension. Data collected from 30 different people are used as dataset. As a result, KPCA with SVM have the highest accuracy (82%) compared with Principal Component Analysis.

TABLE OF CONTENTS

CHAPTER 3 THEORY BACKGROUND

CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION

CHAPTER 5 CONCLUSION AND FURTHER EXTENSIONS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In Myanmar, 673,126 of population are disable in hearing according to 2014 Myanmar national census. 4.6% of population are disable, 1.3% of the population are deaf and hearing impairment and only 0.0006% of the deaf have a university level. They describe their emotion, feelings and thoughts by using gestures or symbols that are organized in linguistic way unlike other languages. These symbols or gestures are Sign Language as a manual communication to communicate each other.

Signs can be divided into two categories: Manual Signs and Non-Manual Signs. Manual Signs (MS) depend on hand shape, position, and movement. Non-Manual Signs (NMS) can be expressed by using movement of head, tension and slack, movement of upper part of body, eyebrows, eyelid, and mouth gesture/mouthing. They are primarily used to express feeling and emotions. But manual signs are mainly used to represent alphabets, numbers, and symbols. Manual sign is also called Fingerspelling. Myanmar finger spelling characters include static and dynamic signs. In Myanmar consonants, 30 numbers of consonants (က to အ except ဇ ၊ ဈ ၊ ဏ) are static and opened finger spelling signs and the rest 3 consonant signs are dynamic signs (ဇ ၊ ဈ) and closed finger spelling signs(ဏ).

Sign language has its own grammar structure unlike in oral language. Since the last two decades, researchers had been paying attention in the area of sign language recogniton. They made many researches in computer vision, image processing, natural language process etc. deal with the area of sign language recognition. But Sign Language Recognition remains a challenging task because of the requirement of expensive devices such as sensors or 3D cameras, gloves.The proposed system aims to develop recognition system for static sign of Myanmar consonants and Myanmar numbers. The main concept of this system is to develop a good Sign Languae Recognition system with the need of less resources and less computational resources by using kernels.

1.1 Feature Extraction

Feature extraction expresses the appropriate shape of information included in a pattern to make easy the processing of classification by a formal procedure. In the image processing field, feature extraction is used for dimensionality reduction. The objectives of feature extraction are to get the most appropriate information from original dataset and to represent that information into low dimensional feature space as feature vectors. Not much information is useful for process of classification. In large and multi-dimensional dataset, such as image datasets, signal processing, etc. Only just a few data can be transformed into feature vectors for classification by using feature extraction methods. This reduction leads to build model with less computational cost and can also speed up the learning and generalization step in machine learning processes.

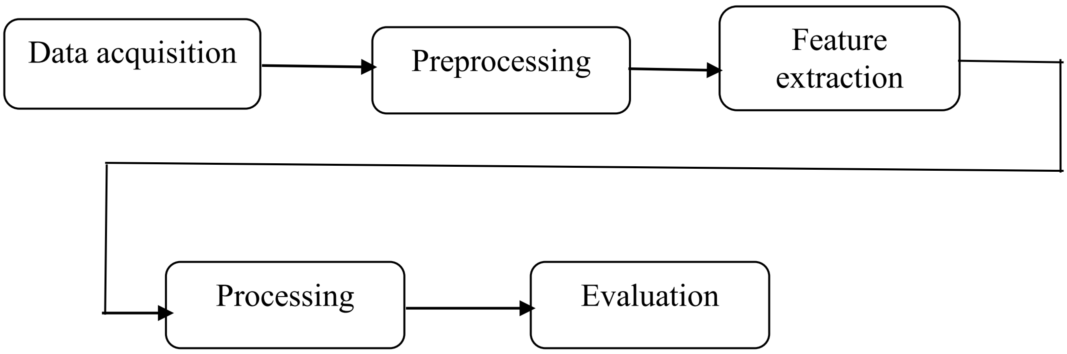


Figure 1.1 General Architecture of Feature Processing

Before feature extraction, data must be preprocessed. Data preprocessing transforms raw data that can be understood and analyzed by computers and machine learning. Preprocessing is an important part for machine learning process to develop a good machine learning model. Without preprocessing, the system will be end up with data-driven decision-making problem. A good preprocessing can make feature extraction step easier. So that much of the relevant features can extract and noise can be dropped by using feature extraction methods.

1.2 Motivation of the Thesis

Sign Language Recognition system is a recognition system based on machine learning model. Sign pictures associated with each class are trained with one of the machine learning methods. From that process, model for future prediction of other sign pictures is extracted.

There are many limitations in sign language recognition system. Some of limitations are:

- The requirements include many expensive resources to assist that system.
- The system must store many features (pixels) associated with train and test pictures.
- The need of speeding up the computation time for classification in machine learning process.
- The difficulties of accurate prediction are solved due to the noise like different dimensions.

According to the above limitations, this system was developed to produce accurate result by keeping only important features rather than maintaining all features. And the system was updated to robust with different dimensions.

1.3 Objectives of the Thesis

Sign Language Recognition System is a translation system from static images for Myanmar Consonants and Myanmar Numbers. Input picture is predicted with model created with supervised machine learning model. The objectives of thesis are as follows:

- To implement a recognition system that contains static sign gestures
- To develop machine learning model for sign language recognition system
- To study Kernel Principal Component Analysis for features extraction
- To improve accuracy of prediction results by combining feature extraction model and machine learning model
- To reduce the computing time with less feature
- To assist the deaf and dump in communication with others and among themselves

1.4 Organization of the Thesis

This thesis is organized into five chapters, abstract, acknowledgment and references.

About Myanmar Sign Language Recognition System Using Supportive Vector Machine (SVM) and Kernel Principal Component Analysis (KPCA) is introduced in Chapter (1). This chapter also includes the motivation, and objective of the research book. In Chapter (2), about the basic knowledge of image processing, Sign Language including Myanmar Sign Language, tools that are used in the proposed system and reviews of the paper related with this work are presented. Chapter (3) will explain about various types of feature extraction methods and machine learning models detailed. Proposed system design, experiments and experimental results are described in Chapter (4). The conclusion of the research work is drawn in chapter (5). In this chapter, further extensions that propose some improvements which could be made are presented. The limitations of the system are also described in this chapter.

CHAPTER 2
SIGN LANGUAGE AND LITERATURE REVIEW

This chapter presents about the image processing, sign language and tools, Myanmar Sign Language that are used in the proposed system and reviews of the paper related with this work.

2.1 Feature Extraction

Image processing is a technique for applying various procedures to an image in order to improve it or extract some relevant information from it. It is one of the types of signal processing techniques that may be the processing from image to image or from image to the features/characteristics related with that image. Today, the popularity of image processing is growing rapidly. Most researches are mainly focus in image processing area and became a vast technology. There are three steps in the field of image processing:

- Using image acquisition tools to import the image
- Examining and modifying the image
- Producing the output based on the features of the image

Image processing techniques can be divided into two categories: analog and Digital. Utilizing only two-dimensional signals, analog image processing is performed to analog signals. Pictures from television, photos, artworks, and medical imagery are some examples of analog images. Digital images can be manipulated with the help of computers. When employing digital technique, all forms of data must go through three general phases: pre-processing, improvement, and display, as well as information extraction.

2.1.1 Digital Image Processing

Digital image processing can be used to manipulate images by using digital computers. Over the past few decades, its use has grown enormously. It can be used for everything from medical to entertainment, as well as for things like remote sensing and geological analysis. There are many classes in digital image processing, namely, image enhancement, image restoration, image analysis and image compression. Image compression is a technique used to reduce the number of bytes in a graphics file without lowering the image quality below a desirable level.

Computers can be applied in the processing of digital images. Numerical representation of a digital image can be operated according to obtain a desired result. Digital image processing is one of the methods to convert from a physical image into a digital image by extracting significant information through various algorithms.

2.1.2 Pattern Recognition

Pattern Recognition is a method of separating objects from input images. These objects can be identified and classified by statistical decision theory.

To perform the process of segmentation, detected objects must be separated from other background. In the feature extraction step, objects are measured in order to extract features that only include qualitative features. And these features are organized into a group to form a feature vectors. The last stage is classification. In that process, the only decision made as a result is which category each object falls under.

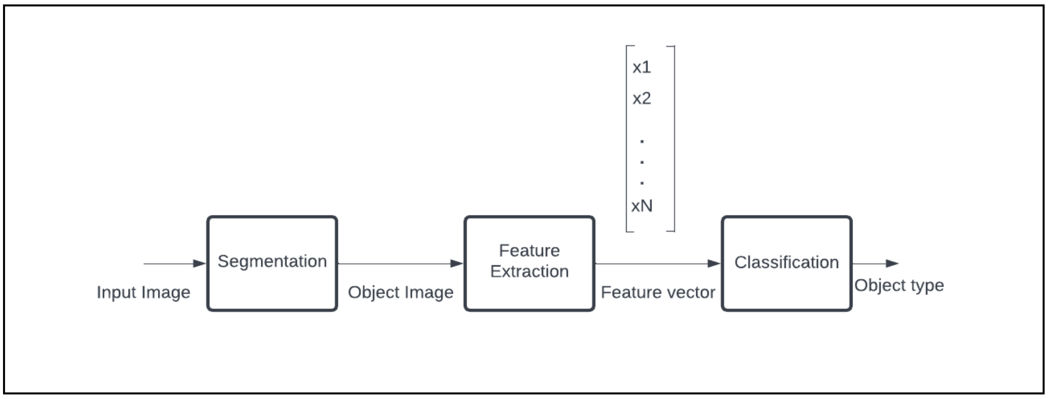


Figure 2.1. Phases of pattern recognition

2.2 Sign Language

The history of Sign Language appeared from Western societies since 17th century. Sign Language is not an oral language. It is just a visual language that can be expressed by using the hands and other parts of body. It is a special communication method for deaf and dumb people. Sign languages are different based on different countries. Some popular types of Sign Languages are: American Sign Language (ASL), Chinese Sign Language (CSL), French Sign Language (LSF), Japanese Sign Language (JSL) Syllabary, Arabic Sign Language, Spanish Sign Language (LSE), Mexican Sign Language (LSM) and Ukrainian Sign Language (USL). Users of American sign language (ASL) who use British sign language (BSL) may find it difficult to understand each other due to different usages.

Most of the deaf people cannot read and write well. Only just a few deaf people can involve in social inclusion due to the communication difficulties. To overcome these barriers, researchers make research in computer vision, image processing, natural language processing etc. deal with Sign Language translation and recognition. But Sign Language Recognition remains a challenging task in research areas.

2.2.1 American Sign Language Recognition

Most researches aiming at sign language recognition or translation are conducted on sign languages corresponding to the researcher's native language. American Sign Language (ASL) has the most published results with Chinese Sign Language (CSL) being the second most frequently researched sign language. This is major because of the popularity of the English language. This thesis work also focuses on American Sign Language Recognition. In this section, we discuss the past researches targeting American Sign Language, highlighting the dataset used, basic model used for recognition and the achieved performance.

Authors look into the potential of Kinect depth-mapping camera for sign language recognition and verification for educational games for deaf children and compared its performance against a system using colored gloves. Garcia et al. used transfer learning for sign language recognition and presented a real time finger spelling translator, utilizing GoogLeNet architecture, for American Sign Language letters. Authors tackle the problem of fingerspelling recognition of ASL alphabets "in the wild", i.e. from naturally occurring video data collected from the websites (YouTube, aslized.org and deafvideo.tv) and not from videos specially collected for recognition tasks. Attention based recurrent encoder-decoders and CTCbased approaches were explored for sequence modeling and an accuracy of 42% was achieved using a CTC-based recognizer. One of the research working with body pose and hand shape features for ASL recognition is [10]. In [11], authors use trajectories of estimated 2D skeletal data from videos and embeddings of hand images. Because of the use of skeletal data, proposed model is signer independent. The model was trained and evaluated on GMU-ASL51 dataset of 12 users and 51 ASL gestures and showed superior performance compared to baseline models.

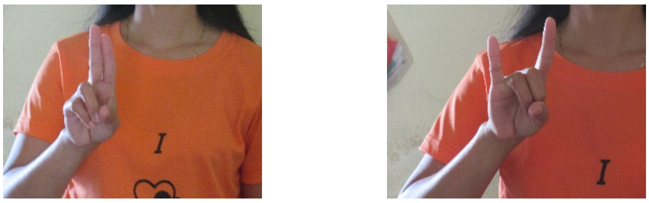
Researchers studied recognition of ASL alphabets and numerals on four publicly available ASL image datasets [12]. They propose use of a convolutional neural network (CNN) model and realize an improvement in accuracy by 9%. Another study using CNN to extract spatial features and a RNN to train on temporal features is. The study focused on a dataset created by authors comprising of videos with ASL signs were made by a single signer. Authors propose to use a YOLOv5 based solution for American Sign Language Recognition. MU_HandImages_ASL dataset was used to train and evaluate the proposed model and a precision of 95% was achieved.

2.2.2 Myanmar Sign Language (MSL)

In Myanmar, 673,126 of population are disable in hearing according to 2014 Myanmar national census. Only 0.0006% of the deaf have a university degree, whereas 1.3% of the population has hearing impairment and 4.6% of the population is disabled. There are four deaf schools in Myanmar: Immanuel School for the Deaf in Kalay, Mary Chapman School for the Deaf in Yangon (est. 1904), in Mandalay (est.1964) and School for Deaf Children in Tamwe, Yangon (est. 2014). (est.2005).

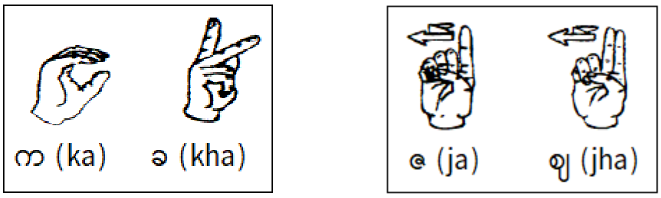
Every nation has its unique sign language, and Myanmar also has. Figure 2.2 illustrates how the upper and lower regions of Myanmar use the Myanmar Sign Language differently. The grammar structures of Myanmar Language and Myanmar Sign Language are also dissimilar.

Sign Language is mainly composed of manual and non-manual signs. Only hand regions with varying hand shape, position, and movement are employed in manual signals. But non-manual signs are expressed by using combination of movement of head, tension and slack, movement of upper part of body, eyebrows, and eyelid is used to express. Non-manual signs are used to express feeling and emotions whereas manual signs are applied to represent Myanmar and English alphabets, numbers and symbols. Static and dynamic signs are also composed in manual sign as shown in Figure 2.3.



a) ယ (ya) in Tamwe b) ယ (ya) in Mary Chapman School

Figure 2.2 Different forms of ယ (ya) between two deaf schools



(a) Static signs (b) Dynamic signs

Figure 2.3 Example of static and dynamic signs for Myanmar consonants

Data of Myanmar consonant and number used in this system are shown in Figure 2.4 and Figure 2.5 respectively.

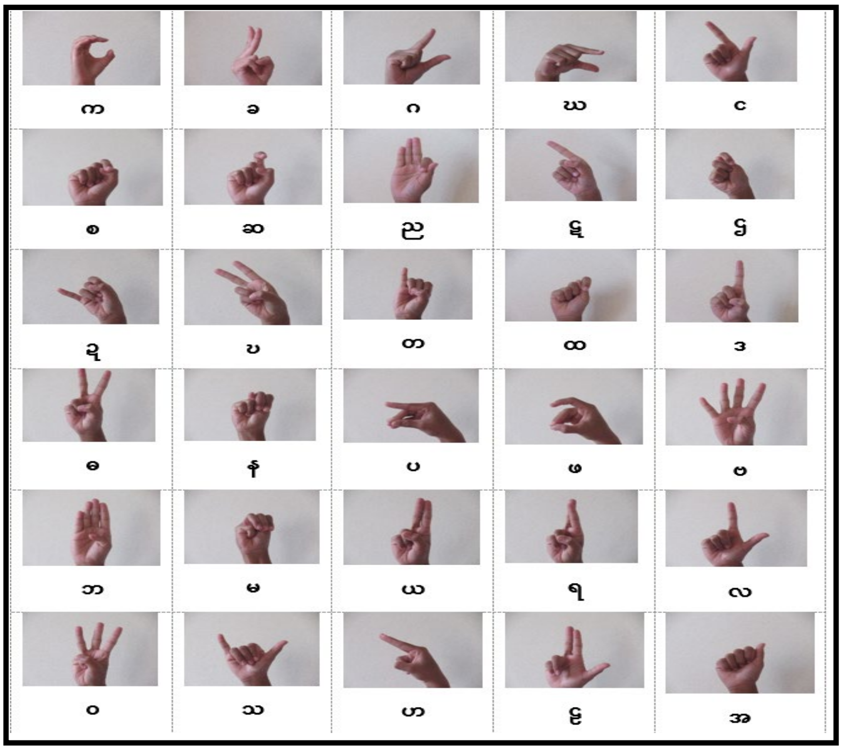


Figure 2.4 Consonant data used for the proposed system

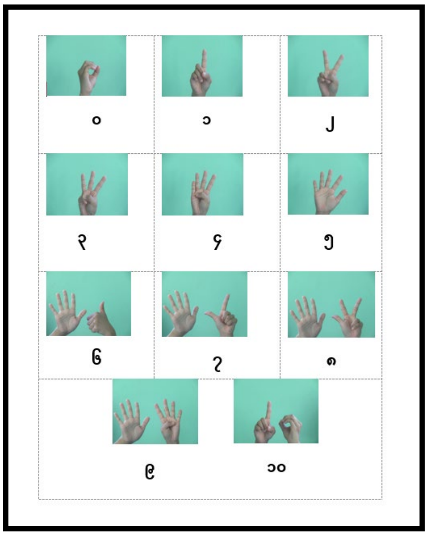


Figure 2.5 Number data used for the proposed system

2.2.3 Sign Language and Hand Gesture Recognition

Sign Language Recognition is the process of transforming from users' given sign images into the text. It fills the communication gap between the general public and those who cannot speak. Feature extraction methods are widely employed in signal processing for data dimension reduction or data decorrelation. The gesture can be mapped to appropriate text in the training data by extracting features with feature extraction methods and classifying the associate category with classifiers.

Dump persons usually face with the problem of lack access to social interaction and regular communication. Because only a few normal people can use sign language proficiently and most of the people do not know how to communicate with the deaf people. Some hearing impairment people cannot communicate with normal people by using oral language. So, they have to depend on visual communication methods. Therefore, Sign Language became a main communication form in the environment of deaf and dump.

In today's technological age, the community of the dumb is in great need of a computer-based system. Although interesting technologies are being developed for speech recognition, the industry currently lacks any genuine commercial products for sign recognition. The goal is to create user-friendly human computer interfaces and enable computers to comprehend language (HCI). Making a machine comprehend speech, human gestures, and facial expressions are some steps to reach that goal. The nonverbal communication that is done through gestures. A person can make countless gestures at once. So, that topic became the most interested point for computer vision researchers since people interpret gestures through vision. This proposed system aims to recognize human gestures by using feature extraction and classification algorithms.

2.3 Image Processing Tools

In the current digital era, data is the most precious resource that organizations have, and a sizable amount of this data is made up of photos. Deep insights for a business can be gained by data scientists by processing these photo and feeding them into machine learning (ML) models. Data scientists must use image processing techniques for machine learning and deep learning tasks in order to process this vast volume of data rapidly and effectively.

2.3.1 OpenCV

In 2000, Intel created and made available the open-source library is known as OpenCV. Open Source Computer Vision Library (OpenCV) is a type of library file that can be used for programming functions. Its role is to develop real-time computer vision. [1] Under the terms of the open-source BSD license, this library is free to use and cross-platform. Some functions of the OpenCV include features that can support machine learning processes. They are:

- Boosting
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- KNN algorithm
- Artificial neural networks
- Random Forest
- Support Vector Machine (SVM)
- Deep Neural Networks (DNN)

2.3.2 NumPy

Numpy is a type of library in Python programming language. It can support high-level mathematical functions to process large, multi-dimensional arrays and metrics. Jim Hugunin originally developed Numeric, the predecessor to NumPy, with assistance from a number of other programmers. By merging components of the rival

Numarray into Numeric and making significant changes, Travis Oliphant created NumPy in 2005.

NumPy targets the non-optimizing CPython bytecode interpreter, which is the Python reference implementation. Numpy can solve the problem of slow processing by providing array with multi dimension, operators and functions to work effectively on arrays. Since both Python and MATLAB are interpreted languages, NumPy may be used to create programs with capabilities similar to that of MATLAB. Both languages also enable users to create quick programs as long as the majority of operations are performed on arrays or matrices rather than scalars. Simulink in MATLAB is outclassed by Numpy as a more advanced and comprehensive programming language. NumPy arrays are used to store and manipulate data in the Python bindings of the widely known computer vision library OpenCV. The workflow of programming and debugging is greatly streamlined by the use of the NumPy array as the default data structure in OpenCV for images, extracted feature points, filter kernels, and many other applications.

Some of NumPy's primary highlights include:

- Tiny data storage
- Processing arrays quickly
- Supporting numerous functionalities
- Interoperability of the data with other libraries

2.3.3 Pillow/PIL

Advanced version of PIL (Python Image Library) is a Pillow. It is also an open-source library that can be used in the tasks of image processing. By using the libraries of Pillow, many works deal with field of image processing such as point operations, filtering and manipulation can be done.

The main functions of the Pillow are:

- Support for a variety of image file types, including JPEG and PNG
- Easy to use
- Provide many image processing methods
- Perform augmenting training data for problems of computer vision

2.4 Literature Review

In [5], there has been extensive research on the Sign Language Recognition (SLR) system, which is necessary to recognize sign languages. This research relied on a range of input sensors, gesture segmentation, feature extraction, and classification techniques. This paper analyzed and compared classification methods included in Sign Language Recognition System. And it offered the approach that will be most effective for further study. About the Hybrid methods and Deep Learning for advanced classification were contributed. According to our review, earlier research has extensively examined HMM-based techniques, including its modifications. Hybrid CNN-HMM and fully Deep Learning approaches gave the best results and opportunities for further exploration.

In [6], Chat programs have developed into a potent medium that helps individuals communicate with one another in various languages. Although there are a lots of chat applications for use to communicate between different people in different languages, there are no such application to chat to facilitate the communication with sign languages. This system was created based on Sinhala Sign. This system includes the processes of converting text messages, voice messages into sign messages and vice versa. The development of speech character recognition for voice communications has made use of the Google voice recognition API. This system was trained by using some text parameters for speech and text patterns and output is displayed by emoji that has the same meaning with Sinhala Sign Language. This technology allows for two-way communication; however, it relies on a pattern of gesture recognition that makes it difficult to obtain the intended results.

In Intelligent Sign Language Recognition Using Image Processing [7], an effective and fast algorithm is provided to determine how many fingers are extended in a gesture that represents an alphabet in Binary Sign Language. Perfect alignment of hand to the camera was not required for that system. The project developed an identification function that can identify English alphabetic sign language by using image processing techniques. The main goal of this project is to create an intelligent computer-based system that will dramatically improve dumb people's ability to communicate with everyone else. The ideas behind image processing, machine learning and artificial intelligence were used to design and build up an intelligent system in order to accept visual inputs and to generate recognizable form of outputs.

This research of Hand Gesture Recognition based on Digital Image Processing using MATLAB [8] developed a system to recognize hand gesture for normal people and to communicate with dumb people effectively. This work mainly focused on hand gesture recognition to provide real time communication system. This system solved the real time problem by using the techniques in Digital Image Processing such as Color Segmentation, Skin Detection, Image Segmentation, Image Filtering and Template Matching. The American Sign Language (ASL) alphabet and a small subset of its words are all recognized by this system as ASL gestures.

Ni Htwe Aung, Su Su Maung, Ye Kyaw Thu [1] developed a sign language recognition system for Myanmar numbers from zero (၀) to ten (၁၀). They also investigated the performance of three different Supportive Vector Machines (SVM) classifiers: SVC with polynomial kernel, SVC with liner kernel and LinearSVC. In the preprocessing stage, recorded videos are transformed into multiple frames. That converted frames are cropped for only hand regions and resized into 128x128 resolution. And then those images are changed into grayscale format. In the feature extraction stage, distinctive invariant features are extracted from converted grayscale image data by using Scale Invariant Feature Transform (SIFT). In the last stage, classification stage, extracted features are fed into three different classifiers for recognition of hand signs. SVM with polynomial kernel achieved the accuracy score of 88%.

[2] Face Recognition between Two Person using Kernel Principal Component Analysis and Support Vector Machines are developed by Ivanna K. Timotius, Iwan Setyawan, and Andreas A. Febrianto compared the performance by using Kernel Principal Component Analysis and Support Vector Machine with a pair of Kernel Principal Component Analysis and Nearest Neighbor classifier. Image data are first transformed into feature space by using kernel function to reduce dimension. Features obtained from feature extraction stage are fed into SVM to classify. Finally, KPCA with SVM had 99.05% of accuracy and KPCA with NN had 97.14% in accuracy.

[3] Myint Tun, Thida Lwin developed Real-time Myanmar Sign Language Recognition System using PCA and SVM. In that system, input video stream is detected by Viola-Jones Object Detection Framework and converted into YCbCr color space. After threshold segmentation, they made removing background from the received binary image and marking hand region for feature extraction. For feature

extraction, Principal Component Analysis (PCA) is used to extract maximum variance feature by reducing dimension. Extracted features are fed into Support Vector Machine to recognize hand signs. Experimental results showed that the system gave the successful recognition accuracy of static sign gestures of MSL alphabets with 89%

CHAPTER 3
THEORY BACKGROUND

3.1 Dimensionality Extraction

The idea behind machine learning is that the more features have, the better the forecast will be, however this is not always the case. A new set of features is created from the original feature set using a family of dimensionality reduction techniques called feature extraction. The new features are less in number than the original ones in order to decrease dimensionality. Feature extraction is a tool that reduces and simplifies the original features.

3.1.1 Cure of Dimensionality

High-dimensional data are frequently include in machine learning. If the work with 50x50 grayscale images, the workspace has 2,500 dimensions. The dimensionality rises to 7,500 dimensions if the images are RGB-colored (one dimension for each color channel in each pixel in the image).

There are two options to reduce dimensionality:

1. Feature elimination: directly eliminating some features.
2. Feature extraction: preserving the majority of the features that matter most.

Types of feature extraction techniques are: LDA (Linear Discriminant Analysis) and PCA (Principal Component Analysis). The LDA concept is really straightforward. Given the training sample set, LDA attempts to project the sample onto a straight line with the projection points of the interclass samples being as near together as feasible and the interclass samples being as far away. We projected a fresh sample onto the same line and classified it. Based on the location of the projected point, this sample's classification is established. LDA aims to increase the data points' ability to be distinguished following dimension reduction, in contrast to PCA, which aims to preserve data as much as possible.

3.1.2 Principal Component Analysis (PCA)

A statistical technique called principle component analysis (PCA) transforms a series of observations of potentially correlated variables into a set of linearly uncorrelated variable. It is a basis transformation technique to diagonalize an estimate of the covariance matrix of the data x_k, k = 1,…,l, $x_k x_k \in R^N$, $\sum_{k=1}^{l} x_k$=0, defined as

$$C = \frac{1}{l}\sum_{k=1}^{l} x_j x_j^T \tag{3.1}$$

The new coordinates in the Eigenvector basis, i.e. Principal components are the orthogonal projections onto the Eigenvectors. Suppose we first map the data nonlinearly into a feature space *F* by

$$\Phi :\ \mathrm{R^N} \quad F, \mathrm{x} \mapsto \mathrm{X} \tag{3.2}$$

We'll demonstrate that, for particular selections of, we can still do PCA in F even if F has arbitrarily huge dimensions. This can be done by using the kernel functions known from Support Vector Machines (Boser, Guyon, & Vapnik, 1992).

3.1.3 Kernel PCA

Assume for the moment that our data mapped into feature space, $\Phi(x_1)$,…, $\Phi(x_l)$, is centered, i.e. $\sum_{k=1}^{l} \Phi(x_k) = 0$. To do PCA for the covariance matrix

$$C = \frac{1}{l}\sum_{j=1}^{l} \Phi\left(x_j\right) \Phi\left(x_j\right)^{\mathrm{T}}, \tag{3.3}$$

We have to find Eigenvalues λ ≥ 0 and Eigenvectors V ∈ *F*\{0} satisfying λV = CV. Substituting (3.3), we note that all solutions V lie in the span of $\Phi(x_1)$,…, $\Phi(x_l)$. This implies that we may consider the equivalent system

$$\lambda(\Phi(x_k).V) = \left(\Phi\left(x_j\right).CV\right) \text{ for all k =1,…., } l, \tag{3.4}$$

and that there exist coefficients α_k,…, α_l such that

$$V = \sum_{i=1}^{l} \alpha_i \Phi(x_i). \tag{3.5}$$

Substituting (3.3) and (3.5) into (3.4), and defining an l x l matrix *K* by

$$K_{ij} := (\Phi(x_i).\Phi\left(x_j\right)), \tag{3.6}$$

we arrive at

$$l\lambda K\alpha = K^2\alpha, \tag{3.7}$$

where α denotes the column vector with entries α_1,…, α_l. To find solutions of (3.7), we solve the Eigenvalue problem

$$l\lambda\alpha = K\alpha, \tag{3.8}$$

for nonzero Eigenvalues. Clearly, all solutions of (3.8) do satisfy (3.7). Moreover, it can be shown that any additional solutions of (3.8) do not make a difference in the expansion (3.5) and thus are not interesting for us.

We normalize the solutions α^kbelonging to nonzero Eigenvalues by requiring that the corresponding vectors in *F* be normalized, i.e. $(\mathrm{V}^k.\mathrm{V}^k) = 1$. By virtue of (5), (3.6) and (3.8), this translates into

$$1 = \sum_{i,j=1}^{l} \alpha_i^k \alpha_j^k (\Phi(x_i).\Phi(x_j)) = (\alpha^k . K\alpha^k) = \lambda_k(\alpha^k.\alpha^k). \tag{3.9}$$

For principal component extraction, we compute projections of the image of a test point $\phi(x)$ onto the Eigenvectors V^k in *F* according to

$$(\mathrm{V}^k.\phi(x)) = \sum_{i=1}^{l} \alpha_i^k (\Phi(x_i).\phi(x)). \tag{3.10}$$

Note that neither (3.6) nor (3.10) requires the $\Phi(x_i)$ in explicit form | they are only needed in dot products. Therefore, we are able to use kernel functions for computing these dot products without actually performing the map Φ (Aizerman, Braverman, & Rozonoer, 1964; Boser, Guyon, & Vapnik, 1992): for some choices of a kernel $k(x,y)$, it can be shown by methods of functional analysis that there exists a map Φ into some dot product space *F* (possibly of infinite dimension) such that *k* computes the dot product in *F*. Kernels which have successfully been used in Support Vector Machines (Scholkopf, Burges, & Vapnik, 1995) include polynomial kernels

$$k(x,y) = (x,y)^d, \tag{3.11}$$

radial basis functions $k(x,y) = \exp(-\|x-y\|^2 / (2\sigma^2))$, and sigmod kernels $k(x,y)$ = tanh($k(x.y)$+Θ).). It can be shown that polynomial kernels of degree d correspond to a map Φ into a feature space which is spanned by all products of *d* entries of an input pattern, e.g., for the case of N = 2, d=2

$$(x.y)^2 = (x_1^2, x_1x_2, x_2x_1, x_2^2), (y_1^2, y_1y_2, y_2y_1, y_2^2) \tag{3.12}$$

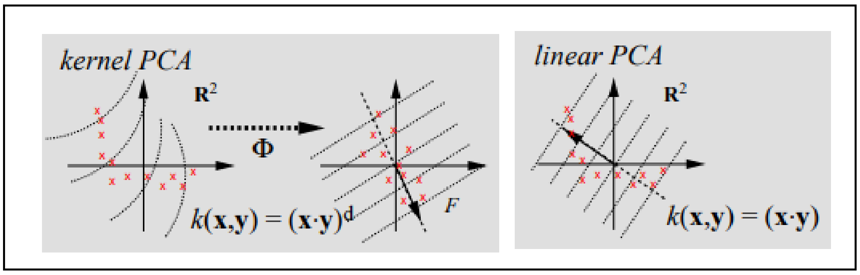


Figure 3.1 Basic idea of kernel PCA

If the patterns are images, we can thus work in the space of all products of d pixels and thereby take into account higher order statistics when doing PCA. Substituting kernel functions for all occurrences of $(\phi(x).\phi(y))$, we obtain the following algorithm for kernel PCA (Fig. 3.1): we compute the dot product matrix (cf. Eq. (3.6)) $K_{ij} := (k(x_i,x_j))_{ij}$solve (3.8) by diagonalizing K, normalize the Eigenvector expansion coefficients α^n by requiring Eq. (3.9), and extract principal components (corresponding to the kernel k) of a test point x by computing projections onto Eigenvectors (Eq. (3.10), Fig. 2).

We should point out that in practice, our algorithm is not equivalent to the form of nonlinear PCA obtainable by explicitly mapping into the feature space F: even though the rank of the dot product matrix will be limited by the sample size, we may not even be able to compute this matrix, if the dimensionality is prohibitively high. For instance, 16 x 16 pixel input images and a polynomial degree d = 5 yield a dimensionality of 100. Kernel PCA deals with this problem by automatically choosing a subspace of F (with a dimensionality given by the rank of K), and by providing a means of computing dot products between vectors in this subspace. This way, we have to evaluate kernel functions in input space rather than a dot product in a 1010{dimensional space.

To conclude this section, we brie y mention the case where we drop the assumption that the $\Phi(x_i)$ are centered in F . Note that we cannot in general center the data, as we cannot compute the mean of a set of points that we do not have in explicit form. Instead, we have to go through the above algebra using $\Phi(x_i) \coloneqq \Phi(x_i) - (\frac{1}{l}) \sum_{i=1}^{l} \Phi(x_i)$. It turns out that the matrix that we have to diagonalize in that case, call it *K* can be expressed in terms of K as $K_{ij} = K - 1_i K - K 1_l + 1_l$K

3.2 Support Vector Machine

Machine learning is regarded as a branch of artificial intelligence that deals with the creation of tools and processes that let computers learn. Creating algorithms that allow machines to learn and carry out jobs and activities is a simple way to put it. There are numerous areas where machine learning and statistics intersect. Numerous methods and strategies for machine learning problems have been developed over time. Boser, Guyon, and Vapnik's COLT-92 paper that introduced the Support Vector Machine (SVM) was published in 1992. Support vector machines (SVMs) are a group of similar supervised learning techniques used for classification and regression. They are a member of the generalized linear classifier family. Support Vector Machine (SVM), in other words, is a classification and regression prediction tool that automatically detects over-fitting to the data while maximizing predictive accuracy using machine learning theory. Support Vector machines are systems that use the hypothesis space of a linear function in a high-dimensional feature space and are trained using an optimization theory-based learning method that incorporates a learning bias. Support vector machines were first well-liked by the NIPS community and are now an important component of machine learning research all across the world. SVM gains notoriety when used with pixel maps as input; In a challenge requiring handwriting identification, it provides accuracy on par with well-developed neural networks with extensive features. It plays an important role in many applications, such as handwriting analysis, face analysis and so on. Vapnik created the foundations of Support Vector Machines (SVM), which have grown in prominence thanks to a number of promising features like improved empirical performance. The Structural Risk Minimization (SRM) concept, which has been demonstrated to be superior to the Empirical Risk Minimization (ERM) principle utilized by standard neural networks, is used in the formulation. While ERM minimizes the inaccuracy on the training data, SRM minimizes an upper bound on the estimated risk. This distinction gives SVM a better chance of generalizing, which is what statistical learning aims for. SVMs were initially created to address the classification issue, but more recently they have also been expanded to address regression issues.

Support Vector Machine (SVM) is a discriminative classifier with high generalization properties that learns the decision surface through a discrimination process. SVM has been shown to be an effective classifier for a variety of traditional pattern recognition issues, including text categorization, image recognition, image classification, objects recognition, cancer classification, spam categorization, face recognition, motion detection, face detection, electricity fraud prediction, electricity load forecasting, signature verification, time series prediction, system identification, web document classification, stock market forecasting, speech recognition and speaker verification.

SVM is a relatively new addition to the numerous classification techniques, appearing in the 1990s. It uses a two-class categorization approach in its simplest form. One new application is as a neural network substitute, and it has received extensive investigation and use. The key benefit of SVM over neural networks is that it offers a strong theoretical framework for considering both structural behavior and experimental data in order to create an effective classifier. This allows for superior generalization ability (Scholkopf, 1999).

In most circumstances, performance of SVM generalization either matches or significantly outperforms that of rival approaches. Structural risk minimization, which underpins SVM's superior generalization performance (SRM) (Vapnik, 1998). Its formulation maximizes the class separation margin to come close to the SRM principle. Thus, Large margin classifier is another name for the SVM classifier. For linearly separable datasets, the basic SVM formulation can be used. It may be applied to nonlinear datasets with a minor adjustment by employing kernel functions to proximate the greatest margin decision function in a linear feature space by indirect mapping the nonlinear input space to a linear feature space (Burges, 1998). Soft-margin SVM formulation is used to address outliers. Non-linear soft-margin SVM is the generic SVM formulation, with linear and hard margin (non-separable) issues being specific examples.

Convex quadratic programming problems with equality and inequality constraints are solved as part of SVM training to approximate the SRM. In the formulation's final solution, nonzero parameters are solved for, and a subset of training data is extracted that corresponds to the parameter. It can be solved rather quickly and executed on a properly configured PC for training on tiny datasets, such as those with less than 1000 samples. Large computational power and memory are needed to solve the quadratic function for large datasets in order to save the kernel matrix throughout calculation. The quantity of training datasets is inversely correlated with memory requirements.

Over the years, a variety of SVM training techniques have been created to address the memory requirement issue, shorten training times, and discover the optimum training model by employing the right kernel and hyper parameters (Burges, 1998). Additionally, because basic SVM can only handle two-class classification, in order to create a multiclass classifier, at the very least numerous two-class classifiers must be trained. Voting methods are also employed in classification to choose the appropriate class (Weston, 1998) (Hastie, 1996) (Hsu, 2002). Although a method of converting the two class SVM formulation into a single multiclass formulation has been proposed, it has not yet been extensively adopted.

The accessibility of solid implementation packages has increased the popularity of SVM. Several implementation packages are publicly accessible and have had widespread use, according to several researchers.

3.3 Theoretical foundation

The concept of support vector machines was developed by Vapnik within the context of statistical learning theory (Vapnik, 1998). Some basic ideas of the theory are shortly discussed first.

3.3.1 Statistical Learning Theory

The classification issue in supervised learning is presented in statistical learning theory (SLT) as follows (Vapnik, 1999):

We are given a set of l training data and its class, $\{(x_1,\ y_1), (x_2,\ y_2), \ldots\ldots, (x_n, y_n)\}$ in $R^n \ \times \ R$ sampled according to unknown joint probability distribution $P(x, y)$ characterizing how the classes are spread in $R^n \ \times \ R$.

We assume that the training data were produced independently and at random based on the joint distribution. A learning machine's objective is to learn $y = f(x)$. We have two options for discovering the unknown mapping:

(a) Calculate a function that, according to the chosen measure, is "near" to the joint distribution.

(b) Discover the best predictor or classifier for the output of the system.

We cannot estimate a reliable predictor of the output in the first scenario. Estimating the joint probability distribution is the aim. However, we can genuinely pursue the objective of discovering the best predictor or classifier for data classification. The process of learning then involves selecting a function from a collection of functions established by the design of the learning machine. The network layout for a gradient-based neural network classifier is predefined, limiting our options to a small number of functions. This is done by calculating the weights of the connections in the predefined network. Some optimality criterion that assesses the caliber or effectiveness of the learning machine is used to determine the best network for the classifier. By lowering the structural risk, SLT enables us to learn the best classifier.

3.4 SVM Formulation

The decision hyper plane in the linear formulation of SVM is located in the space of the input data x. In this case the hypothesis space is a subset of all hyper planes of the form: f(x) = w·x +b. The learning problem is solved by SVM using an ideal hyper plane that is geometrically remote from all classes since it will generalize the best to yet-unknown data in the future.

The best decision hyperplane can be discovered in one of two ways. The first method involves increasing the space between two supporting planes. The second method entails identifying a plane that divides the two convex hulls defined by the collection of points for each class at their two closest points. Both methods will produce the same optimal decision plane and the same set of points that support the solution. These are referred to as support vectors.

3.4.1 Linearly Separable Case

Let's consider SVM formulation for linearly separable case using the method of maximizing margin as outlined in Figure 3.2. For a set of l linearly separable data a $\{(x_1,\ y_1),(x_2,\ y_2),\ldots\ldots,(x_n,\ y_n)\}$where $x_i \in R^d$and $y_i \in \{\ \pm1\}$ we would like to learn a linear separating hyper plane classifier f(x) = w.x + b that has the maximum separating margin with respect to the two classes where w is the normal of the hyperplane.

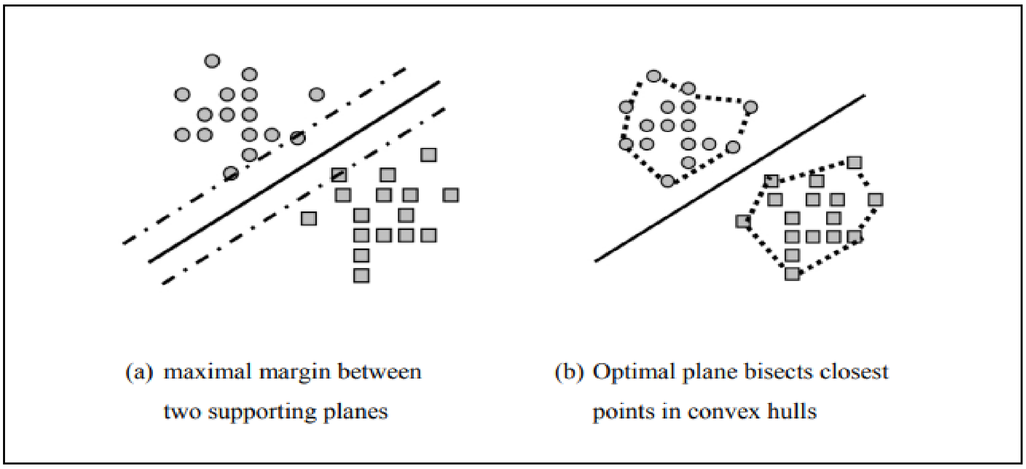


Figure 3.2 Finding the optimal decision hyperplane

Specifically, we're looking for the hyperplane: H: y = w.x + b = 0 and two hyper planes parallel to it and with equal distances to it,

$$H_1: y = w.x + b = +1 \qquad and$$
$$H_2: y = w.x + b = -1$$

only if there are no data points between H1 and H2, and the distance or margin M between H1 and H2 is maximized. Figure 3.3 show the hyper planes in the case of input data x with two dimensions.

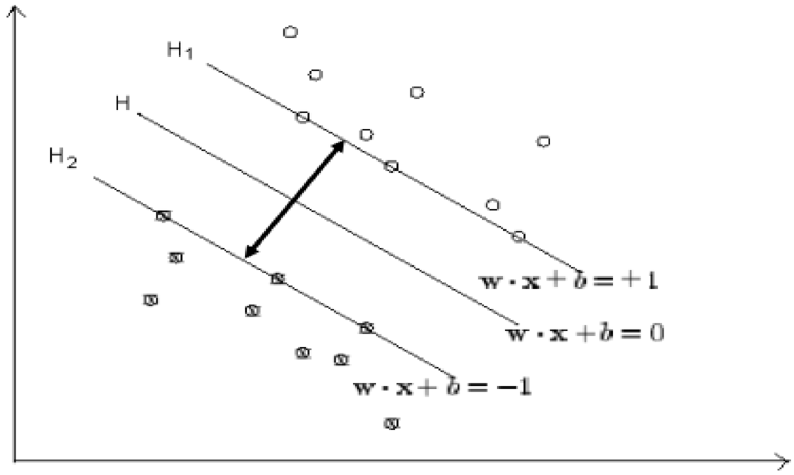


Figure 3.3 Maximal Margin hyperplanes for two dimension examples

We can always normalize the coefficients vector w for every H, H1 and H2 to ensure that:

$$H_1 be\ y = w.x + b = +1 \qquad and$$
$$H_2 be\ y = w.x + b = -1$$

Our goal is to keep *H1* and *H2* as far apart as possible. As a result, some examples on *H1* will be positive, and some examples on *H2* will be negative. These are the support vectors. The distance between *H1* to H is $\frac{|wx+b|}{\|w\|} = \frac{1}{\|w\|}$ and thus between *H1* and *H2* is $\frac{2}{\|w\|}$. Therefore, to maximize the margin, we need to minimize $\|w\| = w^T w$ with the condition that no data points lies between *H1* and *H2*. This is satisfied when:

$$w.x + b \geq +1 \; for\, y_i = +1,$$
$$w.x + b \geq -1 \; for\, y_i = -1,$$

Combining the two conditions, we have: $y_i\,(w.x + b) \geq 1$

For simplicity, the problem can be formulated as:

$$\min_{w,b} \frac{1}{2} w^T w,$$

Subject to $\quad y_i(w.x + b) \geq 1$

This can be solved by introducing Lagrange multipliers α_1, α_2..., $\alpha_l \geq 0$, for every training data (Klein, 2000). See appendix B for a discussion on Lagrange multipliers.

Thus, we have the following Lagrangian:

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=l}^{l} \alpha_i \, y_i(wx_i + b) + \sum_{i=l}^{l} \alpha_i$$

The optimization problem is referred to as having a primal formulation, or L_p, for short. The objective function is the first term on the RHS and is defined as half the square of the norm. The optimization constraints are the other two terms. This is a problem of convex quadratic (because the objective function itself is convex). We have to maximize L_p with respect to α, subject to the constraint that the gradient of L_p with respect to the primal variables w and b should be 0:

$i.e$: $\frac{\partial L_p}{\partial w} = 0$ and $\frac{\partial L_p}{\partial b} = 0$ and that $\alpha \geq 0$.

Finding the gradient and solving for 0, we then have:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i$$

And

$$\sum_{i=1}^{l} \alpha_i y_i = 0$$

Substituting them into L_p, we have the Lagrangian dual L_D where:

$$L_D = \sum_{i=1}^{l} \alpha_i y_i x_i . \sum_j \alpha_j y_j x_j - \sum_j \alpha_i y_i \left(x_i . \sum_j \alpha_j y_j x_j + b \right) + \sum_i \alpha_i$$

$$= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j$$

Observe that the primal variables w and b are eliminated.

Solving for α_i, using L_D constitute SVM learning. In order to obtain the value of w we substitute α_i into the formula $w = \sum_{i=1}^{l} \alpha_i y_i x_i$. The value of b can be averaged from the values of y − wx for each x in the training set, after w is obtained. Thus, we obtained the decision function as:

$$f(x) = sgn(\sum_{i=1}^{l} \alpha_i y_i (x_i . x) + b)$$

where the sign (sgn) is used to classify examples as either in-class or out-of-class.

In other words, the SVM classifier is defined by the equation above. Take note of the fact that the classifier is specified in terms of the training examples. However, not all training instances are used in the classifier's definition. The Support Vectors, which are training instances with non-zero multipliers, by themselves define the classifier. The complexity of the classifier can also be determined by the amount of the dataset. In straightforward classification problems, there are typically few support vectors and vice versa. The complexity of the classifier scales linearly with the number of support vectors, because since there are M dot products involved in the definition of the classifier, where M is the number of support vectors.

3.4.2 Linear Soft Margin and Non-Linear SVM

Real-world data is typically not linearly separable due to nonlinearities or noise. There is no requirement that there be no data points between the planes *H1* and *H2* described in the previous section when the input space is imperfectly separable, and noise in the input data is taken into account instead. Instead, penalty C is imposed if data points cross the boundaries. So, the problem can be formulated as:

$$\min_{w,\xi,b} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

where C is the penalty term, subject to the condition $y_i(wx + b) \geq 1 - \xi_i$

Using similar formulation as in linear case, we obtained the same dual Lagrangian but with a different constraint for α_i, which is bounded above by C (ie: 0< α_i < C). For non-linearly separable input, they can be mapped to higher dimensional feature space as mentioned earlier. If the mapping function is $\Phi(.)$, we just solve:

$$\max L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i)\, \Phi(x_j)$$

Generally, if the dot product $\Phi(x_i).\ \Phi(x_j)$ is equivalent to a kernel $K(x_i,\ x_j)$, the mapping need not be done explicitly. Thus, equation above can be replaced by:

$$\max L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Performing the map into feature space and then applying the dot product in that space is equal to using the kernel in input space. That can be done with several kernels as long as they meet Mercer's requirement. Table 3.1 gives a number of commonly used kernels.

Table 3.1 Commonly used Kernels for SVM

Kernel type	Equation
Linear kernel	$K(x, y) = x.y$
Polynomial kernel	$K(x, y) = (x.y + 1)^d$
Radial basis function (Gaussian) kernel	$K(x\ y) = e^{\frac{-\lvert x-y \rvert^2}{2\sigma^2}}$
Hyperbolic tangent kernel	$K(x, y) = tanh(ax.y - b)$

Beside the above kernels, user defined kernels can also be used as long as they satisfy Mercer's condition (Joachims, 1999). (Burges, 1998) also gives a good description of Mercer's condition.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

This chapter describes the design of the system, implementation, and experimental results in details. The system is developed with Python programming language, and image data are stored in .jpg format. The algorithm is developed on the own dataset that was gathered from 30 different people.

4.1 Design of the System

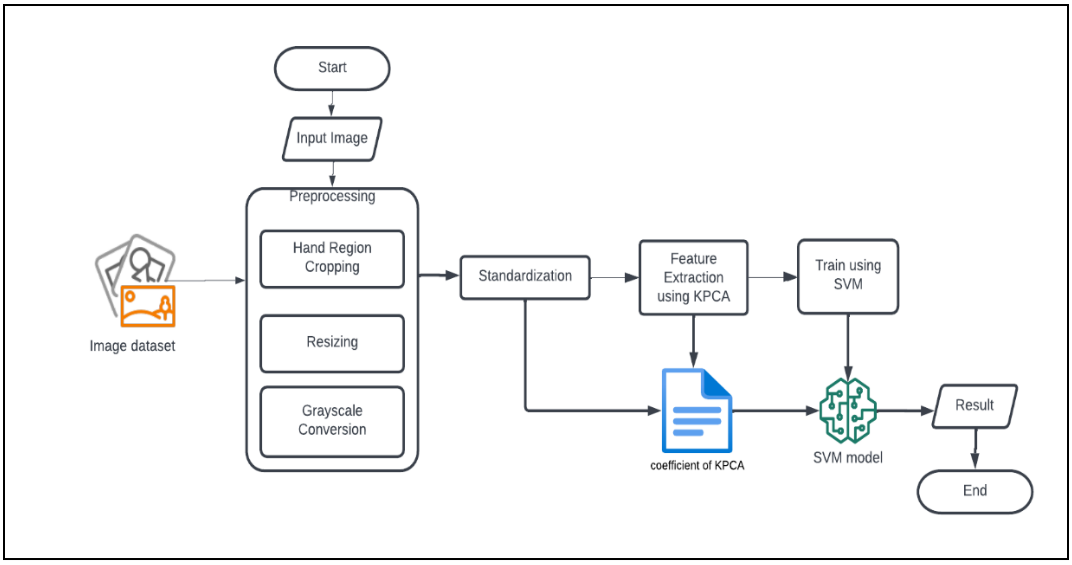


Figure 4.1 Process flow of the proposed system

The proposed system will categorize image data from the created dataset. In the sign langue dataset, there are 41 classes that represent 30 static consonant sign images and 11 static number sign images. The dataset is only focus on Myanmar Sign Language.

For both training and testing stages, sign images are collected by using Canon PowerShot SX620 HS with the resolution of 5184x388 and they are cropped manually for only hand regions. And they are saved into the respective folders for the training phase. In the process of the training, static sign images are read form the directory that stored image files. There are three processes in the preprocessing stage, namely, cropping hand region, resizing cropped images into 128x128 resolution, and converting resized images into binary images (grayscale images). Then gray images are standardized to reduce all the characteristics to a similar scale without distorting the variations in the range of the values. Feature vectors are extracted by using Kernel Principal Component Analysis (KPCA) and coefficient of KPCA is stored as a file. Reduced features are fed into classification process. The dataset is divided into the ratio of 80:20 to test performance by using Supportive Vector Machine (SVM) with radial basis function (RBF). Before the model is created, GridSearchCV performs hypermeter tuning. And the model is created with the best parameter values of SVM. And it is saved for further prediction process.

In the testing stage, input image is preprocessed like processes in training phase. And feature vectors are extracted from converted gray input image by using the saved coefficient of KPCA. Then extracted features are tested with the SVM model that was saved before.

4.2 Implementation of the system

From the stage of data collection to the measurement of system performance, this section explains how to design a sign language recognition system using a machine learning classifier.

4.2.1 Data Collection

For the implementation of this system, data are collected according to the format of dictionary book published by Department of Social Welfare, Ministry of Social Welfare, Relief and Resettlement. Although there are 34 numbers of Myanmar consonants, two dynamic signs that represent ဇ(ja) ဈ(jha) and one closed finger spelling consonant sign ဏ(nna) are discarded for developing recognition system of static signs. So, there are 41 classes in our dataset by combining 11 numbers of Myanmar number sign images and 30 numbers of Myanmar consonant sign images. Data are collected from different 30 people with different backgrounds. Dataset about Myanmar consonants and Myanmar numbers are shown in figure 4.2 and figure 4.3 respectively.

Figure 4.2 Static sign images for 11 Myanmar number from ၀ (zero) to ၁၀ (ten)

Figure 4.3 Static sign images for 30 Myanmar consonants from က (ka) to အ (a)

4.2.2 Preparation for Training Dataset

First, only hand regions are cropped from the gathered data. And OpenCV library is used to read data of image. After reading data of the image, images are resized into 128x128 format. And then Color channel is converted from RGB to Grayscale. OpenCV library is used for preprocessing step. Preprocessed gray images are written into the specific folder to change dimension. There are three actions used to change the dimension of the train images, namely, flipping, straightening with different degrees, and adding color (white and black). The total number of images are 150 for each class after changing dimensions. After the preprocessing step, feature values of each image are read and flatten into one dimensional array. Two array files are created. One file is created to store data of image in one dimensional array. Another file is the target file that includes labels for each image in data file. These two array files are created by using NumPy library and saved them as (.npy) files. Figure 4.4 and figure 4.5 illustrates about the data with different dimensions.

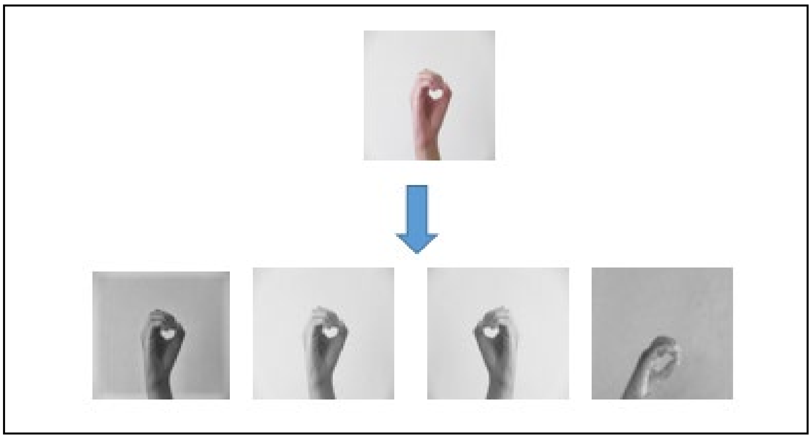


Figure 4.4 Image with different dimensions that represents ၀ (zero)

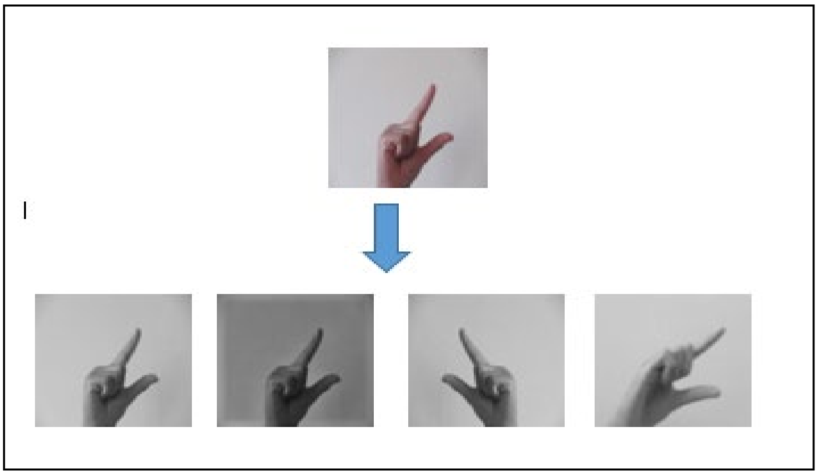


Figure 4.5 Image with different dimensions that represents ဂ (ga)

4.2.3 User Interfaces Design of the System

The user interfaces of "Myanmar Sign Language Recognition System" are in this section. User interface of this system is developed with "tkinter" library. Figure 4.6 represents the main page of the system. The functions for uploading sign images, predicting and clearing are tagged with associate buttons.

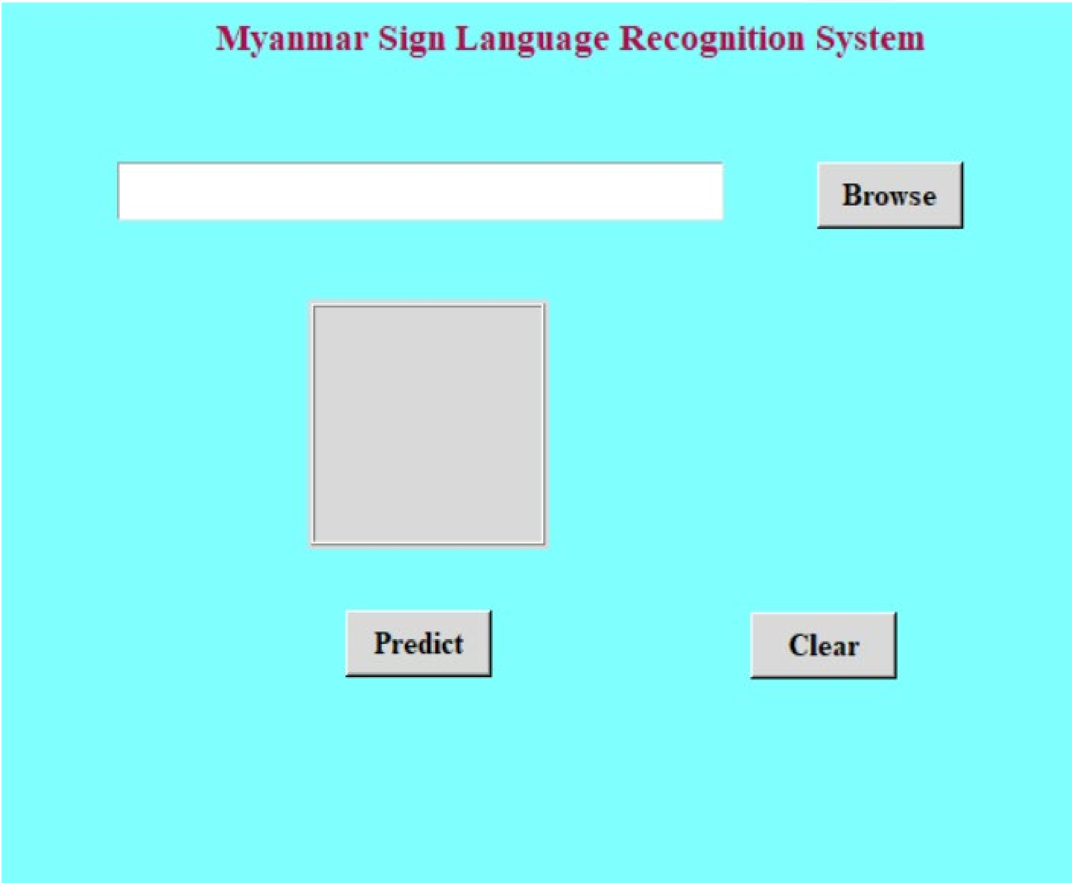


Figure 4.6 Main page of "Myanmar Sign Language Recognition" System

In the function of uploading, users can upload their desired sign images from the directory that the image exists. When the user clicks "browse", the system can work automatically to take user toward the file explore by importing "fileddialog" library from "tkinter". Figure 4.7 illustrates the upload function.

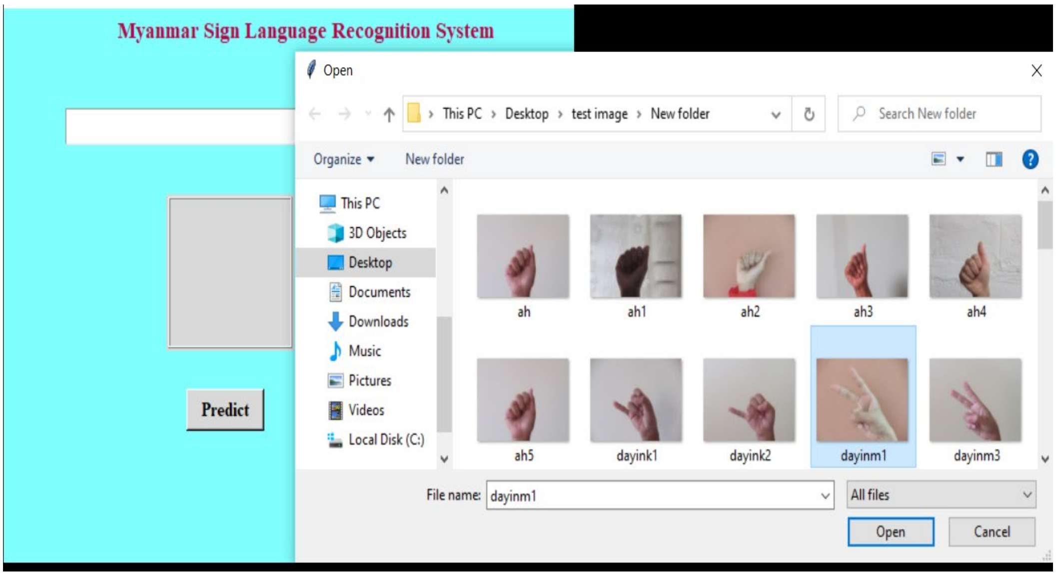


Figure 4.7 Uploading sign images to predict

User can view the meaning of sign image that is uploaded by clicking "Predict" button. The functions of "Predict" button include loading Support Vector Machine Model (SVM), preprocessing the uploaded image, extracting non-linear features by using coefficient of KPCA and showing the result with label based on probability function of the model. Figure 4.8 and 4.9 represent the process of predicting consonant and number sign images.

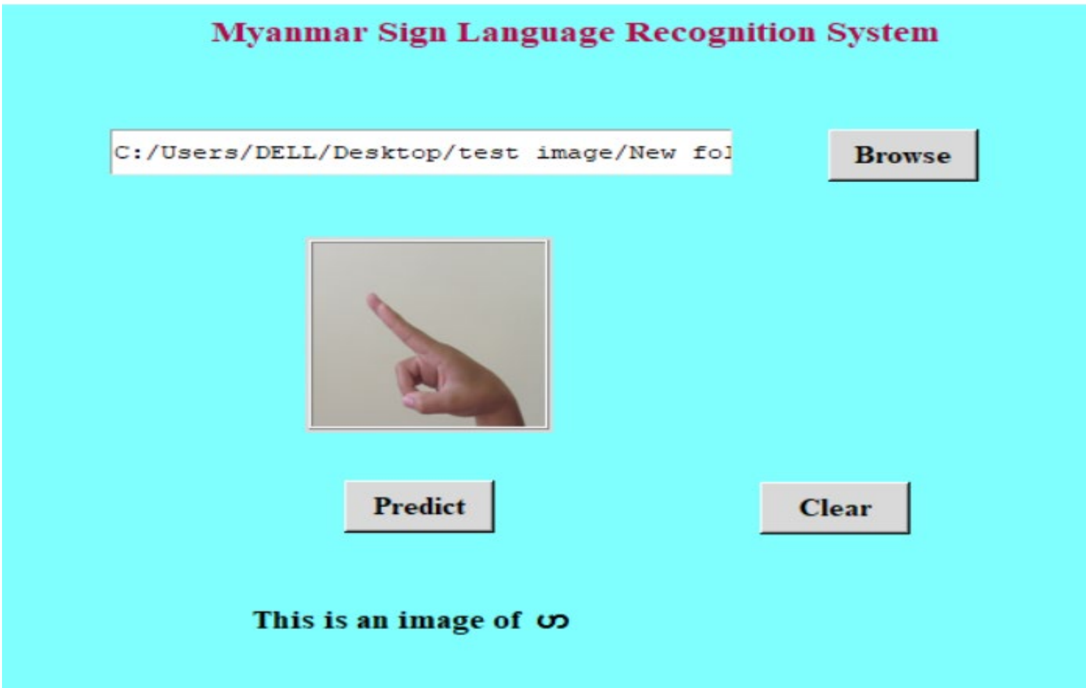


Figure 4.8. Recognition of sign image for ဟ (ha)

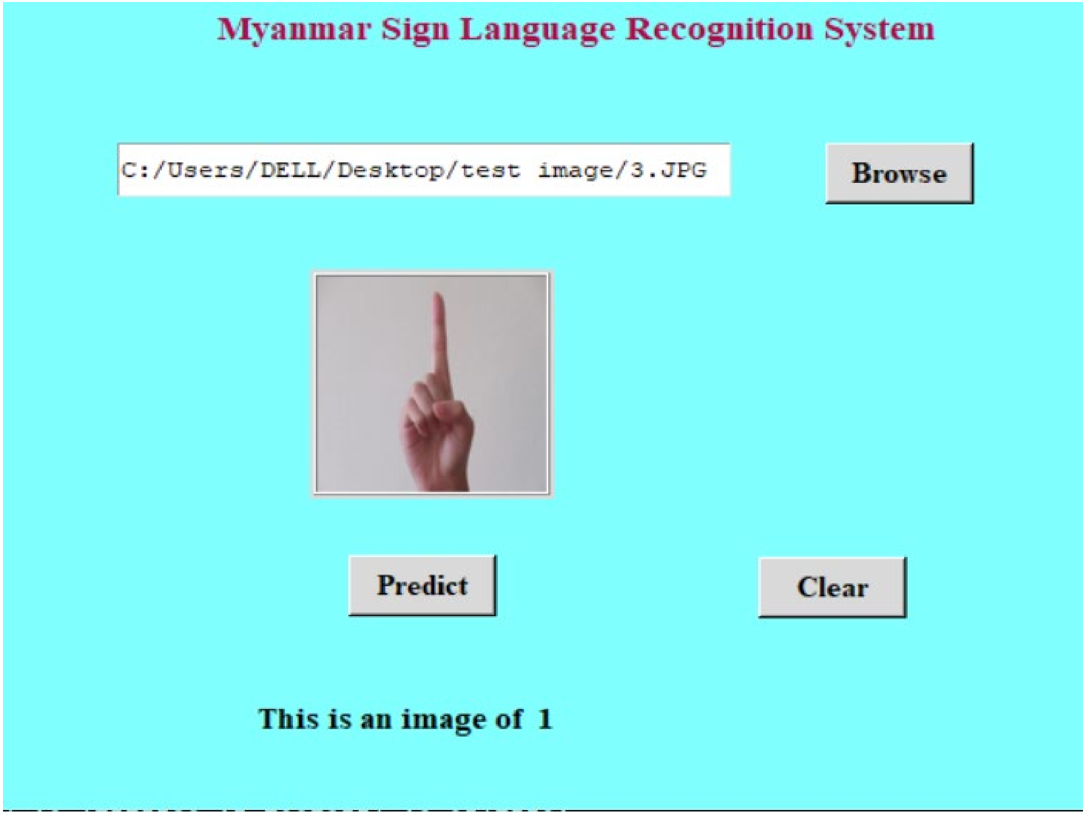


Figure 4.9 Recognition of sign image for ၁ (one)

4.2.4 Performance Criteria

The dataset was split up into 80:20 to test the performance of the system. As performance metrics for the test set in this system, precision, recall, and F-measure are applied. The precision of a category for a test set in the image classification process is determined by the ratio of the number of successfully categorized sign images. Recall is defined as the proportion of correctly classified images in the training set divided by the total number of images in that category. The precision and recall values are weighted to create the F1 score. In order to compute them, use the following formulas:

$$Recall = \frac{TP}{(TP+FN)} \quad (4.1)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (4.2)$$

$$F1\text{- }score = \frac{2*(Precision*Recall)}{(Precision+Recall)} \quad (4.3)$$

Where,

TP = True Positive (the occurrences of targets actually identified images)

FP = False Positive (the occurrences of targets that were not correctly identified images)

FN = False Negative (the occurrences of images which were not Identified at all)

4.2.5 Experimental Results in Number Dataset

The experimental results for dataset that include only the data of the Myanmar number are illustrated below. Figure 4.10 and 4.11 describe the comparison of the form of features after applying with Kernel Principal Component Analysis (KPCA) method and Principal Component Analysis (PCA) method before feeding into the classification stage.

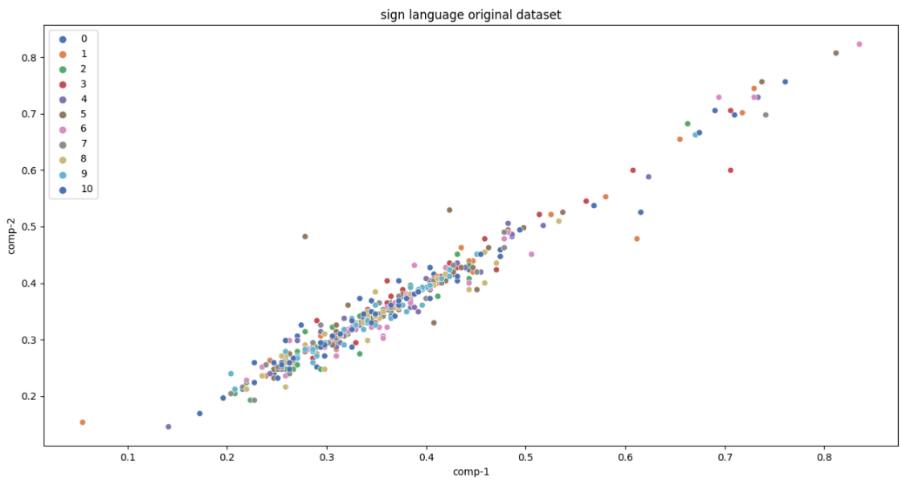


Figure 4.10. Scatter plot of original data

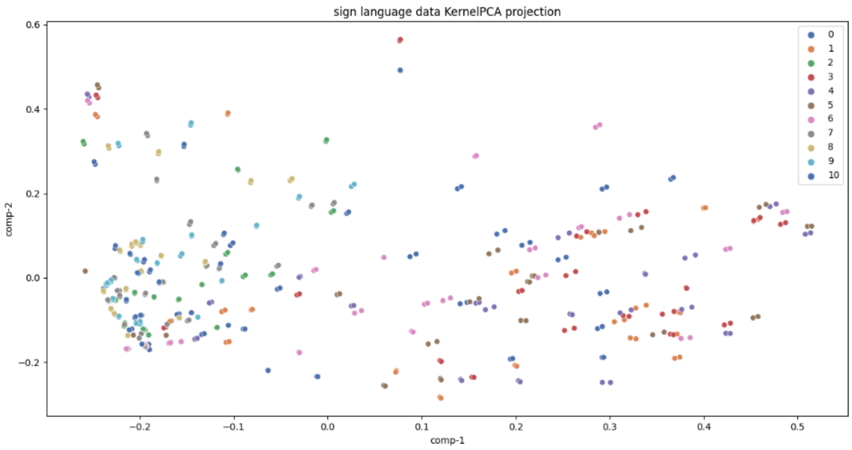


Figure 4.11. Scatter plot of data after reduction the data dimension with Kernel Principal Component Analysis (KPCA)

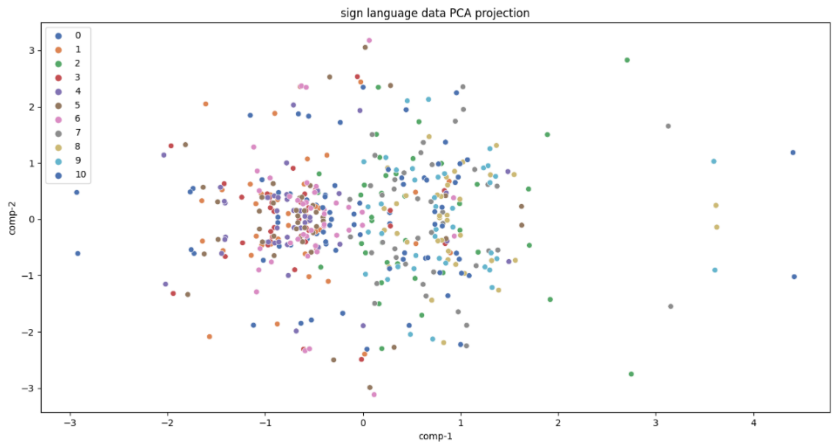


Figure 4.12. Scatter plot of data after reduction the data dimension with Kernel Principal Component Analysis (KPCA)

Performance of Number Dataset is tested with F-1 Score by applying Support Vector Machine (SVM). The original dimension of the data is 16384. That huge dimension was reduced with KPCA not only to keep the most variance non-linear data but also to speed up the machine learning process. Figure 4.13 represents the classification report on Number dataset with 600 features.

	precision	recall	f1-score
0	0.97	0.89	0.93
1	0.82	0.89	0.86
2	0.94	0.92	0.93
3	0.65	0.70	0.68
4	0.67	0.86	0.76
5	0.90	0.70	0.79
6	0.88	0.81	0.85
7	0.97	0.76	0.85
8	0.85	0.94	0.89
9	0.73	0.81	0.77
10	0.78	0.76	0.77
accuracy			0.82

Figure 4.13. Classification Report for Number Dataset

```
[[33  1  0  2  0  0  0  0  0  1  0]
 [ 0 33  0  4  0  0  0  0  0  0  0]
 [ 0  0 33  0  0  0  0  0  0  2  1]
 [ 1  4  0 26  6  0  0  0  0  0  0]
 [ 0  1  0  3 31  0  1  0  0  0  0]
 [ 0  1  0  2  5 26  3  0  0  0  0]
 [ 0  0  0  0  4  3 30  0  0  0  0]
 [ 0  0  1  1  0  0  0 28  0  3  4]
 [ 0  0  0  0  0  0  0  1 34  1  0]
 [ 0  0  0  0  0  0  0  0  4 30  3]
 [ 0  0  1  2  0  0  0  0  2  4 28]]
```

Figure 4.14 Confusion Matrix for Number Dataset

Recognition rates for each class are expressed in table 4.1. Recognition rate is calculated by using the following formula:

Recognition rate = (no. of correctly identified images/ Total no. of images) x 100

Table 4.1. Recognition rates for each class

Classes	Recognition Rate (%)
0	89%
1	89%
2	92%
3	70%
4	86%
5	70%
6	81%
7	76%
8	94%
9	81%
10	76%

Table 4.2 describes the comparison of accuracies between PCA with SVM and KPCA with SVM based on the number of samples. According to the following results, the system with Kernel Principal Component Analysis with Support Vector Machine has the highest accuracy of **82%.**

Table 4.2 Accuracies based on number of samples for number dataset

Number of samples (Number Dataset)	**PCA with SVM (RBF kernel)**	**KPCA with SVM (RBF kernel)**
77	26%	**30%**
462	12%	**36%**
1342	52%	**82%**

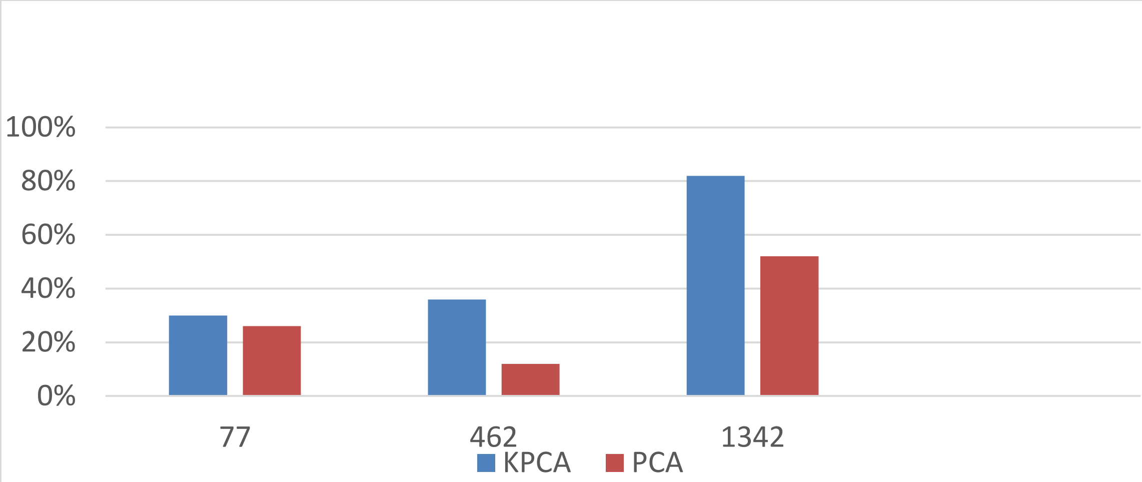


Figure 4.15. Comparison of experimental results for number dataset

4.2.6 Experimental Results in combination of Number and Consonant Dataset

The experimental results ae presented from feature extraction method, Kernel Principal Component Analysis (KPCA), with Support Vector Machine (SVM) to classify sign images. GridSearchCV library is used in the dataset that includes both Number and Consonant Datasets according to tune the best parameter values (C and gamma) of Support Vector Machine (SVM). Table 4.3 represents the accuracies of system based on different features. That figure illustrates about accuracies of reduced features with KPCA on the combination of Number and Consonant dataset.

Table 4.3. Accuracies based on different features

Number of features (Combination of Number and Consonant Datasets)	Accuracy (%)
50	78%
100	81%
700	83%

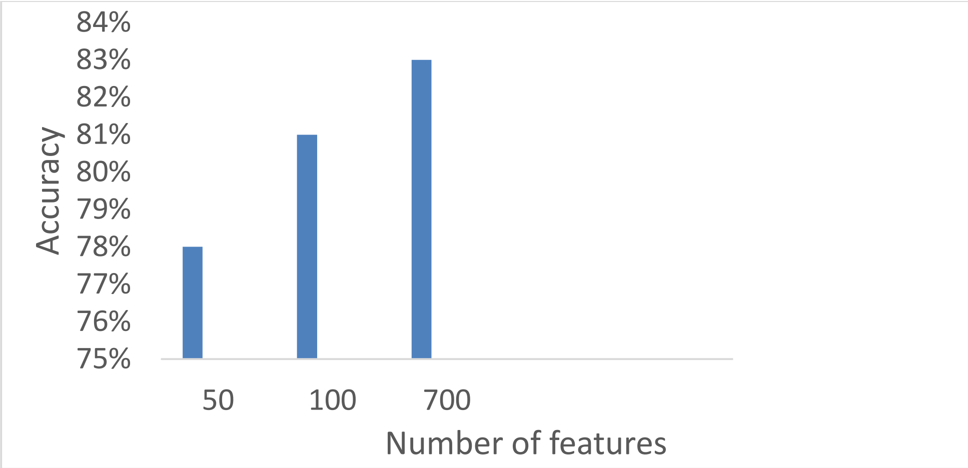


Figure 4.16. Comparison of Experimental Results based on different features

Error rate for each class is described in table 4.4 and table 4.5. According to the following table 4.4, error rates are higher in the classes of ၁ (one), ၂ (two), ၃ (three), ၄ (four), ဒ (da), ဓ (dha), ဗ (ba) and ဝ (wa) due to the same form of sign images. Error rates are calculated according to the below equation.

Percentage Error = ((Estimated Number – Actual Number)/ Actual number) x 100

Table 4.4 Error rate for Number Signs

Classes	**Error Rate %**
၀	0.04
၁	0.63
၂	0.77
၃	0.77
၄	0.78
၅	0.06
၆	0
၇	0.09
၈	0.03
၉	0.07
၁၀	0.03

Table 4.5 Error Rate for Consonant Signs

Classes	Error Rate %
က	0.08
ခ	0.04
ဂ	0.09
ဃ	0.03
င	0.02
စ	0.12
ဆ	0.09
ည	0
ဋ	0.13
ဌ	0.11
ဍ	0.06
ဏ	0.08
တ	0
ထ	0.19
ဒ	0.68

Classes	Error Rate %
ဓ	0.66
န	0.06
ပ	0.11
ဖ	0
ဗ	0.62
ဘ	0.14
မ	0.02
ယ	0.08
ရ	0.11
လ	0.01
ဝ	0.53
သ	0.05
ဟ	0.09
ဠ	0
အ	0.1

CHAPTER 5
CONCLUSION AND FURTHER EXTENSIONS

This thesis intends to develop sign language recognition system for Myanmar Numbers and Myanmar Consonants. In this chapter (5), conslusions, advantages and limitations of the system, and further extensions are described.

5.1 Conclusion

The proposed system presents the recognition of Myanmar Sign Language using Support Vector Machine (SVM) and Kernel Principal Component Analysis (KPCA). This system introduces a system that users can choose the desired sign image to predict. Kernel Principal Component Analysis (KPCA) is used to reduce the dimension and extract the most important information. This system can be used to recognize sign gestures without the need of expensive resources and it is able to speed up the process of machine learning. When the performance of the system is tested with SVM by splitting the dataset into 80:20, KPCA with SVM has the accuracy score of 82% compared with PCA with SVM.

5.2 Advantages and Limitations of the System

The proposed system serves user-friendly, high-performance, and scalable recognition system for sign language. As a result, the sign language recognition system based on KPCA is more accurate in predicting the data non-linearly. Dimensional reduction by KPCA produce correct result and cannot delay the generalization steps in the machine learning.

However, our proposed system doesn't recognize the dynamic signs and finding the parameters for parameters of SVM can take some time.

5.3 Further Extensions

The proposed system is tested by using only static sign of Myanmar Numbers and Myanmar Consonants. The dataset can be extended with dynamic sign by combining some computer vision methods with Kernel Principal Component Analysis (KPCA).

AUTHOR'S PUBLICATIONS

[p1] Eaint Thu Thu Khaing, Yi Mon Shwe Sin, "Myanmar Sign Language Recognition System using Support Vector Machine(SVM) and Kernel Principal Component Analysis (KPCA)", International Journal of Parallel & Soft Computing.

REFERENCE

[1] Ni Htwe Aung, Su Su Maung, Ye Kyaw Thu, "Sign Language Recognition for Myanmar Number Using Three Different SVM Classifiers", NCSE 2019, 27th - 28th June 2019, Yangon

[2] Face Recognition between Two Person using Kernel Principal Component Analysis and Support Vector Machines developed by Ivanna K. Timotius, Iwan Setyawan, and Andreas A.

[3] Myint Tun, Thida Lwin, "Real-time Myanmar Sign Language Recognition System using PCA and SVM", International Journal of Trend in Scientific Research and Development(IJTSRD), Volume 3 Issue 5, August 2019.

[4] Lean Karlo S. Tolentino, Ronnie O. Serfa Juan, August C. Thio-ac, Maria Abigail B. Pamahoy, Joni Rose R. Forteza, and Xavier Jet O. Garcia, "Static Sign Language Recognition Using Deep Learning", International Journal of Machine Learning and Computing, Vol. 9, No. 6, December 2019

[5] Jay Suthar, Devansh Parikh, Tanya Sharma & Avi Patel , "Sign Language Recognition for Static and Dynamic Gestures ", Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence, Vol 21 Issue 2 Version 1.0 Year 2021.

[6] Aishwarya Kulkarni, Pranav Halgekar, Girish.R.Deshpande, Anagha Rao, AishwaryaDinni, "Dynamic sign language translating system using deep learning and natural language processing", Turkish Journal of Computer and Mathematics Education, Vol.12 No.10(2021), 129-137

[7] Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, Malihe Alikhani, "Including Signed Languages in Natural Language Processing", arXiv:2105.05222v2 [cs.CL] 22 Jul 2021

[8] Swe Zin Moe, Ye Kyaw Thu, Hnin Aye Thant, Nandar Win Min, Thepchai Supnithi, "Unsupervised Neural Machine Translation between Myanmar Sign Language and Myanmar Language", Journal of Intelligent Informatics and Smart Technology, VOL. 4, APRIL 2020

[9] Adithya V, Rajesh R., "A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition", Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19), 10.1016/j.procs.2020.04.255

[10] P. S. Santhalingam, P. Pathak, J. Košecké, H. Rangwala, et al. "Body Pose and Deep Hand-shape Feature Based American Sign Language Recognition". In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA). IEEE. 2020, pp. 207–215.

[11] M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta, and M. Aktaruzzaman. "A new benchmark on american sign language recognition using convolutional neural

network". In: 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE. 2019, pp. 1–6.

[12] http://cs231n.stanford.edu/reports/2016/pdfs/214_Report.pdf

[13] A Survey of Hand Gesture Recognition Methods in Sign Language Recognition

[14] Communication between Deaf-Dumb People and Normal People

[15] Intelligent Sign Language Recognition Using Image Processing

[16] Hand Gesture Recognition based on Digital Image Processing using MATLAB

[17]https://www.researchgate.net/figure/System-architecture-for-feature-extraction-and-pattern-recognition-used-by_fig1_341208194

[18]https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/

[19]https://www.researchgate.net/figure/The-conceptual-framework-of-the-kernel-principal-component-analysis-KPCA-method_fig1_319581917

[20] https://www.educative.io/answers/what-is-feature-extraction

[21]https://www.researchgate.net/publication/260952140_A_Detailed_Review_of_Feature_Extraction_in_Image_Processing_Systems

[22] https://www.unite.ai/10-best-image-processing-libraries-in-python/

[23]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7730729/#:~:text=Feature%20extraction%20is%20a%20family,number%20of%20the%20original%20ones.