Visvesvaraya Technological University

BELAGAVI, KARNATAKA

ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ

ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

Report on Project Phase – 2

"Butterfly Species Classification Using Advanced Machine Learning Techniques"

Submitted by

1. **Vaishnavi N S 4JN19IS114**
2. **Srushti H S 4JN19IS103**
3. **Likith L S 4JN19IS040**
4. **Venkatesh G J 4JN19IS116**

Under the guidance of

Dr. Pavan Kumar M.P Ph.D., K-SET

Associate Professor
Dept. of IS&E,
JNNCE, Shivamogga

Department of Information Science & Engineering

J N N College of Engineering
Shivamogga-577204

2022-2023

National Education Society ®

J N N COLLEGE OF ENGINEERING

SHIVAMOGGA - 577204.

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Project entitled

"Butterfly Species Classification Using Advanced Machine Learning Techniques"

Submitted by

1. **Vaishnavi N S** **4JN19IS114**
2. **Srushti H S** **4JN19IS103**
3. **Likith L S** **4JN19IS040**
4. **Venkatesh G J** **4JN19IS116**

students of 8th semester B.E. ISE, in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2022-23.

Signature of Guide **Signature of HOD**

Dr. Pavan Kumar M.P Ph.D., K-SET
Associate Professor,
Dept. of IS&E,
JNNCE, Shivamogga

Dr. R Sanjeev Kunte M.Tech, Ph.D
Professor & Head,
Dept. of IS&E,
JNNCE, Shivamogga

Signature of Principal

PRINCIPAL
JNNCE, Shivamogga

1. Examiner **2. Examiner**

ABSTRACT

Automatic identification of butterfly species is an important research in the field of lepidopterology. The limited ability of the human eye to distinguish typical textures on butterfly species is one of the primary reasons for proposing a system for butterfly identification based on pattern recognition. Various species of butterflies have different textures on their wings. Due to the diversity of species, there is a significant degree of similarity and it is difficult to distinguish butterflies apart. Low accuracy and slow speed recognition are issues with butterfly identification and classification. Further, the number of taxonomists and trained technicians have decreased over the years. A generalized system for automatic identification and classification of butterfly species is proposed for eradicating the constraints of traditional approaches of butterfly species classification. The proposed work involves classification of butterfly images using traditional approach along as well as deep learning approach. The gray level co-occurrence matrix (GLCM and local binary pattern (LBP) is employed as primary tools for feature extraction. The features extracted from these techniques are fed as an input to the KNN classifier for classification and is accounted for traditional approach. The AlexNet classifier is used for the classification of butterfly species as a part of deep learning approach. The results obtained from traditional and deep learning techniques are compared and a comprehensive feasibility study is carried out successfully. Thus, the proposed work helps in the recognition and identification of butterfly species using advanced machine learning techniques and provide contribution to the field of entomology in developing a faster, and more precise techniques for accurate classification of butterfly species.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the effort with success. We would like to express our gratitude and thanks to **Prof. (Dr.) K. Nagendra Prasad**, Hon' Principal, Jawaharlal Nehru New College of Engineering and Head of the Department **Prof. (Dr.) R. Sanjeev Kunte**, Department of Information Science & Engineering for providing a good working environment and for his constant support and encouragement. We would like to thank our guide **Dr. Pavan Kumar M. P.**, Associate Professor, Department of Information science and Engineering for his expert guidance, initiative and encouragement that led me through the presentation. Also, we would like to thank project coordinators, Department of Information Science and Engineering for their support. We are highly indebted to them for their guidance and constant supervision as well as for providing necessary information regarding the project.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff and non-teaching staff of the Department of Information Science and Engineering. Also, we would like to thank **Mr. Chethan G. B.**, Ph.D Research Scholar in Entomology, Department of Entomology, Kittur Rani Chennamma College of Horticulture, Arabhavi for providing his support. Finally, we would like to extend our sincere regards to all our family members for their support.

Vaishnavi N S **(4JN19IS114)**
Srushti H S **(4JN19IS103)**
Likith L S **(4JN19IS040)**
Venkatesh G J **(4JN19IS116)**

TABLE OF CONTENTS

CHAPTER NO		TITLE	PAGE NO
		ABSTRACT	**i**
		ACKNOWLEDGEMENT	**ii**
		TABLE OF CONTENTS	**iii-v**
		LIST OF FIGURES	**vi-vii**
		LIST OF TABLES	**viii**
CHAPTER 1		**INTRODUCTION**	**1-6**
	1.1	Preamble	1
	1.2	Applications and Usage	6
	1.3	Constraints	6
	1.4	Organization of the Report	6
CHAPTER 2		**LITERATURE SURVEY**	**7-22**
	2.1	Related Works	7
	2.2	Problem Statement	22
	2.3	Objectives	22
	2.4	Summary	22
CHAPTER 3		**DATASET AND FEATURE ASSESSMENT**	**23-49**
	3.1	Background	23
	3.2	Dataset Samples	23
	3.2.1	Monarch Butterfly	24
	3.2.2	Red Admiral Butterfly	25
	3.2.3	Common Buckeye Butterfly	26
	3.2.4	Black Swallowtail Butterfly	27
	3.2.5	Cabbage White Butterfly	28

	3.2.6	Crimson Patched Longwing Butterfly	29
	3.2.7	Blue Mormon Butterfly	30
	3.2.8	Common Rose Butterfly	31
	3.3	Image Property Assessment	32
	3.3.1	Standard Deviation	32
	3.3.2	Mean Deviation	34
	3.3.3	Sharpness	35
	3.3.4	Coefficient of variance	37
	3.4	Image Quality Assessment	38
	3.4.1	Peak Signal to Noise Ratio (PSNR)	38
	3.4.2	Structural Similarity Index Method (SSIM)	39
	3.5	Summary	49
CHAPTER 4		**SYSTEM DESIGN AND IMPLEMENTATION**	**50-86**
	4.1	Dataset Collection	50
	4.2	Pre-processing	50
	4.2.1	Resizing and Cropping	51
	4.2.2	Image Enhancement	51
	4.3	System Architecture	61
	4.4	Feature Extraction	62
	4.4.1	Gray level Co-occurrence Matrix	63
	4.4.2	Local Binary Pattern	64
	4.4.3	kNN Classifier for Traditional Butterfly classification	65
	4.5	Introduction to Deep Learning	66
	4.5.1	AlexNet for Butterfly Specie Classification	67
	4.6	Results	73
	4.6.1	Feature Extraction Results	73
	4.6.2	kNN Classification Results	78

	4.6.3	Deep Learning Classification Results	82
	4.7	Feasibility Study	86
	4.8	Summary	86
CHAPTER 5		**EPILOGUE**	**87-90**
		REFERENCES	88-90

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	Monarch Butterfly Dataset	24
3.2	Red Admiral Dataset	25
3.3	Common Buckeye Dataset	26
3.4	Black Swallowtail Dataset	27
3.5	Cabbage White Dataset	28
3.6	Crimson-Patched Longwing dataset	29
3.7	Blue Mormon dataset	30
3.8	Common Rose dataset	31
3.9 (a)	Blur image	36
3.9 (b)	Sharp image	36
3.10	Illustration of the 10-90% rise distance on blurry and sharp edges. (Image Courtesy: imatest.com)	36
4.1	Flow chart for System design	62
4.2	AlexNet architecture	68
4.3	Feature Extraction	73
4.4	Feature Extraction for Monarch	74
4.5	Feature Extraction for Red Admiral	74
4.6	Feature Extraction for Buckeye	75
4.7	Feature Extraction for Black Swallowtail	75
4.8	Feature Extraction for Cabbage White	76
4.9	Feature Extraction for Crimson Patched Longwing	76
4.10	Feature Extraction for Blue Mormon	77
4.11	Feature Extraction for Common Rose	77
4.12	kNN Classification for Monarch	78
4.13	kNN Classification for Red Admiral	78

4.14	kNN Classification for Buckeye	79
4.15	kNN Classification for Black Swallowtail	79
4.16	kNN Classification for Cabbage White	80
4.17	kNN Classification for Crimson Patched Longwing	80
4.18	kNN Classification for Blue Mormon	81
4.19	kNN Classification for Common Rose	81
4.20	AlexNet Classification for Monarch	82
4.21	AlexNet Classification for Red Admiral	82
4.22	AlexNet Classification for Buckeye	83
4.23	AlexNet Classification for Black Swallowtail	83
4.24	AlexNet Classification for Cabbage White	84
4.25	AlexNet Classification for Crimson Patched Longwing	84
4.26	AlexNet Classification for Blue Mormon	85
4.27	AlexNet Classification for Common Rose	85
4.28	kNN accuracy	86
4.29	AlexNet accuracy	86

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
3.1	Property and Quality Assessment for Monarch Butterfly Dataset	41
3.2	Property and Quality Assessment for Red Admiral Butterfly Dataset	41
3.3	Property and Quality Assessment for Buckeye Butterfly Dataset	43
3.4	Property and Quality Assessment for Swallowtail Butterfly Dataset	43
3.5	Property and Quality Assessment for Cabbage White Butterfly Dataset	44
3.6	Property and Quality Assessment for Crimson Longwing Butterfly Dataset	46
3.7	Property and Quality Assessment for Blue Mormon Butterfly Dataset	47
3.8	Property and Quality Assessment for Common Rose Butterfly Dataset	48
4.1	Property and Quality Assessment for Pre-processed Monarch Butterfly Dataset	53
4.2	Property and Quality Assessment for Pre-processed Red Admiral Butterfly Dataset	53
4.3	Property and Quality Assessment for Pre-processed Buckeye Butterfly Dataset	54
4.4	Property and Quality Assessment for Pre-processed Swallowtail Butterfly Dataset	55
4.5	Property and Quality Assessment for Pre-processed Cabbage White Butterfly Dataset	56
4.6	Property and Quality Assessment for Pre-processed Crimson Longwing Butterfly Dataset	58
4.7	Property and Quality Assessment for Pre-processed Blue Mormon Butterfly Dataset	59
4.8	Property and Quality Assessment for Pre-processed Common Rose Butterfly Dataset	60

CHAPTER 1

INTRODUCTION

This chapter provides a brief overview of lepidopterology, butterflies and their importance in the ecosystem, and the different families of butterflies. This chapter also presents the applications, usage, and constraints of the lepidopterology study.

1.1 Preamble

Lepidopterology is a subfield of entomology dedicated to the investigation of moths and butterflies. Entomology is the study of insects, and lepidopterology is the branch of entomology that focuses exclusively on insects with scaly wings that emerge from caterpillars. A considerable amount of the biological variety on our planet is made up of insects, and in recent decades, their number has been drastically dropping. Lepidoptera family, which makes up around 9% of the insect world order and includes about 20,400 species, includes the butterfly class of insects. Mature butterflies fly with noticeable flapping and have huge, frequently brilliantly coloured wings.

Butterflies have a four-stage life cycle and the caterpillar's feeding plant is where wingless adults deposit their eggs. The caterpillars develop, sometimes extremely quickly, before pupating in a chrysalis once they are completely grown. The adult insect crawls out of the pupal skin when the metamorphosis is over, and once its wings have dried and grown, it takes flight. Butterflies are frequently polymorphic, and many species utilize aposematism, camouflage, and imitation to thwart predators. The monarch and the painted woman, however, go great distances. Depending on the species, mature butterflies can live anywhere from a week to over a year. While some species may remain latent in their pupal or egg stages and hence survive winters, many species have lengthy larval life phases. Most butterflies have a four-stage life cycle, consisting of an egg, a larva (or caterpillar), a pupa (or chrysalis), and an imago (adult). The scales that cover butterflies' bodies are each the result of a single epidermal cell. The two enormous compound eyes dominate the little head. The club-tipped antennae are made up of several segments. The butterfly's thorax is dedicated to locomotion. There are two legs on each of the three thoracic segments. The wings are carried by the second and third thoracic segments. The hindwings are smaller, more rounded, and contain fewer

stiffening veins than the forewings, which have dense veins along their leading margins to reinforce them. The friction of their overlapping sections serves to coordinate the forewings and hindwings rather than being hooked together. A pair of spiracles on the front two segments are utilized for breathing. The stomach and genital organs are located in the 10 segments that make up the abdomen. The terminal segment is altered for reproduction, and the first eight segments feature spiracles. A butterfly cannot fly once it leaves the pupal stage until the wings are fully extended. A newly hatched butterfly must spend some time pumping haemolymph into its wings and allowing them to dry; during this period, it is particularly susceptible to predators. Many butterfly wings have vibrant patterns that would alert predators that they are poisonous.

Identifying butterflies begins with learning the six butterfly families. The first five are the Swallowtails, Brush-foots, Whites and Sulphury, Gossamer-wings, and Metalmarks are called the true butterflies and the last group Skipper is considered separately. Swallowtail butterflies, members of the Papilionidae family, are significantly bigger and more widespread. The tail-like projections on the rear of the hind wing are referred to as "swallowtails." Also, they have distinguishable species-identifying wing colours and patterns. The biggest family of butterflies, the Nymphalidae, has over 6,000 species that have been identified worldwide. They are known as brush-footed butterflies. There seem to be just two pairs of legs on several members of this family. They include crescents, checkers, longwings, admirals, and many of the most prevalent butterflies, including monarchs and other milkweed butterflies. Whites and Sulphurs (Family Pieridae) have wings that are either pale white or yellow with black or orange patterns. In contrast to brush-foots, which have shorter front legs, they are tiny to medium-sized butterflies with three pairs of walking legs. They are widespread, with up to 1,100 species identified, and only exist in areas where cruciferous or legume plants are. The term "Gossamer-winged" alludes to the wings' sheer appearance, which is frequently flecked with vibrant hues. Skippers (Family Hesperiidae) are different from other groups; it is easy to differentiate from other butterflies. Compared to other butterflies, a skipper has a robust thorax that makes it seem like a moth. They also have different antennae than other butterflies. The name “skippers” describes their movement, a quick, skipping flight from flower to flower. Skippers tend to be drab in colour. Most are brown or grey with white or orange markings. There are about 3500 skippers described. In 1964, a good historical analysis of butterfly traps was created. There were two different types of traps created; the most popular one was the Van Someran-Rydon trap, which was composed of a cylinder of netting and wire that was

closed at the top and open at the bottom and linked to a base made of a piece of plywood or plastic on which bait is placed. It should ideally be between 100 and 130 cm tall and at least 25 cm in diameter. The trap's top part has a cone-like form. Although the trap netting colour in butterfly catchability has never been tested, dark colours such as green, black, and grey have been successfully used, whereas white netting has apparently kept away some butterflies.

The reestablishment of the food chain in the environment is greatly aided by butterflies. Due to their quick and sensitive responses to modest habitat or climate change, as well as the variety and responses of other flora or fauna, they are used as bio-indicators. These insects' amazing wing colours contribute to the settings' aesthetic appeal. Butterflies are the ecology's natural indicators; we can learn everything about a better ecosystem from these insects. They are efficient pollinators, and butterflies visit the flower to consume nectar, creating a win-win situation. Several butterfly species travel great distances and transport pollen to plants located far away from one another. This pollen movement causes genetic heterogeneity in plant species and improves their resilience to various diseases. These insects serve as biological pest control agents as well as food for other creatures including birds, reptiles, and amphibians. Several butterfly species contribute to lowering air pollution. The carbon dioxide in the atmosphere is reduced by these insects. The monarch butterfly and caterpillar host plants absorb carbon dioxide and lessen air pollution.

With the rapid advancement of insect and plant observation in the recent years, images of insects and plants have become more prevalent and the need for accurate animal and plant identification has also increased dramatically. One of the important tasks is the classification and identification of butterflies. It can support related studies conducted by biologists, natural scientists, ecologists and other researchers. By automatically recognizing the butterflies, scientists may help with butterfly conservation while also collecting data on the distribution of certain butterflies, counting the population of butterfly species, monitoring and evaluating butterfly species and ecosystems, analysing changes in the ecological environment. Butterflies play a crucial role in the environment. It has become apparent in recent years that certain butterfly species have significant medicinal significance because they can give us medications that could potentially save lives. Nevertheless, some butterfly behaviours are also very beneficial to researchers who study and track the effects of global climate change. Knowing the many types of butterflies can help preserve them from extinction by documenting and studying their behaviours. They contribute significantly to the ecosystem's overall stability as well. Due to their short lifespan and ability to react swiftly to these changes,

butterflies are also exceptionally good at detecting small ecological changes. More and more scholars are becoming interested in automatic butterfly recognition, and there have been numerous types of studies on butterfly recognition applications.

The natural environment is complex and harsh, and the image resolution is low. The inter-class differences in butterfly subgroups are typically small and therefore the proposed work frequently need small local variances to distinguish between them. Compared with inter-class differences, there are usually large intra-class differences, including object attitude, lighting, scale, shielding, angle and background. The butterfly's shape, colour, texture, and other properties are retrieved from the input butterfly image for the butterfly classification, and the associated classification result is then returned based on similarity. It can be fairly simple to become polluted by noise during the collection and transmission of butterfly images, which affects the quality and blurs the images. Therefore, improving butterfly images is essential. It is helpful for enhancing the contrast of the picture areas, highlighting specific elements of the image areas, and enhancing the image quality. Image enhancement is the technique of enhancing the original data's quality and information content before processing. It involves modifying the brightness, contrast, sharpness, and colour balance of an image to improve its visual appeal or to draw out specific features.

Common techniques of image enhancement are contrast stretching, brightness correction, sharpening, noise reduction, colour adjustment, histogram equalization, deblurring. There are many more image enhancing methods that can be employed, depending on the particular requirements and objectives of the image processing project. Image enhancement techniques can be used to improve the quality and clarity of images of butterfly specimens, which can be helpful for their accurate classification. By enhancing the images, important features of the butterfly, such as wing patterns, markings, and colours, can be more easily visualized and analysed. The identification and distinction of several species of butterflies based on their physical characteristics is one of the primary objectives of butterfly specie classification. These traits can be made more noticeable and identifiable using image enhancement techniques including contrast adjustments, brightness correction, and noise reduction, enabling more precise classification. The efficiency of computer vision algorithms that are used for automated butterfly classification can also be increased by enhancing the image. These algorithms can more accurately recognise and categorise various butterfly species by enhancing the image quality, which can be helpful for a number of applications like ecological monitoring, biodiversity protection, and many more.

In addition to being a fundamental and significant component of image analysis and computer vision tasks, image segmentation is one of the essential technologies in image processing. Image segmentation is a critical stage in the classification of butterfly species. The base, the veins, and the scales of a butterfly's wing may all be separated using image segmentation. This division can help in locating and analysing the various wing components that are crucial for identifying species. Features such as shape, size, colour, and texture can be retrieved from each region of the wing by segmenting it into distinct sections. The many butterfly species can then be classified using these characteristics. More accurate classification outcomes may result from correct butterfly wing segmentation. The classification method can obtain a more thorough understanding of the butterfly's wing, resulting in more precise species identification, by dividing the various parts of the wing and extracting information from each region. Automating image segmentation can save time and improve the accuracy of butterfly species identification. Automated segmentation can also lower analysis variability and human error.

GLCM stands for "Grey Level Co-occurrence Matrix," an approach to texture analysis that is frequently employed in image processing. The spatial correlations between the grey-level values in an image are examined by the statistical technique known as GLCM. In a GLCM, the grey-level values of the pixels are used to determine the spatial relationship between pairs of pixels. The GLCM is a matrix that keeps track of how frequently different pairs of grey-level values appear in an image at particular distances and angles. A number of texture properties, including contrast, homogeneity, entropy, energy, and correlation, can be extracted from an image using the GLCM. These characteristics can be used to describe an image's texture and have a variety of uses, including image segmentation, classification, and retrieval. Overall, GLCM is a useful tool for extracting texture features from an image that can be used for a variety of applications in image processing and computer vision. In image processing and computer vision, the Local Binary Pattern (LBP) approach is widely used for texture analysis. LBP is a simple yet powerful method for describing an image's local structure, and it has found use in a variety of fields like texture classification, object recognition, and face and object recognition. Based on a comparison of the intensity levels of the pixels surrounding a central pixel, the LBP operator is used. By comparing a pixel's intensity with those of its surrounding pixels, the operator determines the texture pattern at each pixel in the image. In particular, a binary pattern is produced for each pixel in the image by contrasting its intensity value with the intensity values of the pixels around it. The local texture of the image is then

represented by the binary pattern. The effectiveness of LBP's computations is one of its main benefits. The operator is appropriate for real-time applications since it can be computed rapidly and simply, even for huge images. Another benefit is that because binary patterns are formed using local intensity comparisons rather than absolute intensity values, they are resilient to noise and variations in illumination.

1.2 Applications and Usage

The main application of the proposed model is classifying butterfly species with minimized human intervention. It also contributes to entomology in developing a faster, more precise classification system for butterflies. The study also aids in the identification of butterfly species without causing physical harm to them. A programmed butterfly species framework should be created to diminish the weight of manual butterfly species characterization.

1.3 Constraints

Due to the species' close resemblances and difficulty to tell them apart, butterfly identification and classification is a difficult process. Also, compiling datasets is time-consuming. The target butterfly species must be manually captured using a trap in the conventional method of butterfly identification. This technique wastes both time and effort since the insect must be kept in small areas for a lengthy period of time in order to distinguish between the wings and the belly. The recognition of butterflies from an image captured in low and high light conditions is a major constraint. Since there are so many species in the Lepidoptera family with a variety of wing shapes and colours, it is difficult to identify them all. However, we provide an optimistic contribution in identifying every class of butterfly.

1.4 Organization of the Report

Chapter 2 presents a detailed literature survey on the theory of Butterfly species classification using Diversified Computer Vision and Advanced Machine Learning techniques, followed by the problem statement and objectives of the work. The preparation of the dataset is covered in detail in chapter 3. The system design and methodologies are discussed in chapter 4. Chapter 5 includes conclusion of the proposed work.

CHAPTER 2

LITERATURE SURVEY

This chapter presents the related work of diversified computer vision and advanced machine learning techniques. Various algorithms and theories have been proposed by different research authors to determine better performance and accuracy scores. The following section clearly states the significant literature.

2.1 Related Works

R. Andrian et al., (2020) [1] presented an article entitled Butterfly identification using grey level co-occurrence matrix (GLCM) extraction feature and k-nearest neighbour (KNN) classification. The authors of this study used the K-Nearest Neighbour (KNN) algorithm and the grey-level co-occurrence matrix (GLCM) to identify and categorize different species of butterflies. As a dataset, pictures of several butterfly species were gathered. The dataset was pre-processed using scaling where the images were cropped to 256 x 256 pixels, segmentation where the foreground was separated from the background, and grayscale where the image was converted into grayscale and then the final dataset was prepared. The authors extracted a co-occurrence matrix for identification using feature extraction. The co-occurrence matrix depicts the relationship between two pixels in various spatial arrangements and orientation directions, where d indicates the distance between the two pixels and represents the rotational angle. Texture properties including Angular Second Moment (ASM), Contrast, Homogeneity, and Correlation are extracted using the matrix. The KNN algorithm is employed for classification. The closest neighbour (or most similar) picture to the test sample is used by the KNN method to classify the photos. The classification has been successfully applied in the butterfly pattern recognition process, with the highest accuracy rate of 91.1%. The work's limitation is that it calls for adequate lighting and utilizes various angles, such as using the lower wing angle and the angle when they are not flapping their wings to capture butterfly images.

XUE Ankanga et al, (2019) [2] presented an article entitled Automatic Identification of Butterfly Species Based on Gray-Level Co-occurrence Matrix Features of Image Block. This paper is based on automatic identification performed by the development of digital image processing techniques and pattern recognition techniques. It is based on the

principal components and linear discriminant analysis. The content-based image retrieval process is being employed which includes analysing face textures and colour features. The authors propose a computer vision system according to which Gray Level Co-Occurrence Matrix (GLCM) is extremely effective for describing texture and the process is carried out through the image blocks and weight-based k-nearest neighbour (KNN) classifier. On the other hand, a set of weights is taken into account to limit the impact of nearby locations. The comparison outcome according to the feature vector and classification choice is utilized for identification in the identification procedure. There are four main technologies for categorizing butterfly photos that are addressed. The author concentrates on local texture throughout the image processing stage when each input picture is shrunk and placed in the middle of the standard area. In the image block stage, the butterfly can be effectively represented by a grey-level co-occurrence matrix. Before feature extraction, the gray image is blocked to 16 areas where these blocks are named by their positions followed by selecting 3 areas that contain special information in different locations of the butterfly. In the feature extraction stage, a GLCM matrix of an image is created considering 3 important parameters i.e., the distance, the orientation, and the level of grey level conversion parameters. The contrast, homogeneity, correlation, and energy are calculated using their respective formulas. The findings show that a key factor in the decision-making process is the distance in feature space. The paper's approach may also be used to identify specimens that have been shattered, which is a plus. Also, it enhances the regional characteristics of butterflies that are crucial for identification. The disadvantage of the above work is that the identification rate drops to 80% when the butterfly species are increased to 50 species.

D. Prasad Mishra et al., (2018) [3] presented an article entitled CNN for Butterfly Classification. In this work, the authors considered Convolutional Neural Networks (CNN) for the identification and classification of butterfly species. The author trains the CNN model on an image data set and tests the network with the results obtained by it on a different dataset consisting of 90,000 images which are classified into 952 butterfly species. Each image in the data set was of 640x480 pixels in dimensions then resized to 300x300 pixels by cropping and resizing, to fit the training image specifications of the Network. An open-source programming library, TensorFlow, is used for numerical calculation. There are six main layers in each network architecture. An innovative classifier is comparable to a convolutional layer. The weight matrix is shifted through the whole layer in the pooling layer. The purpose of the pooling layer is to display all training data. The best approach to handle overfitting is to drop

out of layering. Nonlinearity is added to the network using the ReLu layer. The introduction of the weights and the learning rate of the CNN system are controlled by batch normalization. The Inception unit consists of seven convolutional layers plus a fully connected layer from a standard neural network and one filtering layer. Stochastic gradient descent is an optimizer in this network. In the next network model build, all models were trained using batches of size 7. After training for 18 hours, the model predicted the correct label with a probability of 0.43. The model performed well in both the validation and test sets, which led the authors to draw the conclusion that the model's overall findings were rather gratifying. The model was partially trained on lesser requirements, which is the observed drawback.

N. Nabila Kamaron Arzar et al., (2019) [4] presented an article on Butterfly Species Identification Using Convolutional Neural Networks. The author analyzed the need to develop a system for the automatic classification of butterfly species. A proper recognition system has benefits in analyzing species and biodiversity studies. The dataset used for the study includes four different species of butterflies. Further, the paper discusses the approaches for developing an automatic system for butterfly classification using Convolutional neural networks. The Convolutional neural network is used for image classification and segmentation processes. A pre-trained CNN model called Google Net is employed due to its high recognition efficiency and is built deep with 22 layers of parameters. The images of butterflies are passed into the first layer, the Convolutional layer where all the necessary features are extracted and transferred to the Pooling layer. In the pooling layers, the images are reduced by volume without affecting the data. In the next layer i.e., the max pooling layer where maximum values are extracted from each divided region. Further, the data is fed into a fully connected layer where the images are linearly classified. This layer is connected to a SoftMax layer which calculates the probabilities of images being classified into the right family. The accuracy of the system is calculated through the confusion matrix. The positive aspect of the paper is that the model not only identifies the dataset for which it was designed but also, the other datasets which it has “not seen”. The negative aspect of the paper is that the methodology missed out on using better data augmentation techniques which could have increased the accuracy of the model.

Ruoyan Zhao et al., (2019) [5] presented an article on Butterfly Recognition Based Faster R-CNN. The paper discusses the classification and recognition of butterfly species using the faster R-CNN method. This work uses a dataset from the third China Data Mining Competition consisting of 5695 photographs of 11 species. The input image considers the shape, colour, and texture of butterflies. Further, the paper considers Deep Learning

technology to apply faster R-CNN. The methodology helps in locating the accurate position of butterflies in each image and marking their categories. It is a traditional method to identify objects in the image and if it is an object, convert them to the size compatible with the convolution layer to determine the kind of object. The R-CNN method is used for feature extraction, classification, and regression processes. It improves the performance, accuracy, and classification speed of the system. The network first uses a set of pooling, convolution, and ReLu layers for feature extraction. The pooling layer extracts the feature maps from the image, the convolution layer specifies the target category. The result from feature extraction is fed to the RPN layer which determines the area target values. Determining the ground truth and a loss function is necessary for RPN training. The intersection over the union (IoU) of the target area serves as the ground truth, and the target area is identified if the IoU is bigger than 0.7. The next step is to feed the output into a convolution layer for classification, from which the system's accuracy may be determined. The advantage of the above work is that the use of Faster R-CNN technology helps in detecting faster, and automatic recognition systems. All the images should be cropped to the same ratio and filtering of the images has about 8 steps for the final testing data set is the negative aspect of the work.

Juanying Xie and Yinyuan Lu., (2021) [6] presented an article on investigations of butterfly species identification from images in natural environments. In this paper, the author describes the competition held in China where the participants were told to use the faster R-CNN for it was the latest deep learning algorithm, but it seemed that the identification rates were not good enough as the faster R-CNN algorithm uses a lot of time to process. In this study, the severely uneven ecological butterfly database is partitioned and enhanced using a novel method. They discovered that the RetinaNet deep learning method is the most effective at tackling butterfly identification using butterfly photos collected from the natural world. In terms of map accuracy, they attained 79% (mean average precision). The usual feature extraction methods are used in the first phase to extract the butterfly wing's form, colour, and texture properties. To predict the butterfly species, a classifier must be trained in the subsequent phase. The Chinese competition demonstrated that the RetinaNet's map accuracy was superior to that of YOLO. Without applying ensemble approaches, they achieved results of 79.8% in terms of map, with their best map being just 77.7%. The researchers used the YOLO-V3 to conduct trials to determine whether or not to include the same type of butterflies in certain photographs (that is to evaluate the redundant species). The right algorithm and network for the butterfly identification challenge are then discovered through tests using the

two-stage deep learning algorithm FPN and one-stage deep learning algorithms such as RetinaNet, SSD, YOLOv3, and CenterNet. They obtain the best map of 79.7% compared to the state-of-the-art results on the same database.

Xin Chen and Bin Wang (2021) [7] presented an article entitled Gaussian Convolution Angles: Invariant Vein and Texture Descriptors for Butterfly Species Identification. This study uses the innovative descriptor known as Gaussian Convolution angles to classify the different species of butterflies. By projecting the butterfly vein image function and intensity image function along a set of vectors that begin at a common contour point and end at the remaining contour points, the proposed GCA creates a set of vectors that captures the intricate vein patterns and texture patterns of butterfly images. To create multiscale GCA descriptors, several scales of Gaussian convolutions are applied to the resultant vector functions. This model has invariant geometric manipulation, including rotation, scaling, cropping, and lighting. The author used a dataset of 832 photos from 10 different species of butterflies that was made accessible to the public. This approach has a 92% accuracy rate. The author has created a novel model called the gaussian convolution angle (GCA), which successfully captures the butterfly vein and texture patterns for precise butterfly species classification. Since butterflies have two wings, which causes them to have significant pose variation, and because the butterfly shape cannot provide enough cues for identification of the species. In this paper they have 2 steps to conduct: First propose a novel descriptor, Gaussian convolution vein angle (GCVA), which is invariant to geometrical transformations including translation, scaling, and rotation to capture the complex vein patterns of butterfly images. Next, in order to capture the texture patterns of butterfly photos, a unique descriptor called Gaussian convolution texture angle (GCTA) that is invariant to both geometrical changes and illumination change is proposed. The model inherits the ability to create detailed vein and texture characteristics due to its multi-resolution property, the suggested model also achieves a classification accuracy of 92.03%, indicating exceptional performance. The paper's main finding is that the GVCA and GVTA are insensitive to changes in illumination and geometrical transformations like rotation and scaling.

Xiaotian Jia et al., (2020) [8] presented an article on Lepidoptera Classification through Deep Learning. This paper is based on two-stage deep learning technologies. The first stage is single phase Single Shot Detection (SSD) and the second stage is Faster R-CNN. The two main areas of image recognition are object detection and object classification. Object detection involves finding all the objects required from an image and categorizing them. The paper focuses on two-stage models for its accuracy and performance. The R-CNN stage is further

classified into two sub-stages; Proposing a region containing the necessary objects using a selective search algorithm followed by running a classification network to get the categories of butterfly images. The dataset used consists of 9187 images of 20 species of butterflies collected from Google ImageNet and Bing Images. Results are further improved by rotation, resizing, and mean normalization processes. The images are classified based on color and pattern, shape, size, tentacles, and resting state. The models used are faster_rcnn_inception_v2 and ssd_mobilenet_v1 for R-CNN and SSD respectively. The proposed methodology involves creating a bounding box for regions of interest and using them to classify. The positive impact of this research is that the Faster R-CNN system provided better results with respect to accuracy and performance due to the adoption of two-stage systems. The negative impact includes; Faster R-CNN took more time than SSD to train the model and the model is sensitive to image size and the number of pixels that are used to identify the objects of interest.

Fathimathul Rajeena P. P and Rasha Orban, (2022) [9] presented the of use transfer-learning-based neural network models to identify the species. Based on varied butterfly orientations, shooting angles, butterfly lengths, occlusion, and backdrop complexity, they chose 15 uncommon butterfly species from the Kaggle dataset, which has 10,000 photos and 75 distinct species. They divided the butterfly species into several groups using a variety of CNN-based architecture, including VGG16, VGG19, MobileNet, Xception, ResNet50, and InceptionV3. In this architecture, they were glad that the inceptionV3 gave an accuracy of 94.66%. Using the GoogLeNet, a pre-trained CNN architectural model they employed the transfer learning-based neural model, and 120 photos of 4 different species of butterfly were used in this as a small dataset and 90% accuracy was achieved in recognizing the species. They investigated 19 distinct species of the pyridine butterfly family using two surface properties known as bidirectional patterning (LBP) and a grey-level co-occurrence matrix, totaling 190 butterfly images (GCLM). LBP and GLCM had accuracy rates of 90.45% and 91.25%, respectively. The model's beneficial effect is that data augmentation is done to lessen overfitting, enhance accuracy, and minimize computing costs and generalization problems. The disadvantage is that in order to attain optimum accuracy, a sizable dataset is necessary.

Ayad Saad Almryad, Hakan Kutucu, (2020) [10] collected the dataset from the Butterfly Monitoring and Photography Society of Turkey. The dataset was prepared considering 10 species of butterflies and the images were resized to 224 x 224 pixels. The authors proposed to use of Deep learning for the identification and classification of butterflies. The core architecture of deep learning, the convolutional neural network (CNN), is utilized.

CNN is a multi-layer feed-forward perceptron neural network comprising layers that include output (the softmax layer) layers for classification and a series of convolutional and pooling layers in succession. The categorization of the butterfly photos was performed using convolutional neural networks (VGG16, VGG19, and ResNet50) built on fine-tuned transfer learning models. The advantage of this study is that it will address the issue of the amount of time needed for entomologists to identify butterfly species from the data set. The negative side of this research is that it mostly uses pattern pictures with obvious morphological features which lack the ecological features of butterflies. The positive inference from the paper is that it results in the highest success when VGG architecture is employed. The model considers images with obvious morphological features and lacks ecological pictures. This is the negative aspect of the proposed work.

Fan Li et al. (2017) [11] presented an article entitled Automated identification of butterfly species based on HoMSC and GLCMoIB. The author utilized the Gray-Level Co-Occurrence Matrix of the Image Block and the Histogram of Multi-Scale Curvature (HoMSC) to automatically identify butterfly species (GLCMoIB). The k-nearest neighbour (KNN) classifier is utilized to achieve precise identification results. Butterflies in a digital image are automatically identified by their form and texture. 750 images of 50 distinct butterfly species were utilized in this project. The accuracy percentage for the automated butterfly species identification method was 98%. The species of a butterfly may be automatically identified using a number of useful techniques. There are still some problems, though, such as a large dimensional vector, inadequate extensibility, and a low recognition rate. Artificial neural networks are discussed by the author as a way to approach issues from several directions. The butterfly's most important characteristic is its texture. Studies show that GLCM has the benefit of reflecting texture variations and regularity over the whole picture, making it a helpful tool for characterizing texture. Improving the butterfly's picture was necessary in order to extract the features for automated butterfly species identification. The HoMSC characteristic was then used to derive the main identification. Three of the eight blocks of the picture that were removed before the texture features were extracted yielded GLCMoIB features and a feature vector for subsequent identification. Compare the feature vector's findings with the classifier's identification of the butterfly species. For butterfly identification, four key technologies have been identified: image pre-processing, image block, feature extraction, and classifier development. The four qualities that have historically been used to distinguish different species of butterflies are shape, texture, colour, and vein. A series of experiments revealed that it was

difficult to obtain vein characteristics from a digital image due to the butterfly's vibrant wing scales. The form and texture of the butterfly's wings were quite constant during its entire metamorphosis. As different species of butterflies have different wing forms, the form feature is used for straightforward identification. The designations left profile block, centre block, and right underwing block have been given to three of the image block's blocks. There were 24 matrices used in the selected picture block. Matrixes should be described by the four properties of contrast, homogeneity, correlation, and entropy. Contrast serves as a reflection of both depth and picture clarity. The GLCM elements that are farther from the diagonal of the matrix have higher contrast values, which translate to higher values. The immediate area's textural change is measured by homogeneity. The homogeneity score is higher the less the texture varies throughout an image. The correlation rates the consistency of the image's texture. The correlation value is higher when the values of the matrix's components are consistent. Energy, which is the quadratic sum of the matrix components, reflects the uniformity of the grayscale distribution and the depth of texture in a picture. The classifier design employs a weight-based KNN classifier. In this work, a reliable method for classifying butterfly species was presented. The results of the identification tests and comparisons demonstrate how well the extra features identified different butterfly species. A useful tool for layer-by-layer identification was the cascade classifier as well. For defining the contour and texture, HoMSC and GLCMoIB were both collected from single-winged butterflies and the average is constructive.

Mohammed Alweshah et al, (2020) [12] presented an article entitled The monarch butterfly optimization algorithm for solving feature selection problems. In this work, the author considered an optimization algorithm. The optimization method used in this research implements the monarch butterfly optimization (MBO) algorithm with FS using the k-nearest neighbour (KNN) classifier. Four metaheuristic methods are compared to the results of the hybrid whale optimization technique with SA (WOASAT), the ant lion optimizer (ALO), and an experiment with 18 benchmark datasets (ALO, GA, PSO, and WOASAT). Second, data preparation is crucial since many datasets contain extraneous data that might be confusing during data analysis. Feature selection is the most important pre-processing step since it reduces computing time and increases classification precision. The creation, assessment, and validation phases of the FS process are its three main steps. During the generating process, subsets are built from whole datasets. When complexity is high, every conceivable dataset will be extracted. A metaheuristic algorithm is suggested as a potential remedy. Metaheuristic algorithms, which are smart search strategies that imitate nature, are used to tackle a number

of FS issues. Population-based algorithms are employed, such as the MBO algorithm. By replicating these butterflies' migratory patterns, several optimization problems are solved. The wrapper FS method used in this work uses a KNN classifier as an evaluator and the MBO as a search strategy. The KNN classifier establishes the accuracy rate for the attributes selected by the MBO. When selecting characteristics, precision and the number of attributes is important considerations. the difficulty of time. The lowest performance level of an algorithm's complexity is represented by Big-O encoding. Positive conclusions from the study include the following: MBO outperformed all other methods in terms of selection size rate; it averaged 88% accuracy across all datasets, which is excellent in this field of study; it also has great potential for use in identifying perfect, integrated solutions for applications that call for a quick search process. The WOASAT method is changed and not used in its original, unaltered form like the other methods in the comparison, which is a restriction of the work.

Yılmaz Kaya et al, (2014) [13] published an article on the Evaluation of texture features for automatically detecting butterfly species using an extreme learning machine. In this paper, the authors have considered the texture of biological images for feature extraction. For the purpose of recognizing entities and understanding surfaces, the texture is a crucial aspect of digital images. The quantification and use of such image texture properties are called texture analysis (TA). The texture that contains important information for image classification is characteristic intensity (or colour) variations that typically originate from the roughness of object surfaces. TA of biological images is a computer-based technique that extracts features mathematically to obtain a distribution of grey-level pixels in. Grey-level co-occurrence matrix (GLCM) and local binary patterns (LBP) are the most commonly used methods for texture analysis. The relationship between the reference pixel and the neighbouring pixels is calculated to determine the textural features of the image. From the information obtained from neighbouring pixel relationships, rough and smooth surfaces can easily be classified using the GLCM technique. The Local Binary Pattern is used to describe the structure of a texture. GLCM and LBP are used to have texture features of butterfly images and an extreme learning machine (ELM) is used for the identification and classification of butterflies. ELM is a single hidden layer feedforward artificial neural network (ANN) model. GLCM is a pixel-based image processing method. An element or a pixel p(Ø, d, i, j) of a GLCM of an image represents the relative pixel detail, where i is the grey level at location (x, y) and j represents the grey level of a neighbouring pixel at a distance d from the element location and an orientation angle Ø from location (x, y). The creation of the GLCM matrix is based on the distance between pixels,

the orientation or the pixel angle. In the proposed papers, the authors have used energy entropy, contrast, and homogeneity from the grey-level co-occurrence matrix for feature extraction. LBP methods attempt to decompose the texture into small texture units. The LBP operator labels the pixels of an image by using the value of the centre pixel as a threshold value of the pixel. If the neighbouring pixel value is greater than or equal to the centre pixel value, this pixel takes the value 1 otherwise it takes 0. ELM network is a customized single hidden layer feedforward ANN model. In ELM, input weights and bias values are generated randomly, but outputs are obtained analytically.

Hui Tang et al, (2020) [14] presented an article entitled Deep learning techniques for automatic butterfly segmentation in ecological images. This paper mainly focuses on efficient methods for the segmentation of butterfly images. The images used for classification are ecological images of butterflies, which not only has butterflies but a lot of other objects in the image including leaves, flowers, background, and so on. As a result, segmenting the butterflies from their ecological photos is a problem that must be solved before tasks of identification and classification as the effectiveness of the segmentation has a direct impact on the efficiency of classification. Traditional methods of object segmentation from ecological images include manually segmenting objects based on colour, texture, and other features. These methods can be further categorized into thresholding, region-based methods, clustering-based methods, and active contour models. The authors have designed a novel network structure called Dilated encoder network (DE-net) for automatic ecological butterfly image segmentation. In addition, the authors have used Hybrid cascade dilated convolution (HCDC) blocks to capture more high-level features and get high-resolution output. The DE-net block consists of two major modules, the dilater feature encoder module, and the decoder module. Each block in the encoder contains two convolution layers and one max pooling layer. The decoder module comprises simple up-sampling and deconvolution operations. The positive inference of this paper is that the dilated encoder network was able to capture more high-level features and get high-resolution output while segmentation and the authors were able to reduce the segmentation duration.

Reema Rodrigues et al., (2020) [15] proposed an article on Butterfly Species Identification Using Convolutional Neural Network. The project's basic goal is to consider about 10 butterfly species in order to classify them. The manual classification of butterflies involves time-consuming and expensive tasks such as data collection, recognition, and physical specimen picture archiving. Convolutional Neural Network (CNN) has been used as a result

because it is an automated method that doesn't need human involvement. 90% of the 832 photos in the 10 classes that were taken into consideration during the testing were correctly classified overall. The application's start is the upload of the butterfly image. The data's images are downsized to 224x224 pixels. Resized butterfly photos are classified using Google Net, a pre-trained neural network. Confusion Matrix is used to determine accuracy after the procedure is complete due to the successful CNN technique implementation utilizing Google Net as CNN. In this study, the data are divided into training and testing, each receiving 80% and 20% of the total data. The 4 split categories of the testing findings of the categorized butterfly species are the Black, Veined Tiger, Chocolate Grain Yellow, Grey Pansy, and Plain Lacewing. Around thirty images are considered for training and testing the count of TRUE and FALSE outcomes of identification in confusion matrix form. To distinguish between the various categories, the butterfly photos are kept in data frames. The most advantageous traits of each species were used to establish simpler and quicker similarity assessments than image texture techniques. First, we should resize the photos to 128x128 pixels, which is a typical size. Use the Label Encoding function and the One Hot Encoder function to split the columns containing the numerical category data and convert the labels to numeric form. We should divide the 832 photographs we have into a significant portion for training and the remainder for testing. The construction of CNN's network using a sequential model with around five layers is complete. In order to fit the model, batches of data are formed, the epoch is made equivalent, and the result is raised, including loss and accuracy %, thereby categorizing photographs of diverse species. The total accuracy of the technique used is close to 97.5%.

Nebojsa Bacanin et al., (2020) [16] presented an article entitled Monarch Butterfly Optimization Based Convolutional Neural Network Design. The paper proposes two sets of experiments to classify the Monarch Butterfly species. The main goal of the study is to create an automated system that will accurately design and produce CNN architectures. A fine-tuning of hyperparameters is carried out to achieve better accuracy. The hyperparameters considered for building the model are the number of convolution layers along with the number of filters, kernel size of each layer, dropout rate, activation function, and size of dense layers. The first part of the study involves developing a hybridized model and testing it on a set of standard benchmarks. The CNN structures are generated by employing an MBO (Monarch Butterfly Optimization) algorithm. The movement of the monarch butterfly has offered a foundation for the algorithm. The MBO algorithm contains two important operators, which are described as follows; The migration operator is the first and operates to update the monarch butterfly's

migration between Lands 1 and 2. The monarch butterfly's position in Subpopulation 2 is updated using the second operator, the butterfly adjusting operator. The MBO algorithm is a robust method to solve NP-based optimization problems. The second part of the simulation experiment is to adapt the proposed hybrid MBO and test it for CNN design. A Convolution Neural Network is a series of connected layers. The useful layers of the CNN architecture are; convolutional layers, pooling layers, which are used for down sampling, and dense layers. The parameters that have been applied to the optimization process are the hyperparameters of convolutional layers, fully-connected layers, and general CNN hyperparameters. To decrease the computation time, values of all the hyperparameters are segmented and defined with lower and upper bounds. The computation time is improved by training the structure with 50% of the data, out of which 20 best CNN architectures were fully trained and the training process was repeated 20 times to accomplish maximum accuracy. The results from both simulation processes are compared for their performance and accuracy.

Yilmaz Kayaa et al., (2015) [17] proposed an article entitled Two novel local binary pattern descriptors for texture analysis. In this work, the author discusses 2 novel versions of Local Binary pattern (LBP) descriptors for the extraction of texture features. The LBP uses the statistical intensity of an image, compares each pixel with its neighbours, and defines a relationship between them. This relationship is represented by a binary pattern and these binary patterns are converted into histograms. The primary objective of the proposed work is to develop a novel texture-searching approach. The traditional LBP descriptors suffer from a drawback. They only search predefined textures present in an image and may not detect the textures in special cases. To overcome this drawback, the author has proposed 2 novel versions of the descriptors, namely; nLBP and dLBP. The first histogram describes the relations between neighbouring pixels around the centre pixel. The neighbouring local binary pattern (nLBP) operators search the micro-textures in a local region. The neighbours of a pixel are limited to 8 and by thresholding these neighbouring pixels with a centre pixel, the binary pattern can be obtained. By applying the method, an LBP histogram consisting of pixel values varying from 0-255 is obtained. Each value of a pixel corresponds to different patterns. These 256 different patterns can be reduced by using uniform patterns, without compromising any loss of information. The second histogram was formed by comparing the values of a pixel in the centre with its neighbouring pixels in a specified orientation. The comparisons are carried out using a distance parameter. In the directional local binary pattern (dLBP), the comparison is carried out using the same orientation of the pixels based on the angle, which may bear 0◦, 45◦, 90◦ or

135◦ degrees, through clockwise. The texture features obtained from the nLBP and dLBP methods are fed into an Artificial Neural Network. This network architecture consists of an input layer, a hidden layer, and the output layer. The hidden layer extracts some of the important patterns from the input. 3 The activation function performs two significant purposes; It seizes the non-linear relationship between the inputs and helps to convert the input into a more useful output. The resulting accuracies from both histograms were higher than the traditional Local Binary Pattern descriptors.

Zhongqi Lin, Jingdun Jia, Wanlin Gao, and Feng Huang, (2019) [18] presented an article entitled "Increasingly Specialized Perception Network for Fine-Grained Visual Categorization of Butterfly Specimens". The paper presents a method for fine-grained visual categorization of butterfly specimens using an increasingly specialized perception network. The authors proposed a multi-stage perception network that consists of four stages, with each stage being increasingly specialized in recognizing the visual characteristics of butterfly specimens. The first stage performs general feature extraction using a pre-trained deep learning model, and the subsequent stages perform increasingly specialized feature extraction and classification tasks based on the butterfly's wing patterns. The authors evaluated the proposed method on a large dataset of butterfly specimens and compared its performance to other existing methods. The results showed that the proposed method achieved state-of-the-art performance in terms of accuracy and efficiency, demonstrating the effectiveness of the increasingly specialized perception network. One potential drawback of the proposed method is that it requires a large dataset of labelled butterfly images for training, which may be difficult to obtain for certain butterfly species or in certain environments. Additionally, the multi-stage nature of the approach may increase computational complexity and training time compared to other methods. Overall, the proposed method presents an effective approach for fine-grained visual categorization of butterfly specimens, with potential applications in biodiversity monitoring and conservation efforts. High Accuracy, Robustness, Scalability, Explain-ability are the advantages of the proposed work. Limited Generalizability, Complexity, Limited Interpretability and Lack of Comparison are the few drawbacks of the proposed work.

Maopeng Li et al., (2019) [19] presented an article entitled "Image-Based Butterflies Fine-Grained Classification". The paper presents a method for fine-grained classification of butterfly images using a deep learning framework. The authors collected a large dataset of butterfly images from different species and pre-processed the images by resizing and augmenting them. They then trained a deep learning model using a convolutional neural

network (CNN) architecture, which was designed to learn both local and global features from the images. The trained model was then used to classify new images of butterflies based on their species. The proposed approach utilizes a pre-trained convolutional neural network (CNN) model, namely ResNet-50, to extract the features of butterfly images. Then, a new classification layer is added to the pre-trained model to fine-tune it for butterfly classification. The study also investigates the effectiveness of transfer learning by comparing the performance of the proposed approach with that of the traditional CNN model trained from scratch. The experimental results show that the proposed approach achieved high accuracy in both bird and butterfly classification tasks. In particular, the fine-tuned ResNet-50 model achieved an accuracy of 93.1% and 98.8% in butterfly classification, outperforming the traditional CNN model trained from scratch. The results of the study showed that the proposed method achieved high accuracy in classifying butterfly species, even for visually similar species, and outperformed other existing methods for butterfly recognition. The advantages of the proposed work include Efficient training, Accurate classification, use of transfer learning model, and Applicability. However, there are few drawbacks such as Dataset bias, Model complexity, lack of Scalability and Interpretability.

Dongjun Xin, Yen-Wei Chen and Jianjun Li, (2020) [20] presented an article entitled "Fine-Grained Butterfly Classification in Ecological Images Using Squeeze-And-Excitation and Spatial Attention Module". The article proposes a deep learning approach for the classification of butterfly species in ecological images. The authors introduce a model that incorporates both squeeze-and-excitation (SE) and spatial attention modules to improve the accuracy of fine-grained classification. The SE module is used to recalibrate the feature maps of the convolutional neural network (CNN) by learning the importance of each channel. The spatial attention module is used to highlight the most relevant regions of the image, based on the learned attention maps, to improve the classification performance. The proposed model is evaluated on two publicly available datasets, i.e., the Butterfly Eco-Visual Dataset and the Butterfly Non-Visual Dataset. The results demonstrate that the proposed model outperforms state-of-the-art methods in terms of accuracy, with an accuracy of 98.4% on the Butterfly Eco-Visual Dataset and 95.6% on the Butterfly Non-Visual Dataset. Overall, the paper presents a promising approach for the fine-grained classification of butterfly species in ecological images, which could have important applications in biodiversity monitoring and conservation. The merits of the work are; The use of SE and SAM techniques helps to enhance the feature representation and selectively attend to informative regions in the input image, respectively.

The approach is computationally efficient and requires less memory compared to other methods, which makes it suitable for practical applications. The study provides insights into the effectiveness of SE and SAM in fine-grained image classification tasks. The proposed approach is evaluated only on butterfly classification datasets, and its generalization to other fine-grained classification tasks remains unknown. The study does not provide a comprehensive analysis of the interpretability of the proposed approach, which limits its applicability in certain domains is the potential drawback.

Lin Nie, Keze Wang et al., (2019) [21] presented an article entitled " Fine-grained Butterfly Recognition with Deep Residual Networks: A New Baseline and Benchmark ". The article proposes a new benchmark for fine-grained butterfly recognition using deep residual networks (ResNets). The authors first introduce the butterfly recognition dataset, which contains 24,780 images of 200 butterfly species. They then compare the performance of ResNets with other state-of-the-art methods, such as deep convolutional neural networks (CNNs) and support vector machines (SVMs). The results show that ResNets outperform other methods in terms of accuracy, achieving a top-1 accuracy of 94.7% and a top-5 accuracy of 99.2%. The authors also propose a new baseline for fine-grained recognition, which achieves a top-1 accuracy of 92.9% and a top-5 accuracy of 98.6%. Additionally, the authors perform an ablation study to investigate the importance of different components of ResNets, such as residual connections and global average pooling. They find that residual connections and global average pooling both contribute significantly to the performance of ResNets. Overall, this paper provides a new benchmark for fine-grained butterfly recognition and demonstrates the effectiveness of ResNets for this task. The main advantage of the proposed work is that the study used transfer learning to fine-tune the pre-trained deep residual network for the task of butterfly species recognition, which improved the performance of the model. The model may be prone to overfitting, especially when the dataset is limited or when the model is trained for a specific subset of butterfly species is the observed drawback.

Zhu et al., (2017) [22] presented an article entitled “ SVM and AlexNet deep learning for automatic butterfly segmentation in ecological images” suggested a cascade architecture employing support vector machine (SVM) as the classifier and a portion of the deep convolutional neural network (DCNN) to extract features. There are 1301 photos in all from 22 distinct Lepidoptera species in their dataset. Each image has its background removed before being scaled and cropped to a resolution of 227x227. The pre-trained AlexNet model's first

fully connected layer was used to extract the picture features. The collected characteristics from AlexNet were then used to train Chi-squared SVMs to categorise Lepidoptera species. They also conducted a comparison using several VGGNet layers, however the outcome was unsatisfactory. Their proposed methodology achieved 100% accuracy when AlexNet-fc6 was utilized as a feature extractor and took 200ms to classify species from an image.

2.2 Problem Statement

There are more than 20,000 species of butterflies, and 1,500 of them have been found in India. The existing approach for butterfly detection and classification approaches have low accuracy and a sluggish recognition speed because to its high similarity, numerous types, and features. This research aims to increase the effectiveness of automatic butterfly species identification and categorization via the application of a variety of computer vision and cutting-edge machine learning approaches. It takes time and experience to distinguish between different species of butterflies, and these two things are not always available. This project involves the development of a dataset, and methodology to classify and improve the efficiency of the automatic identification of butterfly species.

2.3 Objectives

1. To propose a dataset on different Butterfly species.
2. To conquer detailed literature.
3. To propose traditional approaches for the detection of butterfly species.
4. To develop an advanced machine-learning model for automatic butterfly classification.
5. To implement a feasibility study of butterfly classification.

2.4 Summary

In this chapter, brief literature on diversified computer vision and advanced machine learning techniques are described. The comprehensive literature involves articles from 2018 to 2022 is considered. Based on this survey, the pros and cons of butterfly species classification and techniques and advanced approaches adopted by the author are clearly illustrated. Finally, this chapter presents and formulates the problem statements and objectives of the work based on significant literature.

CHAPTER 3

DATASET AND FEATURE ASSESSMENT

3.1 Background

The domain of image-based classification and computer vision involves training machine learning models by feeding it with the dataset. In computer vision, a dataset is a curated collection of digital images that developers use to train and evaluate the efficiency of their algorithms. Automatic classification of butterfly images requires a huge dataset to increase the accuracy of the model. Image classification requires extraction of features from an image to observe some patterns in the dataset for classification. Efficiency of such feature extraction depends on the amount of information in the image. In this regard, a sizable dataset is gathered that includes photographs of butterflies that are incredibly detailed and rich in information. In this work, samples of eight species of butterflies are considered. The proposed dataset consists of images of *Danaus plexippus* (Monarch Butterfly), *Junonia coenia* (Common Buckeye), *Pachliopta hector* (The Crimson Rose) and *Papilio polyxenes* (Black Swallowtail), *Pieris rapae* (Cabbage white), *Vanessa atlanta* (Red Admiral), Papilio polymnestor (Blue Mormon), Pachliopta aristolochiae (Common Rose). There are around 100 images of each butterfly speices in JPEG and PNG formats in the dataset. The proposed samples of diversified species include both aesthetically enriched and ecological images. Ecological images are those that include the surrounding environment in the background and the subject is not separated from background. The images samples are captured in diversified distance with diversified environmental illumination and few sample are taken from internet sources. In this work, we have intended to collect our own dataset and incorporate it into the existing dataset. We were able to collect two species dataset samples from Butterfly Park, Bannerghatta National Park.

3.2 Dataset Samples

This section includes image samples of Monarch Butterfly (Danaus plexippus), Common Buckeye (Junonia coenia), Crimson Rose (Pachliopta hector), Black Swallowtail (Papilio polyxenes), Cabbage White (Pieris rapae), Red Admiral (Vanessa atlanta), Blue Mormon (Papilio polymnestor), Common Rose (Pachliopta aristolochiae) as well as details on the characteristics and behaviour of each butterfly species.

3.2.1 Monarch Butterfly

The Monarch butterfly belongs to the Nymphalidae family of milkweed butterflies. Depending on the locale, other popular names for this species include milkweed, common tiger, black veined brown, and wanderer. As shown in Fig. 3.1, it has wings with a distinctive black, orange, and white design. This species' wingspan ranges from 8.9 to 10.2 cm (3.5–4.0 in). Monarchs are mostly found in North, Central, and South America, although they are also occasionally seen in Australia, Hawaii, India, and other places.

Fig 3.1: Monarch Butterfly Dataset

3.2.2 Red Admiral Butterfly

The red admiral butterfly as shown in Fig 3.2, is one of several eye-catching and vibrant butterfly species that are found all over the world. It is a member of the Nymphalidae family and is known by the scientific name Vanessa atalanta. The red admiral butterfly has wings that are mostly black with vivid red bands and white markings, with an average wingspan of 6 to 8 cm. While the butterfly is resting on tree bark or other surfaces, its brown and speckled underside aids to give concealment.

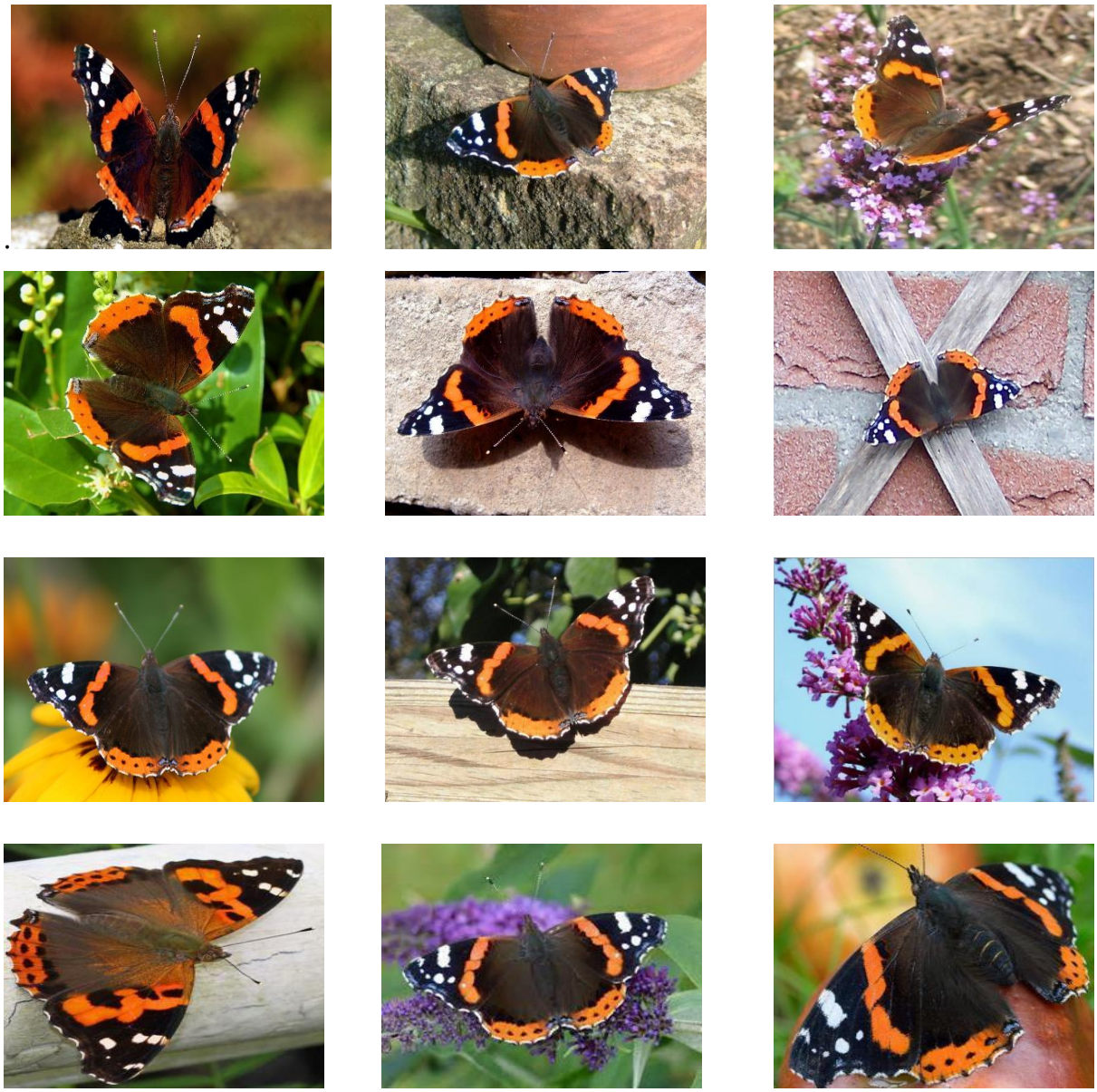


Fig 3.2: Red Admiral Dataset

3.2.3 Common Buckeye Butterfly

Butterfly belonging to the Nymphalidae family, Junonia coenia is sometimes referred to as the common buckeye. Brown dominates the common buckeye butterfly's coloration, with hints of orange, black, white, blue, and magenta. As seen in Fig. 3.3, the forewing has two proximal orange bars, a postmedian white band that encircles a large black eyespot and borders a smaller, more distant eyespot, and two distal orange marks. The forewing also has two proximal orange bars.

Fig 3.3: Common Buckeye Dataset

3.2.4 Black Swallowtail Butterfly

Black The Papilionoidea family of butterflies, which includes approximately 550 species of swallowtail butterflies, are big, vibrant butterflies. The birdwing butterflies of the genus Ornithoptera, the biggest butterflies in the world, are members of this family. Swallowtail butterflies engage in Batesian mimicry, a defense mechanism in which the butterflies' appearance closely resembles that of repulsive species. The forked appearance is found on the hindwings, which may be noticed when butterfly is resting with its wings wide as seen in Fig 3.4.

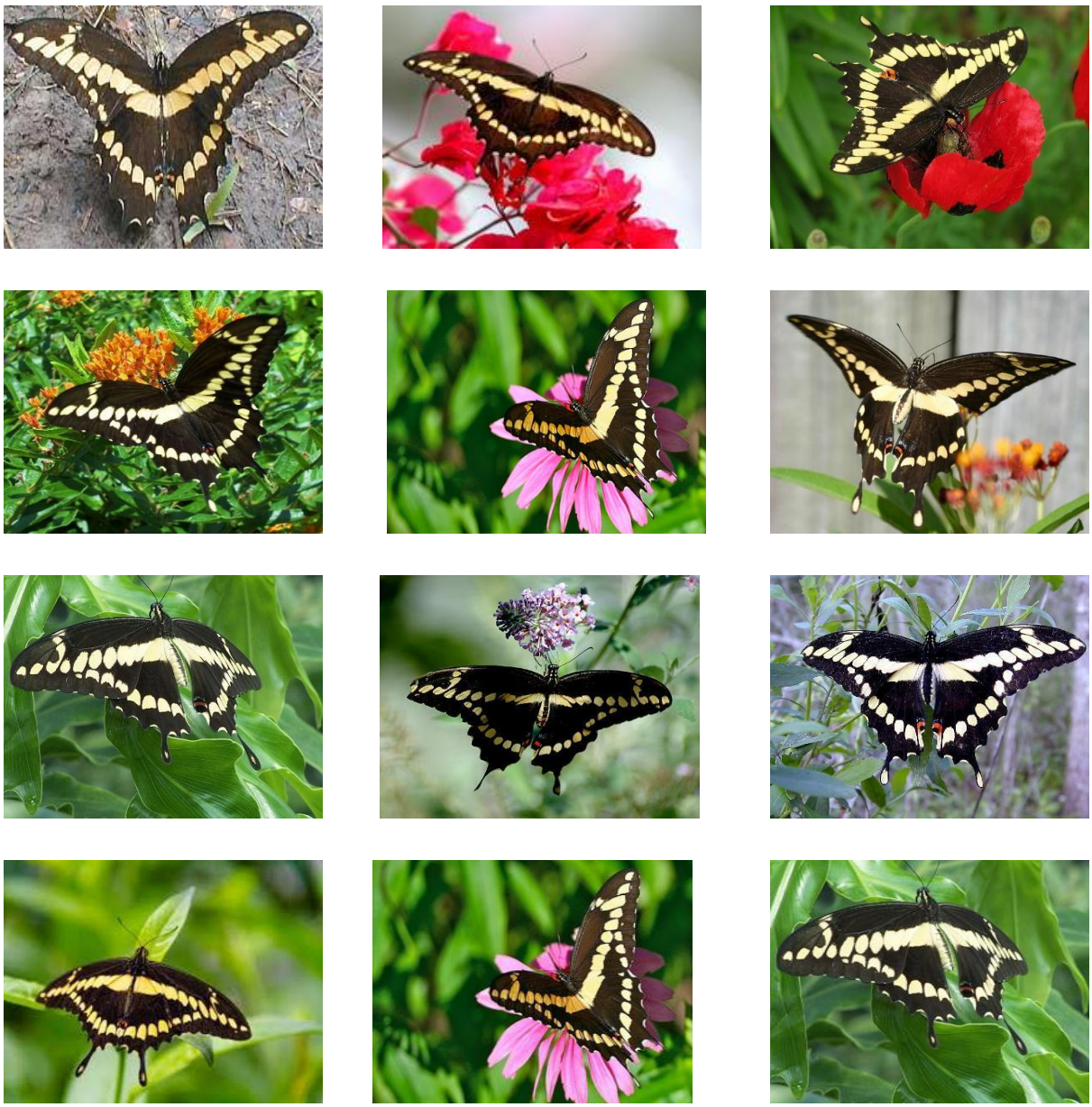


Fig 3.4: Black Swallowtail Dataset

3.2.5 Cabbage White Butterfly

The common butterfly species known as the cabbage white butterfly shown in Fig 3.5 may be found all throughout Europe, Asia, and North Africa. It is sometimes referred to as the little white butterfly or the cabbage butterfly. Pieris rapae is its scientific name, and it is a member of the Pieridae family. The cabbage white butterfly has forewings that are mostly white with black ends and a wingspan of around 4-5 cm. The number of black dots on a butterfly's forewings can be used to identify it as male or female. Male butterflies have one black mark on each forewing, while female butterflies have two.

Fig 3.5: Cabbage White Dataset

3.2.6 Crimson-Patched Longwing Butterfly

Heliconius erato, or the red postman, is one of about 40 neotropical species of butterfly belonging to the genus Heliconius. It is also commonly known as the small postman, the red passion flower butterfly, or the crimson-patched longwing. H. erato is a neotropical species, found from southern Texas to northern Argentina and Paraguay, and resides on the edges of tropical rainforests. It is philopatric, having a particularly restricted home range. As seen in Fig 3.6, it is a highly distinctive butterfly with noticeable white stripes on its forewings.

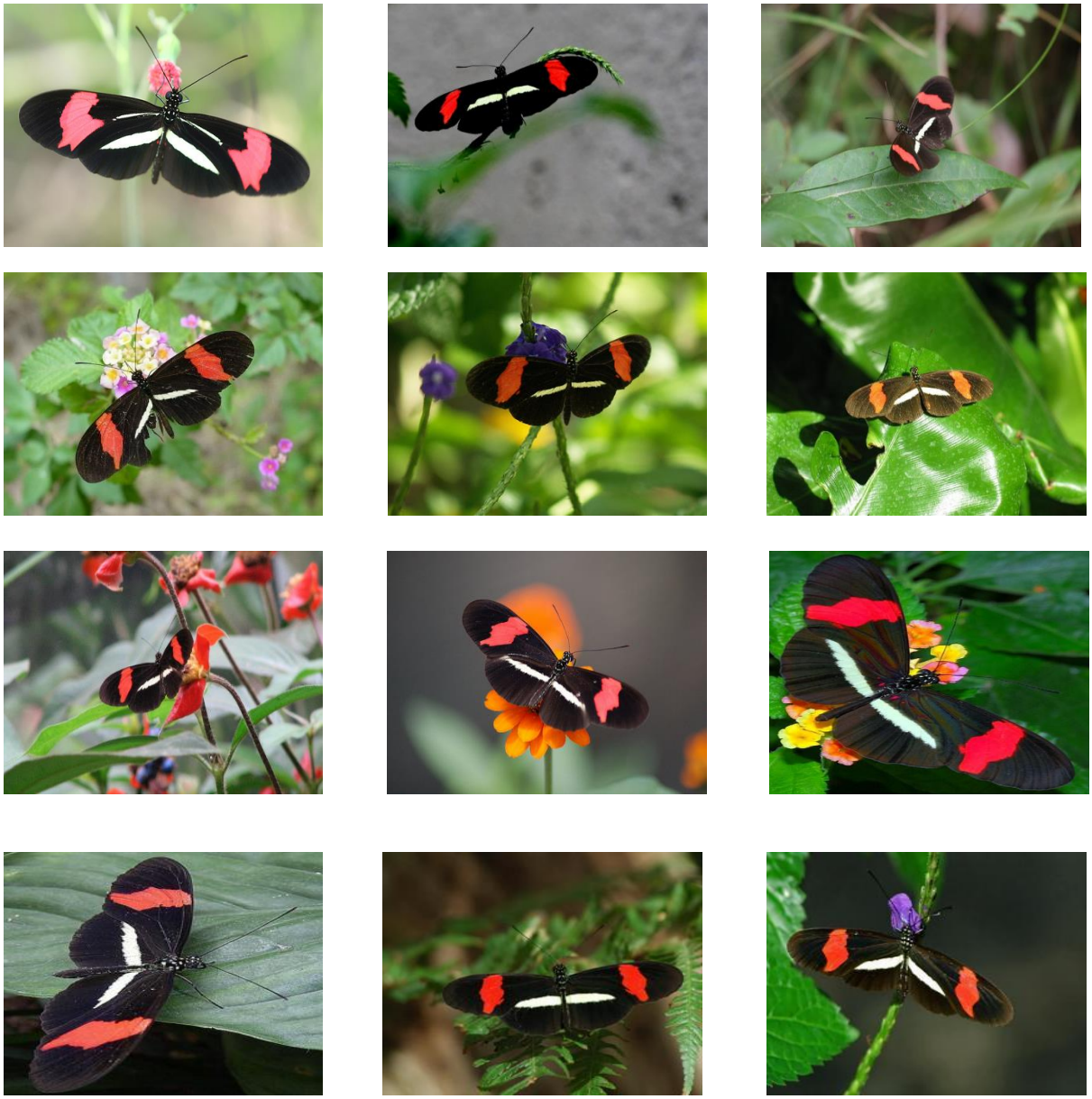


Fig 3.6: Crimson-Patched Longwing dataset

3.2.7 Blue Mormon Butterfly

The Blue Mormon butterfly (Papilio polymnestor) is a large and striking butterfly as shown in Fig. 3.7 is found in South Asia. It belongs to the family Papilionidae, which includes many other species of butterflies commonly known as swallowtails. The Blue Mormon butterfly has a wingspan of about 10-15 cm, and the upper side of its wings is a brilliant blue color with black margins and white spots. These butterflies are found in a variety of habitats, including forests, gardens, and urban areas. The Blue Mormon butterfly is considered to be an important species in its range due to its ecological and aesthetic value.

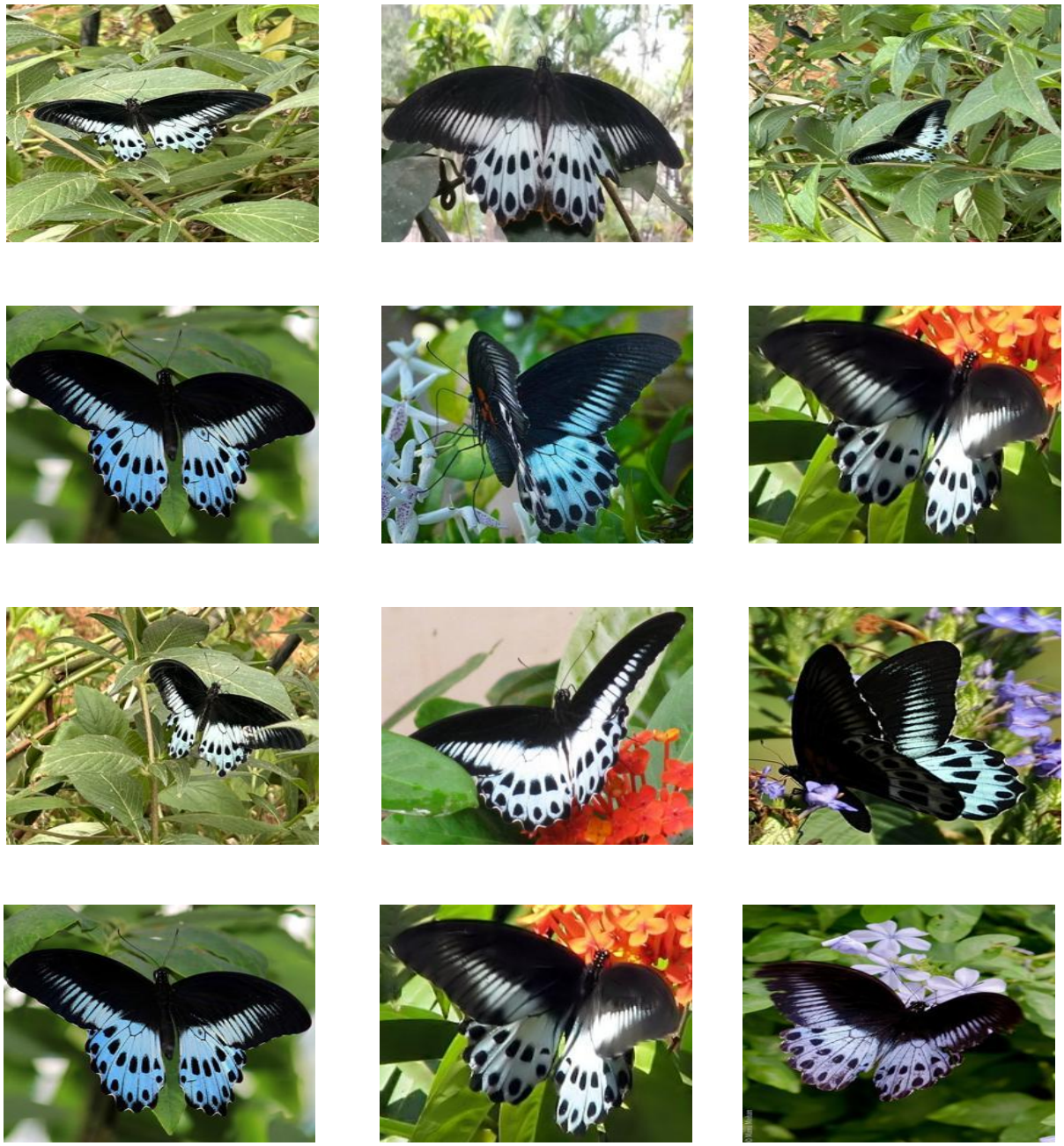


Fig 3.7: Blue Mormon dataset

3.2.8 Common Rose Butterfly

The Common Rose butterfly (Pachliopta aristolochiae) is a medium-sized butterfly belonging to the family Papilionidae. It is found throughout Asia, from India to Japan, and is widely distributed across Southeast Asia. The Common Rose butterfly has a wingspan of about 7-8 cm, and the upper side of its wings is black with bright red or pink markings. These butterflies are usually found in forested areas and gardens, and are active during the day. The Common Rose butterfly is considered to be a beautiful and popular species due to its striking coloration and graceful flight. It is also of ecological importance as it serves as a pollinator for many plant species as seen in Fig 3.8

Fig 3.8: Common Rose dataset

3.3 Image Property Assessment

Image property assessment refers to the process of evaluating the quality of an image based on various visual properties. This is accomplished by employing a collection of objective measurements that quantify the perceived image quality. When assessing the quality of an image, a number of factors are frequently taken into consideration. These include measures of distortion, noise level, noise, sharpness, contrast, brightness, and colour correctness. Each of these attributes is evaluated using a different metric, and the selection of metric is frequently determined by the particular application or use case. In many aspects of image processing, such as image compression, picture restoration, and image enhancement, image property assessment is crucial. It is feasible to optimise image processing methods and raise the overall quality of images for a particular application by precisely evaluating image quality. In this project standard deviation, mean, variance, and sharpness features are calculated for proposed dataset and experimental dataset samples. The detailed statistical property assessment values are presented in this chapter.

3.3.1 Standard Deviation

The standard deviation is a statistical measure used to quantify the amount of variation or dispersion in the pixel values of an image. It is a measure of the average amount of deviation of each pixel value from the mean pixel value of the image. The standard deviation is calculated as the square root of variance by determining each data point's deviation relative to the mean. Hence, it also used to measure the amount of variation in the set of pixel values in an image. If the data points are further from the mean, there is a higher deviation within the dataset. Thus, the more spread out the data, the higher the standard deviation.

The standard deviation of an image can be used to define a number of its characteristics. It can be used, for instance, to estimate the amount of noise in an image. The standard deviation can be used to assess other aspects of an image, such as texture and contrast, in addition to noise. Image processing algorithms can be created to enhance or restore photos, get rid of noise, and enhance overall image quality by looking at the standard deviation of an image. Mathematical notation of standard deviation is presented in the equation (3.1)

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{3.1}$$

Where:

σ = Standard Deviation

x_i = each value in the dataset

μ = mean of all values in dataset

N= number of values in dataset

An image with a low standard deviation has pixel values that are closely gathered around the mean value. As a result, the image is more uniform and the pixel values are less variable. In other words, a low standard deviation indicates that the values of the pixels in the image are generally similar to one another and that there aren't many outliers or extreme values. This can suggest that the image is generally homogeneous or that there is little noise visible. A high standard deviation in image processing indicates that there is a lot of variation or dispersion in the pixel values around the mean pixel value. This suggests that there is a lot of noise or contrast in the image. In general, images with a lot of detail, sharp edges, or textures frequently have a large standard deviation. High standard deviation can be utilised to modify image processing algorithms like noise reduction or image enhancement. The following are the general steps to calculate the standard deviation of a set of numbers, including the pixel values in an image:

1. Calculate the mean: Add up all the values and divide by the total number of values. This will give you the mean or average value.
2. Calculate the difference between each value and the mean: Subtract the mean from each value in the set.
3. Square the differences: Square each difference obtained in step 2.
4. Calculate the mean of the squared differences: Add up all the squared differences and divide by the total number of values. This will give you the mean of the squared differences.
5. Calculate the square root of the mean of squared differences: Take the square root of the mean of the squared differences to obtain the standard deviation.

3.3.2 Mean Deviation

In image processing, the mean deviation is a statistical measure used to quantify the average difference between the pixel values in an image and the mean pixel value of the image. It is a measure of the average absolute deviation of each pixel value from the mean pixel value. The mean deviation is similar to standard deviation but it considers the absolute value of the differences between each pixel value and the mean rather than squaring them. As a result, the mean deviation is less susceptible than the standard deviation to outliers and extreme values in the image. The mean deviation can be used to assess the contrast and texture of an image, among other aspects.

The existence of noise or artefacts in an image is determined by the mean deviation, which is also used to measure image quality. An image with a low mean deviation is more likely to be of higher quality, while one with a large mean deviation is more likely to have a lot of noise or artefacts. In order to change the brightness and contrast of an image, mean deviation can be utilised as a parameter in image enhancement algorithms. Image enhancement algorithms can alter the mean deviation to change the image's overall brightness and contrast for a better visual look. Denoising algorithms can employ mean deviation to reduce noise in an image. Denoising techniques can decrease noise by removing pixels that significantly vary from an image's mean value. Mathematical notation of mean deviation is presented in the equation (3.2).

Formula:

$$MD = \frac{1}{2}\Sigma_{i-1}^{n} x_i - m(x) \tag{3.2}$$

Where:

MD = Mean Deviation

m(x) = average value of the data set

n = number of data values

xi = data values in the set

A high mean deviation indicates there is a lot of variances in the image's pixel values around the mean, whereas a low mean deviation means the pixel values are more consistent and closer to the mean value. Images with a lot of contrast or texture frequently have a high

mean deviation. On the other hand, a low mean deviation is frequently linked to images that have less contrast and texture and a more uniform distribution of pixel values. The following are the general steps to calculate the mean deviation of a set of numbers, including the pixel values in an image:

1. Calculate the mean: Add up all the values and divide by the total number of values. This will give you the mean or average value.
2. Calculate the absolute difference between each value and the mean: Take the absolute value of the difference between each value in the set and the mean value obtained in step 1.
3. Calculate the mean of the absolute differences: Add up all the absolute differences obtained in step 2 and divide by the total number of values. This will give you the mean of the absolute differences.

3.3.3 Sharpness

Sharpness refers to the perceived clarity and detail of an image. A sharp image contains edges and details that are well defined, whereas a blurry image has edges that are softened or smeared. A number of variables, including camera settings, lens quality, image resolution, and post-processing methods, can influence how sharp an image is. The sharpness index (SI) is a frequently employed statistic for assessing image sharpness. Based on the edge contrast and spatial frequency content of an image, the sharpness index is computed. It is a way to assess an image's overall sharpness while accounting for both high- and low-frequency elements. The sharpness of an image can be enhanced or improved using a variety of methods. Sharpening filters, such the unsharp mask filter and the high-pass filter, and deblurring techniques, like the Wiener filter and the Lucy-Richardson algorithm, are some of the more popular ones. The sharpness of images is a key component for determining image quality in image processing applications. Sharp images are more useful for applications like microscopy, satellite imaging, and medical imaging because they are visually appealing and simpler to comprehend.

The sharpness of an image can vary depending on a number of variables. The sharpness of an image can be affected by camera settings including the aperture, shutter speed, and ISO. A tiny aperture (high f-number) can enhance the depth of field and make the entire image appear sharper, whereas a wide aperture (low f-number) can cause a shallow depth of field and

blur the background. The sharpness of an image can be significantly influenced by the lens's quality. Sharper images can be produced by a high-quality lens with good optical characteristics than by a low-quality lens. For sharp photos, precise focusing is essential. An image that is out of focus will appear unclear. When utilising a small depth of field or taking pictures up close, this is especially crucial. Images that have more detail can be sharper because of a greater image resolution. Motion blur, which reduces the clarity of an image, can be caused by any movement during the taking of the picture. An image's sharpness might vary depending on the kind and degree of post-processing used. For instance, shrinking or compressing an image might cause the sharpness and detail to be lost. The sharpness of an image can be impacted by optical aberrations such chromatic aberration, distortion, and vignetting.

Very importantly sharpness is a psychoacoustics metric that provides a numerical measure of sensation based on the number of high-frequency components in an image. Its unit is the “acum”. The definition of 1.0 acum is a narrow band noise one critical band wide centred at 1 kHz, and a level of 60 dB RMS. The values produced by the sharpness calculation are always non-negative integers. Image sharpness can be measured by the use of “rise distance” of an edge within the image. With this technique, sharpness can be determined by the distance of a pixel level between 10% to 90% of its final value is also called 10-90% rise distance and the illustration is presented in the Fig 3.10. An example of blur image and sharp image is shown in fig 3.9 (a) and 3.9 (b).

Fig 3.9 (a): Blur image

Fig 3.9 (b): Sharp image

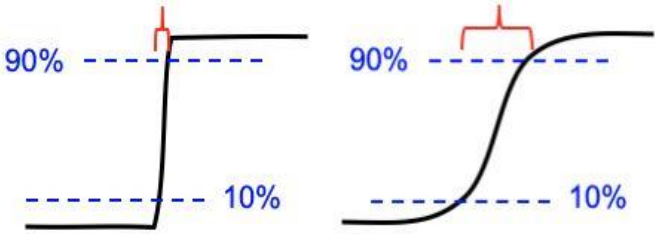


Fig 3.10: Illustration of the 10-90% rise distance on blurry and sharp edges. (Image Courtesy: imatest.com)

3.3.4 Coefficient of variance

The coefficient of variation (CV) is a statistical measure that expresses the amount of variation or dispersion in a dataset relative to its mean. The CV can be used in image processing to determine how much pixel intensities vary across an image. It shows the extend of variability in relation to the mean of the population. Depending on the image's topic of interest, the CV can change. Images with more contrast or detail will often have a higher CV than images with more evenly distributed pixel intensities. The CV can be impacted by an image's resolution because higher-resolution images can capture more detail and pixel intensity variation. An image's pixel intensities can vary more widely due to image noise, which raises the CV. A lower CV can be achieved by reducing the fluctuation of pixel intensities in an image and introducing artefacts. Different image processing methods, such as contrast enhancing, smoothing, and filtering, can have an impact on an image's CV.

The coefficient of variation is useful because the standard deviation of data must always be understood in the context of the mean of the image pixel data. The CV is determined as the ratio of standard deviation (SD) to mean. The equation of coefficient of variation (CV) is as given in equation (3.3)

Formula:

$$\mathrm{CV} = \frac{\sigma}{\mu} \tag{3.3}$$

Where:

σ = Standard Deviation

μ = Mean Deviation

A low CV value denotes a more uniform distribution of pixel intensities, while a high CV value denotes a greater degree of pixel intensity variability. The CV can be used to separate regions of an image with high contrast and variability from regions with low contrast and uniformity in image processing applications like image segmentation.

For the proposed work, the standard deviation, mean, coefficient of variance and sharpness are measured. The image property statistical features of the proposed dataset of butterfly images as well as experimental dataset are presented in the Table 3.1, Table 3.2, Table 3.3, Table 3.4, Table 3.5, Table 3.6, Table 3.7, Table 3.8.

3.4 Image Quality Assessment

The process of assessing an image's quality is known as image quality assessment (IQA). IQA is a crucial task in image processing since it allows us to assess how well various techniques for image enhancement, restoration, and compression perform. It is a crucial component of image processing since it aids in evaluating the effectiveness of different image enhancement and restoration methods. Image quality can be evaluated objectively or subjectively. Human perception is involved in subjective assessment, which is frequently employed to evaluate the visual quality of images. This is accomplished through the use of techniques like Mean Opinion Score (MOS), in which a group of people score the image quality on a scale of 1 to 5. However, subjective evaluation takes time and might not be practical for big datasets. On the other hand, objective assessment makes use of mathematical formulas to analyse an image's quality. The Full-Reference (FR), Reduced-Reference (RR), and No-Reference (NR) approaches are the most often utilised techniques for objective evaluation. While RR methods just employ a portion of the original image's information and NR methods don't require any, FR methods compare the original image with its distorted capture. Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) are a few of the often-utilised metrics for IQA. While SSIM examines the structural information of the two images, PSNR compares the signal-to-noise ratio between the original and deformed images.

3.3.1 Peak Signal to Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is a widely used image quality metric that measures the difference between a reference image and a distorted image. PSNR is used in a variety of fields, such as computer vision, image processing, and video and image compression. It is frequently employed as a benchmark metric to assess how well various image processing algorithms perform. One of the main advantages is its simplicity and ease of calculation. Since PSNR has been in use for many years, its findings are simple to understand and interpret. It is excellent for identifying even slight distortions because it is sensitive to small variations between images. It can also be used together with other metrics to provide a more thorough evaluation of image quality. In order to reduce the amount of data needed for defining an image while minimising the visual quality loss, PSNR is frequently employed in image compression. When comparing the quality of a compressed image to the original uncompressed version in various applications,

PSNR is employed. Usually, the optimum compression technique is the compression algorithm that yields the highest PSNR value.

The PSNR is expressed in decibels (dB) and is defined in equation (3.4).

$$PSNR = 10 * \log_{10} (MAX^2 / MSE) \quad (3.4)$$

Where:

MAX = maximum pixel value of the image

MSE = Mean Squared Error between the reference and distorted images

The average squared difference between each pixel value in the two images is calculated using the statistical metric known as the MSE. The MSE calculates the average size of the pixel value differences between the two images. A lower MSE suggests less distortion and greater similarity between the images.

The difference in distortion between the reference and distorted images decreases with increasing PSNR values. A completely distorted image yields a PSNR value of 0 dB while a perfect reconstruction yields an infinite PSNR value. In general, a greater PSNR value implies a lower level of distortion, whereas a lower PSNR value suggests a higher level of distortion. The image with the greater PSNR value is thought to be of higher quality when compared to the other(s), as it exhibits less distortion. Therefore, a high PSNR value is preferred because it shows that the images are highly similar and that it is challenging to see the differences between them. A low PSNR score means that the distortion between the images is obvious and that there are considerable variances between the images. This might happen when an algorithm used to compress or process images introduces considerable noise or artefacts into the image. A low PSNR value occasionally renders an image useless or makes it impossible to detect crucial details.

3.3.2 Structural Similarity Index Method (SSIM)

SSIM (Structural Similarity Index) is a method for measuring the similarity between two images. It is frequently used in image and video processing to evaluate the effectiveness of denoising algorithms, image and video compression algorithms, and other image processing methods. Luminance, contrast, and structure are three elements that SSIM considers when determining how similar two photos are to one another. The structural similarity between the

two images is calculated by comparing the local patterns of pixels in the two images. The SSIM index is a number between -1 and 1, where 1 denotes perfect similarity and -1 denotes perfect dissimilarity.

By calculating the mean of the pixel values in the two images, the SSIM approach compares the luminosity of the two images. By computing the standard deviation of the pixel values in the two images, it compares the contrast between the two images. By computing the covariance of the pixel values in the two photos, it compares the pixel structure of the two images. The formula to calculate SSIM index is presented in equation (3.4).

Formula:

$$\text{SSIM}(x,y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x + \sigma_y + C_2\right)} \tag{3.4}$$

Where:

x and y = The two images being compared.

μ = the mean of the pixel values.

σ = the standard deviation of the pixel values.

σ_{xy} = the covariance of the pixel values.

C1 and C2 = small constants that prevent division by zero.

A lower SSIM value implies a lower level of similarity or higher level of dissimilarity between the two images. A higher SSIM (Structural Similarity Index) number shows a higher level of similarity or correlation between the two images being compared. The image with the higher SSIM value is thought to be of greater quality when comparing two or more of them since it is more similar to the reference image. Therefore, a high SSIM score is preferred since it shows that the images are quite similar in terms of their structure, brightness, and contrast, making it challenging to visually distinguish between them. The results of quality assessment are presented in Table 3.1, Table 3.2, Table 3.3, Table 3.4, Table 3.5, Table 3.6, Table 3.7, and Table 3.8.

Table 3.1: Property and Quality Assessment for Monarch Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Monarch-1	52.46557	127.8909	2.43761611	2.056778	36.6652	0.9843
Monarch-2	55.47011	118.2239	2.13130834	2.978923	34.0912	0.9804
Monarch-3	61.76586	120.6941	1.95405864	2.603368	35.261	0.98126
Monarch-4	53.36007	98.11416	1.83871869	2.606842	35.1277	0.9811
Monarch-5	49.31162	112.4974	2.28135753	1.390783	39.4142	0.99185
Monarch-6	67.59601	69.94068	1.03468657	2.608398	36.2589	0.98517
Monarch-7	48.31295	67.55389	1.39825631	1.7453	37.3662	0.98946
Monarch-8	68.64032	83.38127	1.21475635	2.942942	34.6509	0.9747
Monarch-9	53.05127	92.75688	1.74843829	2.157862	36.0168	0.98335
Monarch-10	47.92259	67.28515	1.40403831	1.485397	38.0708	0.99113
Monarch-11	67.14327	127.0663	1.89246516	4.289098	32.4625	0.96975
Monarch-12	66.30639	104.0229	1.56882177	2.810954	35.1514	0.98459
Monarch-13	58.62857	76.90057	1.31165699	2.344559	36.408	0.98967
Monarch-14	55.07725	125.1155	2.27163676	1.640238	38.2025	0.99123
Monarch-15	59.9693	122.2221	2.03807762	2.410207	35.4363	0.98566

Table 3.2: Property and Quality Assessment for Red Admiral Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Red Admiral-1	50.13206	91.3377	1.82194	1.39822	38.1115	0.98644

Red Admiral-2	89.25282	101.646	1.13885	2.06778	36.8065	0.99053
Red Admiral-3	50.81059	91.6022	1.80282	2.46571	35.3895	0.97079
Red Admiral-4	69.21549	101.157	1.46148	2.42727	35.5569	0.97467
Red Admiral-5	65.57809	104.777	1.59775	2.4759	34.535	0.96967
Red Admiral-6	78.63083	119.096	1.51463	1.6029	37.6656	0.97482
Red Admiral-7	65.02092	93.1563	1.43271	1.34667	37.9836	0.98612
Red Admiral-8	65.87182	89.0686	1.35215	1.67182	37.9789	0.99162
Red Admiral-9	67.14871	73.37	1.09265	1.90233	36.8224	0.99055
Red Admiral-10	53.78811	84.3426	1.56805	1.49278	38.1928	0.98799
Red Admiral-11	57.76523	74.3873	1.28775	2.38729	36.0017	0.98397
Red Admiral-12	53.20731	80.2426	1.50811	1.77285	37.4731	0.98829
Red Admiral-13	62.0627	95.2624	1.53494	1.88608	37.0116	0.98201
Red Admiral-14	79.50805	105.64	1.32868	2.57633	35.6579	0.97637
Red Admiral-15	53.78698	70.2617	1.3063	2.37939	36.2767	0.97526

Table 3.3: Property and Quality Assessment for Buckeye Butterfly Dataset

Names	Standard Deviation	Means	Covariance	Sharpness	PSNR	SSIM
Buckeye-1	61.28802	98.7349	1.610999	4.075709	32.7546	0.96984
Buckeye-2	58.415	58.9933	1.0099004	2.370204	36.852	0.98231
Buckeye-3	40.09112	135.724	3.3853934	1.894262	36.3741	0.98581
Buckeye-4	49.71974	152.338	3.0639272	1.381739	38.3596	0.98954
Buckeye-5	50.64884	103.949	2.0523417	4.90598	32.7722	0.96092
Buckeye-6	62.11255	106.069	1.7076943	2.438014	35.2583	0.98418
Buckeye-7	44.15791	76.4192	1.7305884	1.649282	38.2844	0.98324
Buckeye-8	48.06163	101.982	2.1219095	1.90981	36.8714	0.98711
Buckeye-9	74.9851	145.802	1.9444117	4.444239	32.8472	0.9628
Buckeye-10	40.40539	74.577	1.8457181	2.056691	36.8877	0.98109
Buckeye-11	47.58112	96.4639	2.0273564	3.025513	34.5861	0.96774
Buckeye-12	73.54509	88.8484	1.2080814	1.850978	36.6465	0.98894
Buckeye-13	57.92448	75.1142	1.2967607	1.646281	37.1325	0.99065
Buckeye-14	79.65116	89.6359	1.1253556	2.113647	36.5208	0.98274
Buckeye-15	71.70899	133.486	1.861491	6.178258	30.7126	0.94299

Table 3.4: Property and Quality Assessment for Swallowtail Butterfly Dataset

Names	Standard Deviation	Means	Covariance	Sharpness	PSNR	SSIM
Swallow Tail-1	64.267736	118.9614	1.8510283	2.7241262	35.05121	0.9726973
Swallow Tail-2	62.439635	81.435099	1.3042213	2.1765289	35.115669	0.9746989

Swallow Tail-3	57.885099	98.624318	1.7037946	2.3995463	35.996027	0.9655577
Swallow Tail-4	58.429138	101.35594	1.7346814	3.7779694	33.228651	0.9653138
Swallow Tail-5	47.170677	138.08646	2.9273791	4.5170797	32.095956	0.9329839
Swallow Tail-6	61.935013	69.84727	1.127751	3.4567769	34.801382	0.9733851
Swallow Tail-7	69.920009	57.819728	0.8269411	2.2563985	36.198454	0.9817151
Swallow Tail-8	53.839596	90.428233	1.679586	2.089325	37.121691	0.9878266
Swallow Tail-9	65.046083	108.47473	1.6676597	2.2227681	37.785206	0.982182
Swallow Tail-10	62.833289	82.554189	1.3138607	2.7106374	35.362132	0.980566
Swallow Tail-11	53.349003	95.296666	1.7862877	2.2919362	35.442758	0.9757883
Swallow Tail-12	52.76473	87.150187	1.651675	2.0460002	37.633261	0.9879537
Swallow Tail-13	63.420968	65.065343	1.0259279	2.2813009	36.71862	0.9841027
Swallow Tail-14	54.050505	127.03052	2.350219	1.992569	36.503828	0.9861363
Swallow Tail-15	63.058805	97.198858	1.5414002	2.1821187	35.040914	0.9836855

Table 3.5: Property and Quality Assessment for Cabbage White Butterfly Dataset

Names	Standard Deviation	Means	Covariance	Sharpness	PSNR	SSIM
Cabbage White-1	69.77202	89.18406	1.278221	0.740479	42.67943	0.99258

Cabbage White-2	65.90608	113.1357	1.71662	3.005473	33.8536	0.964346
Cabbage White-3	86.34915	146.9572	1.701895	1.714411	37.12498	0.971175
Cabbage White-4	65.85959	150.4386	2.284232	1.962865	35.82245	0.985637
Cabbage White-5	69.07357	124.3283	1.79994	1.353836	39.50198	0.983244
Cabbage White-6	55.12781	105.6662	1.916749	1.303375	39.41557	0.979675
Cabbage White-7	90.80252	137.497	1.514242	1.95693	35.82206	0.974256
Cabbage White-8	90.7773	119.5109	1.316528	1.395574	38.29746	0.979428
Cabbage White-9	81.26175	105.829	1.302322	1.961797	36.15655	0.974518
Cabbage White-10	72.35387	124.3985	1.719307	1.6527	37.33491	0.979869
Cabbage White-11	71.21687	136.033	1.910123	1.299413	38.56849	0.989029
Cabbage White-12	77.8652	100.8531	1.295227	2.634705	34.85221	0.965591
Cabbage White-13	67.04126	119.0315	1.775497	1.637421	38.09839	0.978523
Cabbage White-14	74.01632	154.1361	2.082461	1.799362	37.35648	0.967129
Cabbage White-15	69.93969	112.238	1.604783	1.918701	35.84106	0.980589

Table 3.6: Property and Quality Assessment for Crimson Longwing Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Crimson Longwing-1	67.08203	65.36986	0.974477	1.693456	38.31153	0.981932
Crimson Longwing-2	49.51482	80.13933	1.618492	2.041199	37.02957	0.984672
Crimson Longwing-3	59.00738	90.10263	1.526972	1.943324	36.65671	0.976219
Crimson Longwing-4	30.02521	60.59263	2.018059	0.671595	42.8308	0.989041
Crimson Longwing-5	51.34794	86.29408	1.680575	1.046651	40.33315	0.990843
Crimson Longwing-6	54.48031	74.02832	1.358809	0.974813	41.71502	0.989822
Crimson Longwing-7	77.39719	129.7961	1.677013	1.315689	39.40057	0.991031
Crimson Longwing-8	42.8808	73.02586	1.702997	1.006856	40.77027	0.987886
Crimson Longwing-9	71.2031	95.89227	1.346743	1.676295	37.36633	0.989277
Crimson Longwing-10	53.79986	52.71874	0.979905	1.164719	40.30847	0.987759
Crimson Longwing-11	39.20797	52.96024	1.350752	0.961919	40.33803	0.989096
Crimson Longwing-12	75.80214	84.97509	1.121012	1.188924	40.42405	0.989786
Crimson Longwing-13	61.22389	96.83179	1.581601	1.295425	39.27503	0.991305

Crimson Longwing-14	45.20219	121.4243	2.686248	0.986328	40.95697	0.991633
Crimson Longwing-15	52.21311	79.95512	1.531323	1.145716	40.56736	0.989639

Table 3.7: Property and Quality Assessment for Blue Mormon Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Blue Mormon-1	56.44507	115.6442	2.048792	3.454412	33.55999	0.974904
Blue Mormon-2	54.70631	116.8685	2.136289	3.681081	33.14986	0.968502
Blue Mormon-3	55.55871	115.207	2.073608	3.551463	33.52826	0.972612
Blue Mormon-4	55.32229	114.5272	2.070182	3.521596	33.54675	0.972445
Blue Mormon-5	56.95863	122.5858	2.152191	4.6421	31.87192	0.95966
Blue Mormon-6	56.02276	119.4998	2.133057	4.308172	32.23188	0.962485
Blue Mormon-7	55.82598	118.3175	2.119399	4.164703	32.41756	0.964496
Blue Mormon-8	59.22947	115.3707	1.94786	4.086222	32.75973	0.970036
Blue Mormon-9	54.43833	116.5197	2.140398	3.431371	33.58556	0.972376
Blue Mormon-10	55.86802	118.2132	2.115938	4.216431	32.34949	0.96361
Blue Mormon-11	56.45064	117.112	2.074591	4.27682	32.30959	0.963101

Blue Mormon-12	57.54727	116.1316	2.018021	4.192886	32.48021	0.964156
Blue Mormon-13	56.86437	116.8939	2.055662	4.283279	32.32784	0.96344
Blue Mormon-14	55.95282	115.3649	2.061824	3.460861	33.58438	0.973628
Blue Mormon-15	57.90353	114.8613	1.983666	3.537532	33.5638	0.973628

Table 3.8: Property and Quality Assessment for Common Rose Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Common Rose-1	52.6728	113.6564	2.157781	4.4773865	32.02511	0.964805
Common Rose-2	50.13621	117.6093	2.3457965	3.9774272	32.60685	0.968372
Common Rose-3	49.57497	117.6669	2.3735134	3.8577983	32.75031	0.969513
Common Rose-4	49.54658	117.6765	2.3750677	3.7735087	32.85601	0.970505
Common Rose-5	49.53638	118.2118	2.3863635	3.7128245	32.94931	0.971131
Common Rose-6	49.19549	118.2909	2.4045071	3.7816111	32.83603	0.970319
Common Rose-7	49.62055	118.1478	2.3810262	3.8405457	32.73792	0.969968
Common Rose-8	52.64496	113.6933	2.1596232	4.4336243	32.09637	0.964708
Common Rose-9	51.95048	114.1575	2.1974282	4.3969727	32.11635	0.964875

Common Rose-10	52.27592	117.1564	2.2411161	4.2533468	32.28716	0.96657
Common Rose-11	52.33121	117.0136	2.2360203	4.2070262	32.33619	0.967084
Common Rose-12	51.35549	116.9454	2.2771742	4.1562551	32.4154	0.967288
Common Rose-13	50.78432	117.34	2.310555	4.0683085	32.51931	0.967845
Common Rose-14	50.25864	117.8166	2.3442061	3.9768931	32.59333	0.968264
Common Rose-15	50.29924	117.8191	2.3423626	3.9602559	32.62536	0.968722

3.4 Summary

In this chapter, a description of the butterfly species taken into consideration is presented. Property assessment techniques is stated and their property assessment statistical values such as standard deviation, mean, variance and sharpness are tabulated. This chapter emphasises the importance of quality assessment techniques such as SSIM and PSNR while assessing the quality factor in image samples.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

In this chapter, the system design and implementation are discussed in detail. A framework is designed considering traditional models involving Gray-level Co-occurrence matrix and Local binary patterns, and Deep Learning model involving AlexNet.

4.1 Dataset Collection

The process of gathering and assembling data from multiple sources into a structured and organised manner is known as dataset collection. In order to make the data useable for analysis or modelling, this process involves locating the data's sources, retrieving the data, and then cleaning and formatting the data. Datasets can be gathered using a variety of techniques. It is crucial to remember that dataset gathering should be done ethically and responsibly, taking into account problems like data protection and informed consent. The experimental dataset is obtained from a reputed research work. Further, efforts are made to capture butterfly images from surrounding environments. Few dataset samples have been collected from Butterfly Park, Bannerghatta, Bengaluru. The number of images proposed for each butterfly specie is 100. Overall, the dataset consists of 8 different species of butterflies, totalling 800 images.

4.2 Pre-processing

An important step in image processing is image pre-processing, which involves carrying out a number of operations on raw image information to improve its quality and get it ready for future analysis. Enhancing an image's suitability for particular image processing tasks, including object recognition, feature extraction, or image segmentation, is the main objective of image pre-processing. Prior to using advanced approaches, it aims to enhance the features of interest, decrease noise, and improve the image quality. Image acquisition is the first stage of the image pre-processing process. Image acquisition refers to the process of capturing images with cameras, scanners, or other imaging devices. For precise image processing, the acquired image quality is essential. To capture high-quality photographs, it is crucial to modify the camera's settings, including exposure, shutter speed, and ISO. Image scaling and cropping follow next. Resizing an image involves increasing or decreasing its size while keeping its aspect ratio. On the other hand, cropping involves eliminating undesirable portions of the

image. Enhancing the image is the next step. Image enhancement involves improving an image's aesthetic appeal by changing its brightness, contrast, and colour balance. It seeks to improve the attraction and understanding of visuals. Overall, digital image pre-processing involves obtaining ready digital images for additional analysis and manipulation. It involves a number of phases, including the acquisition, scaling, cropping, filtering, and augmentation of the image. The accuracy and efficiency of image processing applications can be greatly increased by using appropriate image pre-processing methods. In this project, image resizing, cropping and image enhancement techniques have been employed as a part of image pre-processing step in order to achieve high classification accuracy and hence improve the performance of the model.

4.2.1 Resizing and Cropping

Resizing and cropping are two essential techniques in image pre-processing used to adjust the size and composition of digital images. To change the size and composition of digital photos, resizing and cropping are two crucial image pre-processing procedures. Resizing an image involves modifying its dimensions while preserving its aspect ratio. It can be used to enlarge the image for printing or display reasons or to resize the image to save storage space. Maintaining an image's aspect ratio when scaling is crucial to prevent distortion. Image cropping, on the other hand, involves removing parts of the image that are not relevant or distracting. Cropping can be used to adjust the composition of an image, eliminate distracting backgrounds, concentrate on particular subjects or areas of interest. Additionally, cropping can minimise the size of an image, making it more manageable. To prevent image distortion while cropping, it's crucial to keep the image's aspect ratio intact. By utilising a cropping tool to select the area of the image to be cropped or by defining the cropped image's size, cropping can be done manually. The latter approach is especially helpful when it's necessary to crop multiple photographs to the same size. Resizing and cropping are commonly used together in image pre-processing to prepare images for further analysis or manipulation. In this work, the dataset consisting of butterfly image samples have been resized and cropped to match 256*256 pixels.

4.4.2 Image Enhancement

Image enhancement is a method of image processing that involves raising the visual appeal and readability of digital images by enhancing their visual quality. The brightness, contrast, and colour balance of an image can be changed, as well as its noise level and blurriness.

Techniques for image enhancement can be performed either manually or automatically. Using manual procedures, the enhancement algorithms' parameters, such as contrast or colour balance, are adjusted to get the desired effects. Deep neural networks and other machine learning algorithms are used by automatic systems to discover the optimal image enhancement parameters. image negativity, contrast stretching, power law transformation, noise reduction, sharpening, and filtering are a few of the techniques used to improve images. Depending on the particular application and the requirements of the user, these techniques can be used manually or automatically.

Contrast stretching, image negativity, and power law transform are all image enhancement techniques used to adjust the contrast and brightness of digital images. Contrast stretching is a simple and commonly used image enhancement technique that stretches the range of pixel values in an image to improve its contrast. Stretching the range of pixel values in an image to increase contrast is a straightforward and often used image enhancing technique. It functions by transforming the original pixel values into a new range of values, often between 0 and 255, where the image's minimum and maximum pixel values are transferred to, respectively, 0 and 255. As a result, the image's histogram is improved and the visual contrast is enhanced. Image negativity is an image enhancement technique that inverts the pixel values of an image, effectively creating a negative of the original image. By subtracting the original pixel values from the highest pixel value in the image, the negative transform can be applied. The power law transform is an image enhancement technique that is often used to adjust the brightness and contrast of images. In order to darken or brighten an image, the pixel values are raised to a power value, which is normally between 0 and 1. This transformation is especially helpful for making details more visible in low contrast or images with a limited dynamic range. All the three techniques can be used in combination to achieve maximum for accurate classification of butterfly species.

The proposed work involves three image enhancement techniques, namely, Image Negativity, Power Law Transform and Contrast Stretching. Image quality and property assessment have been carried out for the pre-processed images. A comparative analysis is acted upon the raw dataset samples and pre-processed samples. It is observed that after enhancing the images with contrast stretching techniques, there are significant developments in accuracy of the model. The results of image property analysis and quality analysis are tabulated in the Table 4.1, Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6, Table 4.7, Table 4.8.

Table 4.1: Property and Quality Assessment for Pre-processed Monarch Butterfly Dataset

Names	Standard Deviation	Means	Covariance	Sharpness	PSNR	SSIM
Monarch-1	28.87324	215.6819	7.469959	1.196457	38.0991	0.979326
Monarch-2	22.72124	220.0523	9.684873	1.321396	37.80948	0.969103
Monarch-3	25.66068	208.638	8.130649	1.498413	37.68647	0.963819
Monarch-4	24.99521	212.0168	8.482298	1.358582	37.40199	0.965623
Monarch-5	26.16066	215.2316	8.2273	0.825363	39.35015	0.986532
Monarch-6	41.94885	184.0135	4.386615	2.079422	35.92204	0.970828
Monarch-7	27.30908	185.5707	6.795203	1.281494	38.0776	0.971968
Monarch-8	36.34108	195.2725	5.373325	1.979477	35.81504	0.9593
Monarch-9	27.74252	199.6434	7.196299	1.228058	38.11135	0.970926
Monarch-10	26.37881	186.0329	7.05236	1.018356	39.40818	0.977136
Monarch-11	32.94003	217.557	6.604639	2.013565	34.87244	0.9586
Monarch-12	30.81022	214.2493	6.953837	1.622879	36.30272	0.970543
Monarch-13	40.52171	168.735	4.164065	2.223938	34.85267	0.965152
Monarch-14	28.27512	217.1581	7.680185	1.012436	39.18139	0.982002
Monarch-15	38.35015	191.9407	5.004953	1.802414	36.16214	0.970261

Table 4.2: Property and Quality Assessment for Pre-processed Red Admiral Butterfly Dataset

Names	Standard Deviation	Means	Covariance	Sharpness	PSNR	SSIM
Red Admiral-1	24.89584	205.8829	8.269773	0.8529053	39.83346	0.978448
Red Admiral-2	46.97995	192.4684	4.09682	1.3080597	37.69431	0.967194

Red Admiral-3	34.05041	183.6415	5.393223	2.1416779	35.40144	0.951676
Red Admiral-4	43.95387	199.539	4.539738	2.0775146	35.45771	0.960294
Red Admiral-5	37.54654	192.5282	5.127722	1.6928558	36.06056	0.95378
Red Admiral-6	40.19787	206.6043	5.139683	0.9366913	39.20789	0.974936
Red Admiral-7	42.32707	176.4699	4.169197	1.1603699	38.21108	0.972298
Red Admiral-8	32.4112	209.2774	6.456947	0.9206543	39.81964	0.978648
Red Admiral-9	39.28963	193.331	4.920662	1.2249603	38.32887	0.973164
Red Admiral-10	33.97185	199.6573	5.877139	0.9595032	39.8078	0.975239
Red Admiral-11	31.49887	197.8483	6.281124	1.4209595	38.10378	0.963462
Red Admiral-12	32.02551	200.5774	6.263051	1.2259216	37.99913	0.973093
Red Admiral-13	42.64919	199.3585	4.674379	1.2137909	38.44495	0.970129
Red Admiral-14	54.02556	194.0672	3.592136	1.720047	36.62153	0.963854
Red Admiral-15	40.90244	187.0497	4.573071	2.1647491	35.51978	0.952517

Table 4.3: Property and Quality Assessment for Pre-processed Buckeye Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Buckeye-1	34.7396	198.4276	5.711855	2.187439	34.95683	0.957047
Buckeye-2	56.11479	169.3549	3.018009	2.357697	35.04982	0.958836

Buckeye-3	18.55038	222.897	12.01577	0.862701	39.3386	0.978233
Buckeye-4	19.32347	229.6981	11.887	0.58287	41.76309	0.986038
Buckeye-5	42.62381	206.1309	4.836052	3.130554	34.12903	0.945641
Buckeye-6	37.55313	199.5419	5.313588	1.370804	37.6309	0.970949
Buckeye-7	24.95892	204.2735	8.184388	1.166748	39.34902	0.971771
Buckeye-8	19.67927	215.4555	10.94835	0.841812	40.78443	0.978759
Buckeye-9	50.77078	216.7049	4.2683	2.190414	35.3485	0.955053
Buckeye-10	29.67095	199.7204	6.731176	1.439926	37.76206	0.969579
Buckeye-11	25.08183	207.6874	8.280393	1.669342	37.20409	0.962156
Buckeye-12	36.93697	193.3146	5.233634	1.017426	38.54664	0.975433
Buckeye-13	43.23829	190.1573	4.397891	0.84845	40.18141	0.978252
Buckeye-14	40.15632	195.3355	4.864377	1.369308	37.62134	0.969566
Buckeye-15	36.91462	216.2505	5.858127	3.027878	33.26307	0.93473

Table 4.4: Property and Quality Assessment for Pre-processed Swallowtail Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Swallow Tail-1	46.97724	206.3447	4.3924408	2.140961	35.36537	0.963407
Swallow Tail-2	44.92706	164.5316	3.66219432	1.910492	35.18572	0.960429
Swallow Tail-3	35.64152	201.7738	5.66120126	1.450394	37.50774	0.9646
Swallow Tail-4	32.33281	201.3013	6.22591323	2.098434	35.73961	0.95194

Swallow Tail-5	21.45923	224.6312	10.4678139	1.559052	37.35341	0.950055
Swallow Tail-6	41.00139	190.6477	4.64978678	2.530243	34.58662	0.947021
Swallow Tail-7	46.63155	172.9675	3.70923839	1.608185	36.58023	0.968915
Swallow Tail-8	31.65519	195.8995	6.18854481	1.334702	37.74539	0.968725
Swallow Tail-9	66.83679	195.4163	2.923783	2.254303	35.92138	0.973883
Swallow Tail-10	27.84283	206.0205	7.39940985	1.525742	36.55597	0.962122
Swallow Tail-11	32.71437	187.4877	5.7310502	1.534058	36.50839	0.963614
Swallow Tail-12	29.44029	211.0968	7.17033715	1.223572	37.83541	0.972381
Swallow Tail-13	39.75023	187.3511	4.7132087	1.552002	36.45453	0.967156
Swallow Tail-14	24.91775	222.704	8.93756601	1.08197	37.33889	0.976303
Swallow Tail-15	29.55	199.6569	6.75657795	1.267059	37.37654	0.964823

Table 4.5: Property and Quality Assessment for Pre-processed Cabbage White Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Cabbage White-1	62.114	142.765	2.298434	0.65094	42.20492	0.985664

Cabbage White-2	42.62	196.88	4.619427	1.261185	38.29813	0.9597
Cabbage White-3	50.24073	204.9015	4.078394	0.955551	39.52285	0.973414
Cabbage White-4	32.76393	215.9158	6.590044	0.91861	40.23442	0.974935
Cabbage White-5	40.13666	198.5197	4.946095	0.846664	40.99876	0.979633
Cabbage White-6	32.91638	209.2784	6.35788	0.960922	40.2042	0.970233
Cabbage White-7	61.51455	204.6619	3.327048	0.767731	40.47871	0.978655
Cabbage White-8	44.17936	205.0674	4.6417	0.768555	40.30857	0.983215
Cabbage White-9	48.59233	192.4821	3.961162	0.995651	39.59645	0.973472
Cabbage White-10	37.14247	208.6598	5.617824	0.876297	40.51594	0.977819
Cabbage White-11	21.88842	223.8097	10.22503	0.393066	45.38657	0.987284
Cabbage White-12	56.89415	190.7903	3.353426	1.258011	38.4642	0.967297
Cabbage White-13	61.28852	171.2269	2.793784	1.551254	37.74067	0.960626
Cabbage White-14	27.83054	226.0757	8.123296	0.655441	41.41536	0.982446
Cabbage White-15	34.18676	202.2651	5.916475	0.771042	40.03265	0.974684

Table 4.6: Property and Quality Assessment for Pre-processed Crimson Longwing Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Crimson Longwing-1	31.5449	195.7173	6.2044033	1.21199	38.29539	0.970897
Crimson Longwing-2	27.44646	195.5221	7.123763	1.477859	37.72154	0.958703
Crimson Longwing-3	34.84456	193.2977	5.5474278	1.220428	38.37377	0.969091
Crimson Longwing-4	26.89526	186.4039	6.9307347	0.676361	41.26132	0.985182
Crimson Longwing-5	21.02265	204.0224	9.7048841	0.659058	41.72756	0.982528
Crimson Longwing-6	45.28789	191.0212	4.2179307	1.213593	38.45791	0.977024
Crimson Longwing-7	46.10893	215.8735	4.6818153	0.934357	40.1994	0.983953
Crimson Longwing-8	27.03607	192.4414	7.117952	0.734634	41.35254	0.980482
Crimson Longwing-9	48.24212	187.1934	3.8802906	1.181381	38.85241	0.969913
Crimson Longwing-10	42.42451	171.4266	4.0407445	1.215546	38.19224	0.974783
Crimson Longwing-11	22.62971	182.9483	8.0844276	0.724289	40.78249	0.98202
Crimson Longwing-12	42.53329	198.578	4.6687669	0.87915	39.73297	0.980266
Crimson Longwing-13	41.72621	192.2314	4.6069694	0.892441	40.46906	0.978953

Crimson Longwing-14	20.2917	221.0761	10.894903	0.40947	44.41369	0.98904
Crimson Longwing-15	35.74872	196.8225	5.5057227	0.976181	39.5631	0.979184

Table 4.7: Property and Quality Assessment for Pre-processed Blue Mormon Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Blue Mormon-1	23.62447	219.9208	9.30902552	1.437424	37.79423	0.960015
Blue Mormon-2	25.54315	218.6332	8.5593666	1.549927	37.40327	0.958451
Blue Mormon-3	26.04674	218.7181	8.39713801	1.610107	37.08238	0.957997
Blue Mormon-4	25.99691	218.3966	8.40086761	1.57811	37.26969	0.958722
Blue Mormon-5	27.12626	220.2087	8.11791623	1.899918	35.83658	0.947815
Blue Mormon-6	27.08316	219.1433	8.09149477	1.809631	36.2228	0.950492
Blue Mormon-7	26.97741	219.2132	8.12580563	1.78093	36.44755	0.951063
Blue Mormon-8	29.98992	213.7579	7.12766023	2.073242	35.54969	0.94861
Blue Mormon-9	24.86661	219.5744	8.83008888	1.45401	37.79583	0.95965
Blue Mormon-10	26.92355	218.8769	8.12956998	1.776108	36.39944	0.95151
Blue Mormon-11	27.73421	218.6456	7.88360683	1.866119	36.09633	0.949556

Blue Mormon-12	28.26782	217.8889	7.70802052	1.91806	35.9564	0.949442
Blue Mormon-13	27.79139	218.2636	7.85363962	1.908081	36.03251	0.9493
Blue Mormon-14	24.12834	219.3893	9.09259569	1.471695	37.69125	0.959177
Blue Mormon-15	27.94998	218.3172	7.81099619	1.608093	37.14614	0.95837

Table 4.8: Property and Quality Assessment for Pre-processed Common Rose Butterfly Dataset

Names	**Standard Deviation**	**Means**	**Covariance**	**Sharpness**	**PSNR**	**SSIM**
Common Rose-1	23.77464	217.6347	9.15407	1.723862	37.13407	0.947922
Common Rose-2	20.99249	220.0043	10.48014	1.443649	38.31686	0.95452
Common Rose-3	21.05319	220.2878	10.46339	1.392075	38.52859	0.956764
Common Rose-4	21.45051	220.1861	10.26484	1.390808	38.43274	0.957699
Common Rose-5	21.29927	220.4394	10.34962	1.353912	38.5687	0.958808
Common Rose-6	20.92205	220.7037	10.54885	1.373978	38.48824	0.957708
Common Rose-7	20.89963	220.3812	10.54474	1.37999	38.4688	0.956899
Common Rose-8	25.04218	217.4248	8.682342	1.738663	37.25298	0.948091
Common Rose-9	23.81221	217.9201	9.151611	1.693954	37.29324	0.9482

Common Rose-10	22.6676	219.3395	9.676343	1.584549	37.86635	0.951266
Common Rose-11	22.75922	219.3916	9.639682	1.573807	37.90262	0.95122
Common Rose-12	21.83702	219.5558	10.05429	1.520401	38.04493	0.952426
Common Rose-13	21.56211	219.7804	10.1929	1.477478	38.25576	0.953619
Common Rose-14	21.38857	220.2203	10.29617	1.449539	38.36607	0.954552
Common Rose-15	21.06726	220.2519	10.4547	1.442993	38.27894	0.9545

4.3 System Architecture

The proposed system framework involves two different approaches for the classification of butterfly images. The main methodology of the model is shown in Fig 4.1.

The main methodology employed in the proposed work contains two path, traditional approach and deep learning approach. Traditional approach uses Gray Level Co-occurrence matrix and Local Binary Pattern for feature extraction. The results of feature extraction have been applied to KNN classifier to classify the butterfly species with precision. KNN classifier considered 70% images for training and 30% images for testing. Classification is achieved by calculating features of required image and then using kNN classifier to classify the image base on the training data obtained during Machine learning model training. Deep learning approach considers a small number of species, using a pre-trained AlexNet model with 800 images for butterfly species identification can produce good results. By substituting the final layer with a fully linked layer and a SoftMax output layer, the model can be adjusted. On the butterfly images, the fully connected layer may be trained, and the SoftMax layer can translate the outputs into probabilities. The model may reach excellent accuracy for differentiating between the eight species with 800 images.

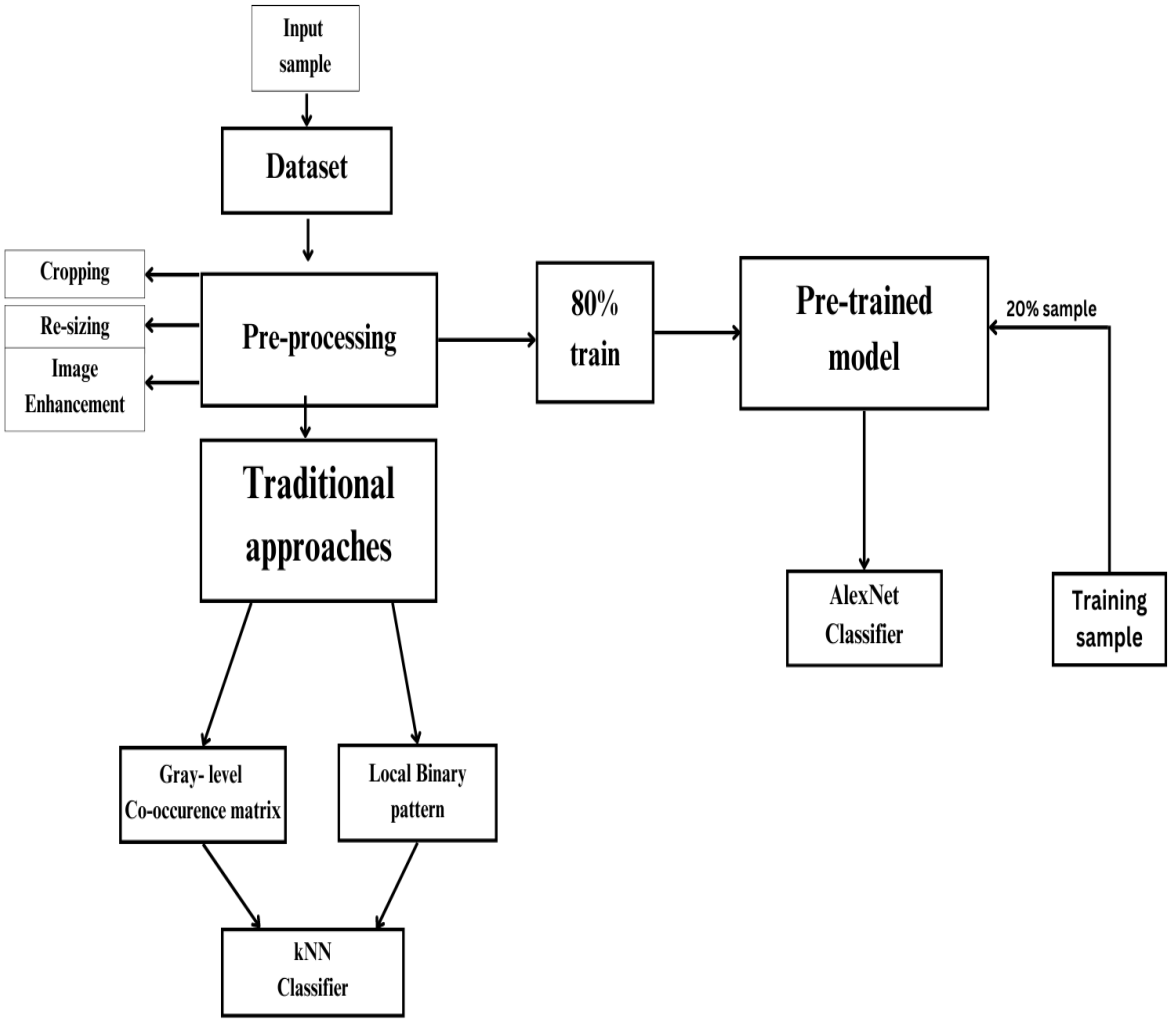


Fig 4.1: Flow chart for System design

4.4 Feature Extraction

Feature extraction is one of the main tasks associated with image processing. The technique of locating and extracting particular features or characteristics from an image that are relevant to a given application is known as feature extraction. For tasks like object detection, image classification, and image segmentation, these attributes can be used. Depending on the application, many kinds of features can be derived from an image. These comprise information about colour, texture, shape, and space. While texture features describe the patterns and

structures inside an inside, colour features define how the colours are distributed in an image. While spatial information defines the positioning and direction of items in an image, shape features represent the geometric characteristics of objects in an image.

4.4.1 Gray level Co-occurrence Matrix (GLCM)

GLCM (Gray Level Co-occurrence Matrix) is a texture analysis technique used to quantify the spatial relationship between pairs of pixels in an image. It is a matrix that depicts how pixel pair distribution is shared across an image. A window slides over an image to count the occurrences of pairs of pixels with specific grey level values within a given range and direction. This is how the GLCM is calculated. As a result, a square matrix is produced, with each element denoting the frequency of two different grey level values appearing in the image at a certain distance and angle. It is widely used in computer vision and image processing for texture classification, segmentation, and analysis. Specifically, it counts the number of times that a particular grey-level value (i) occurs at a given distance (d) and angle (θ) relative to another grey-level value (j) in the image. Contrast, energy, entropy, and homogeneity are a few of the texture features that the GLCM may be used to extract. These features are helpful in a variety of applications, including image classification, segmentation, and identification. Once the GLCM matrix is calculated, various statistical measures can be computed to describe the texture of the image. Four main statistical features are considered in the proposed work. They are:

1. Contrast: The difference in pixel intensity levels between adjacent pixels in the image is measured by GLCM contrast. The Grey Level Co-occurrence Matrix (GLCM), a statistical representation of the joint distribution of pixel pairings in the image, is used to calculate it. The total of the squared differences between pixel pairs in the GLCM matrix is used to calculate the contrast feature. While low contrast values show that adjacent pixels have comparable intensities, high contrast values show that they have dramatically different intensities.
2. Energy: The uniformity of pixel intensities in an image is determined by the GLCM energy. It is one of the textural elements that the Grey Level Co-occurrence Matrix (GLCM) allows for the extraction of. By adding up the squared values of each component in the matrix, the energy of the GLCM is determined. A greater energy value denotes a more uniform texture, and it measures how homogeneous the image texture is. For image classification, segmentation, and recognition tasks, GLCM energy has

been extensively used in a variety of applications, including medical imaging, remote sensing, and material research.

3. Homogeneity: Homogeneity is a texture feature extracted from the Gray Level Co-occurrence Matrix (GLCM) of an image. It evaluates how closely adjacent pixel values in an image resemble one another. Homogeneity is calculated as the sum of the product of each element of the GLCM matrix and its inverse distance. A greater homogeneity rating denotes a more consistent and smooth texture in the image.
4. Entropy: GLCM entropy is a texture feature that measures the randomness or unpredictability of pixel intensities in an image. The Grey Level Co-occurrence Matrix (GLCM) method, which examines the spatial correlations between pixel pairings in an image, is used to calculate it. The amount of information or uncertainty in an image's texture is represented by the GLCM entropy value. While a low entropy value denotes a more regular and predictable pattern, a high entropy value denotes a more complex texture with more variable and surprising patterns.

These statistical measures can be used as features to characterize the texture of the image and can be used in subsequent analysis tasks such as texture classification and segmentation. The GLCM technique is flexible and can be customized by adjusting parameters such as distance, orientation, and grey-level quantization. However, it can also be computationally expensive, especially for larger images, and may require pre-processing steps to reduce noise and enhance contrast in the image.

4.4.2 Local Binary Pattern

LBP stands for Local Binary Pattern, which is a texture descriptor used in computer vision and image analysis. LBP is a simple yet effective feature extraction method that can be used to represent the local texture of an image. The basic idea behind LBP is to encode the local structure of an image by comparing the grey levels of a central pixel to its neighbouring pixels.To compute the LBP descriptor, each pixel in the image is compared to its neighbouring pixels, and a binary code is generated based on the result of the comparison. This binary code is then used to represent the texture of the image at that pixel. By repeating this process for all pixels in the image, a complete LBP representation of the texture of the image can be obtained. LBP has many applications in computer vision, including object recognition, face detection, and texture classification. It is a popular technique because it is computationally efficient, robust to noise and illumination changes, and can be easily combined with other feature

extraction methods. The LBP descriptor is based on the concept of binary pattern coding, which involves comparing the intensity values of a central pixel with those of its neighbouring pixels in a circular or rectangular region around the pixel. A binary code is generated based on whether each neighbour's intensity is greater than or less than the central pixel's intensity. The resulting binary code is then converted to a decimal value, which represents the texture information at the centre pixel.

The LBP operator can be applied to grayscale or colour images and can be further extended to capture more complex texture information by using various types of histograms and image transformations. For example, the Local Binary Pattern Histogram (LBPH) is a popular extension of LBP that calculates the frequency distribution of the LBP codes over a region of the image and uses the histogram as a feature vector for classification. A matrix is derived involving the GLCM statistical features along with LBP characteristics for each image in the proposed dataset.

4.4.3 kNN Classifier for Traditional Butterfly Classification

Machine learning classifiers are algorithms that automatically find links and patterns in data and forecast the behaviour of fresh data. Decision trees, random forests, support vector machines (SVMs), naive Bayes, k-nearest neighbours (k-NN), and neural networks are just a few examples of the many distinct kinds of classifiers. The best classifier to employ will depend on the nature of the problem and the qualities of the data. The classifier predicts the most likely class label based on the learned associations during inference. During training, the classifier learns the relationships between the input characteristics and the output classes. Predicting the classes of new data based on patterns discovered from labelled or unlabelled data, classifiers are an effective machine learning tool overall. Careful selection and tweaking of these components are essential for attaining effective classifier performance since the choice of classifier and its hyperparameters depends on the nature of the issue, the properties of the data, and the desired performance metrics.

A non-parametric classification approach in machine learning is the k-NN algorithm. The model is merely a representation of the training dataset in this sort of instance-based learning. Based on how close the new data points are to the training samples, it categorises them. Since comparable data points are likely to belong to the same class, the algorithm makes this assumption. The k-NN algorithm's ability to handle non-linear decision boundaries and

perform effectively with a limited number of training instances is one of its benefits. The number of training examples required to generalise correctly can rise exponentially as the number of characteristics in the data increases, which is known as the "curse of dimensionality." The k-NN algorithm's sensitivity to the value of k is another crucial factor to consider. The model is sensitive to noise and outliers while k is small but becomes less sensitive to regional changes in the data when k is big, which may lead to underfitting. Therefore, it's crucial to carefully choose the value of k to prevent either overfitting or underfitting.

On the basis of their images, eight different species of butterflies are identified using the k-nearest neighbour (k-NN) technique. 100 images each species make up the dataset, totalling 800 images. The images are resized to 250x250 pixels to eliminate any noise or background. Grey Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP) are two distinct feature extraction techniques we employ to extract meaningful information from the photos. Four separate texture measures—contrast, correlation, energy, and homogeneity—are computed for GLCM feature extraction. We employ the uniform pattern operator to lower the dimensionality of the feature space prior to LBP feature extraction.

In the proposed work, the dataset is divided into training and testing sets in the ratio of 70:30. The testing set is categorised using the k-NN algorithm based on their feature vectors. After testing several values for k, it is observed that by setting k=3 highest accuracy is attained.

4.5 Introduction to Deep Learning

A branch of machine learning called "deep learning" is motivated by the structure and operation of the human brain. It includes the automatic learning of data representations using artificial neural networks, which are made up of several linked layers of nodes or neurons. Many different applications, including speech and picture identification, natural language processing, and autonomous driving, have found success using deep learning algorithms. Convolutional neural networks (CNN), one of the most extensively used varieties of deep neural networks, are frequently employed for image and video processing tasks such object recognition, detection, segmentation, and tracking. Convolution is a method used by CNNs to extract pertinent characteristics from an input picture by using a series of filters. The enormous computing demands of deep learning are one of its difficulties. It frequently takes a lot of data and processing power to train deep neural networks, which can be a bottleneck for many applications. However, larger and more complicated networks can now be trained because to

improvements in hardware and software, and the subject of deep learning is still evolving quickly.

The difficult work of identifying and categorising butterfly species can be aided by the application of deep learning. Based on their distinctive characteristics, such as wing patterns, colours, and forms, the work entails identifying and differentiating between hundreds or thousands of different butterfly species. A huge collection of labelled images of various butterfly species is required to employ deep learning for butterfly species identification and classification. This dataset should be sufficiently varied to reflect the variation in butterfly traits among several species. Transfer learning can be utilized, which entails employing a pre-trained CNN model as a starting point for our own model, to increase the model's accuracy. Once the model has been trained, we might use it to categorise fresh images of butterflies according to their species. By comparing them to the known species in the dataset and selecting the most similar match using the learnt characteristics, the model may also be used to identify undiscovered butterfly species. Overall, the identification and categorization of butterfly species using deep learning has the potential to advance our knowledge of their variety and conservation efforts.

4.5.1 AlexNet for Butterfly Specie Classification

AlexNet is a popular convolutional neural network architecture that has been widely used for image classification tasks, including the classification of different species of animals, plants, and insects. Therefore, it is a suitable choice for butterfly species identification and classification. We would require a sizable collection of labelled butterfly images to apply AlexNet for butterfly species identification and classification. The AlexNet architecture consists of five convolutional layers followed by three fully connected layers. The max-pooling layer, which decreases the spatial dimensions of the feature maps, comes after each convolutional layer. The SoftMax layer, which is the network's last layer, produces the predicted class probabilities. Rectified linear units (ReLU) are used as activation functions in these layers, which enhances the model's performance and training efficiency. With a stride of 4, the first convolutional layer uses a filter size of 11x11, while the next two use a filter size of 5x5 with a stride of 1. With a stride of 1, the last two convolutional layers have a filter size of 3x3. The final convolutional layer's output is sent into two fully linked layers, each with 4,096 and 1,000 neurons. The second fully connected layer comprises 1,000 neurons that are equivalent to the 1,000 item categories in the ImageNet dataset, while the first fully connected

layer utilises a dropout regularisation approach to prevent overfitting. The outputs of the last fully connected layer are converted into class probabilities by the SoftMax layer at the end of the network.

The images should be pre-processed by being resized to a predetermined size, like the input size for AlexNet is 227 by 227 pixels. Then, using supervised learning, the AlexNet model is trained using the labelled butterfly dataset. The model would become more adept at identifying each pattern and characteristics in the images of butterflies as it was trained. Once the model has been trained, it can be used to categorise new images of butterflies according to their species. After receiving an input image, the model would provide a probability distribution for each species of butterfly. The species suggested for that image would be the one with the highest probability. Data augmentation methods like random cropping, flipping, and rotation can be utilized to increase the variety of the training data and help the model generalise to new photos more effectively. By unfreezing some of the previous layers and retraining them on the butterfly dataset, the accuracy of the pre-trained AlexNet model can be improved. In doing so, the model would be able to pick up on more particular aspects that are unique to butterfly images.

One of the main advantages of utilizing AlexNet for the classification of butterfly species is the ability to add and alter the layers into the network as per the requirement of the user or application. The proposed system architecture for AlexNet is illustrated in Fig 4.2.

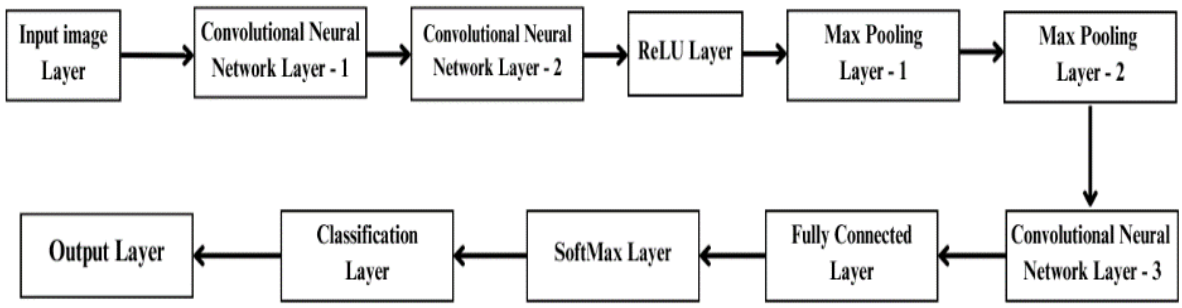


Fig 4.2: AlexNet architecture

1. Image input layer: A butterfly image is supplied into the input layer, which then resizes it to a predetermined size, such 224x224 pixels. After that, the input is normalised by dividing by the training set's standard deviation and removing the mean pixel intensity.
2. Convolutional layer -1: The first convolutional layer creates a feature map with a size of 55x55x96 by applying 96 filters of size 11x11 with a stride of 4 and a padding of 0. This layer is used to extract basic features from the source image, like corners and edges. The feature maps' dimensionality is decreased and spatial interactions between pixels are better captured due to the big kernel size and stride.
3. Convolutional layer -2: The second convolutional layer produces a feature map with a size of 27x27x256 by applying 256 filters of size 5x5 with a stride of 1 and a padding of 2. This layer is used to extract the input image's more complex features, like textures and patterns. Smaller kernel size and stride aid in preserving more information in the feature maps and capturing finer details in the image.
4. ReLU layer: After this layer, the feature map size is decreased to 27x27x96 using a ReLU activation function. ReLU's primary role in AlexNet is to add nonlinearity to the network. The network would only be able to undergo linear transformations without a non-linear activation function, which is insufficient for learning complex patterns in data. ReLU gives the network a non-linear activation function that enables it to learn more complex characteristics and representations. ReLU also helps to solve the disappearing gradient issue, which is a further benefit. It can be challenging to update the weights in deep neural networks because the gradient of the loss function with respect to the weights might become very small as it propagates through numerous layers. ReLU helps to avoid this by ensuring that the gradient is consistently non-zero for positive inputs, making the gradient more stable and propagating more easily.

In AlexNet, there are two max-pooling layers, which follow the first and second convolutional layers respectively. The function of these max-pooling layers is to down sample the feature maps produced by the convolutional layers, reducing their spatial dimensions while retaining the most salient features.

5. Max Pooling layer - 1: The first max-pooling layer features a 3x3 kernel with a 2 stride. This means that, with a stride of 2 pixels between each 3x3 region of the input feature map, the maximum value is taken and placed in the appropriate position in the output feature map. This reduces the computational cost of the succeeding layers by down

sampling the spatial dimensions of the feature map by a factor of 2 in both the horizontal and vertical directions.

6. Max Pooling layer – 2: Similar to the first max-pooling layer, AlexNet's second max-pooling layer has a kernel size of 3x3 and a stride of 2. The second convolutional layer's feature maps are used by this layer, which further shrinks their spatial dimensions while keeping the most important features.

The max-pooling layers help in expanding the receptive field of the succeeding layers, enabling them to capture more complex and abstract features by reducing the spatial dimensions of the feature maps. They also contribute to the reduction of the number of parameters in the network and the introduction of a type of translation invariance, where the network is able to recognise the same features regardless of where they are in the input image.

7. Convolutional layer -3: The third convolutional layer has 384 filters, each having a 3x3 kernel size and a 1 stride. These layers are used to extract even more intricate characteristics from the input image, including the forms and sections of objects. Even more minute features are captured in the image and more data is preserved in the feature maps due to the lower kernel size and stride.
8. Fully Connected Layer: Using the characteristics identified by the convolutional layers, the fully connected layers of AlexNet act as a classifier to generate a prediction for the input image. By mapping this feature representation to class probabilities, the fully connected layers carry out a classification task. The fully connected layer in AlexNet has 4096 neurons, which means that there are 4096 learned weights for each feature map output from the last max-pooling layer.

AlexNet can learn complex, non-linear relationships between the input image and its class because of its completely connected layers. The fully connected layers of the network are able to capture more complex data than would be achievable with a straightforward linear classifier since they are fed a high-dimensional feature representation as input.

9. SoftMax Layer: The SoftMax layer, is in responsible for generating the predicted class probabilities for the input image. A vector of probabilities with each element representing a different class in the classification task is the output of the SoftMax layer. The SoftMax layer functions by normalising the outputs from the network's last completely linked layer so that they add up to one. The output values can be understood

as probabilities because of the normalisation process. In the final fully connected layer, the SoftMax function is used to each output value to create a probability distribution over the classes. The estimated class for the input image is determined by selecting the class with the highest probability.

The AlexNet architecture's softmax layer is crucial because it enables the network to make accurate predictions for the input images. The outputs of the last fully linked layer would be arbitrary values without a clear interpretation as probabilities without the softmax layer. The network can also be trained using the cross-entropy loss function, which is frequently employed in classification tasks, because of the usage of the softmax layer. The cross-entropy loss motivates the network to provide accurate and confident predictions by measuring the discrepancy between the true probability and the anticipated probabilities.

10. Classification Layer: The final layer of AlexNet's neural network, the classification layer, is responsible for determining the class label of an input image. A class label from the dataset is associated with each of the 1000 output neurons that make up the classification layer, which is a fully connected layer. Prior to feeding it as input to the classification layer, the output of the preceding layer is first flattened into a vector. A score indicating how likely it is that the input image belongs to the specified class is produced by each neuron in the classification layer. The purpose of the classification layer is to map the high-level features learned by the convolutional layers to the class labels in the dataset.
11. Output Layer: Similar to other neural networks, AlexNet's output layer's task is to take the input and produce the model's final output. A fully connected layer of 1000 neurons, one for each class in the ImageNet dataset, makes up AlexNet's output layer.

Hyperparameters are settings made for a machine learning system that cannot be learnt from data and are made prior to training. Finding the best settings for hyperparameters may greatly increase the model's accuracy for deep learning models like AlexNet since they are crucial to the model's performance. Here are several hyperparameters that might impact AlexNet's performance,

1. Learning rate - The learning rate hyperparameter regulates how rapidly the model picks up new information during training. While a low learning rate can lead to sluggish convergence and the model being trapped in local optima, a high learning rate can cause

the model to converge fast but may also provide unstable or poor solutions. The learning rate in AlexNet typically starts at 0.01, however the dataset and model complexity may need a different setting.

2. Batch size - The number of samples utilised in each training batch is determined by this hyperparameter. The training process can be sped up by using a large batch size, but more memory is needed and the convergence process may be slowed down by noisy gradients. Smaller batches might have a quicker convergence rate but take longer to train. In AlexNet, the batch size is commonly set at 32,64,128 or 256.
3. Number of epochs - The total number of times the complete training dataset is run through the network during training is the number of epochs. The accuracy of the model can be increased by increasing the number of epochs. However, overfitting is also a possibility. The dataset will decide the best number of epochs, which may be found by testing. AlexNet usually can have 30, 50, 60 epochs for training.

AlexNet's architecture may be changed in order to enhance its performance on particular tasks, such identifying the species of butterflies. For instance, depending on how complicated the dataset is, the number of filters, filter sizes, and layer levels may be changed. On a smaller dataset of butterfly photos, transfer learning may also be used to polish a pre-trained AlexNet model.

The cross-entropy loss function, which measures the difference between the predicted class probabilities and the actual class labels, is used to optimise the output layer during training. By changing the network's weights and biases, the training process aims to reduce this loss function. After the network has been trained, fresh data can be predicted using the output layer. The network calculates the output of the final layer from an input image and predicts the class with the highest probability.

In the proposed work, accuracy for classification of butterfly species using deep learning AlexNet model is 93.5%. The obtained value of accuracy is because of the limited number of image samples for each species in the dataset. Dataset contains 100 image samples for each class in the dataset. Higher accuracy can be achieved by increasing the size of dataset and selecting the right train:test ratio of data sample. Training the model for huge number of image samples of each class increases the classification accuracy of the model. Selecting the suitable hyperparameters such as learning rate, batch size and number of epoches for the deep learning model could also improve the accuracy of the AlexNet model up to some extent.

4.6 Results

4.6.1 Feature Extraction Results

Feature extraction for the dataset is stored in feature variable of *feature1.mat* file as shown in Fig 4.3. Feature variable contains n cells where n represents number of classes considered. Each cell stores 100x8 matrix represented 8 features extracted for each image in the dataset as 100 images per class is selected. Size of matrix varies as per the number of image samples considered for a class.

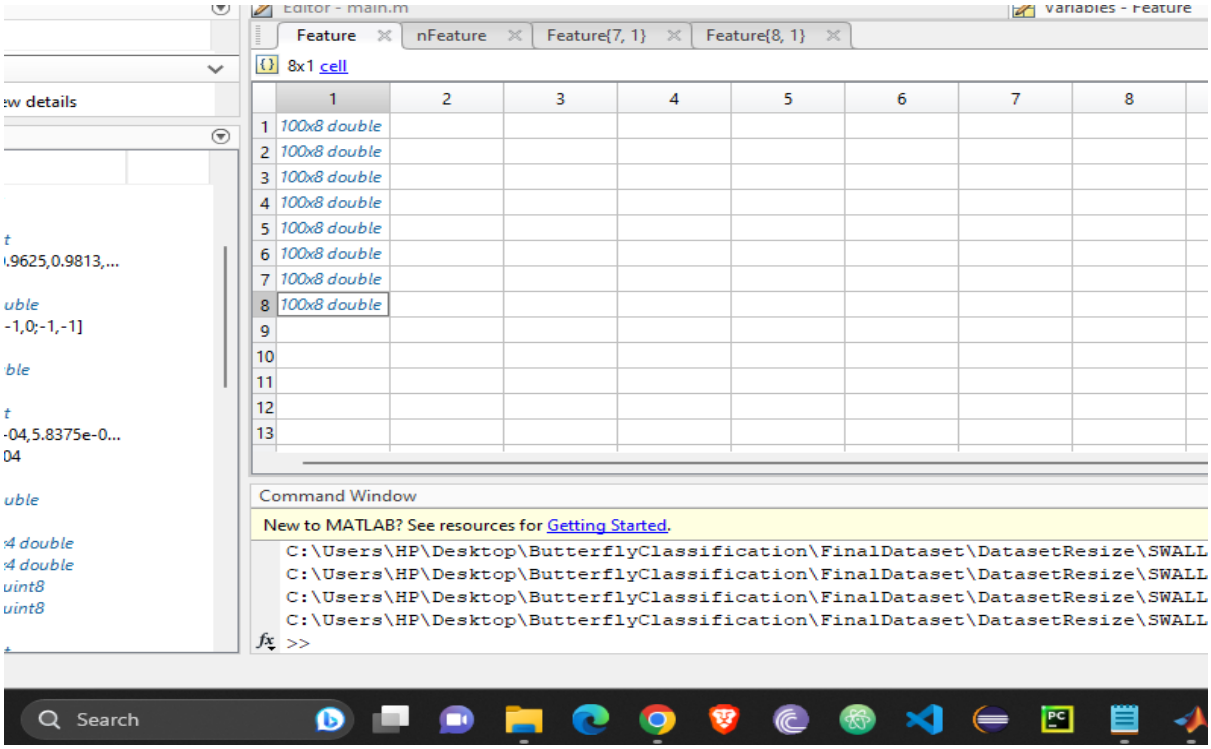
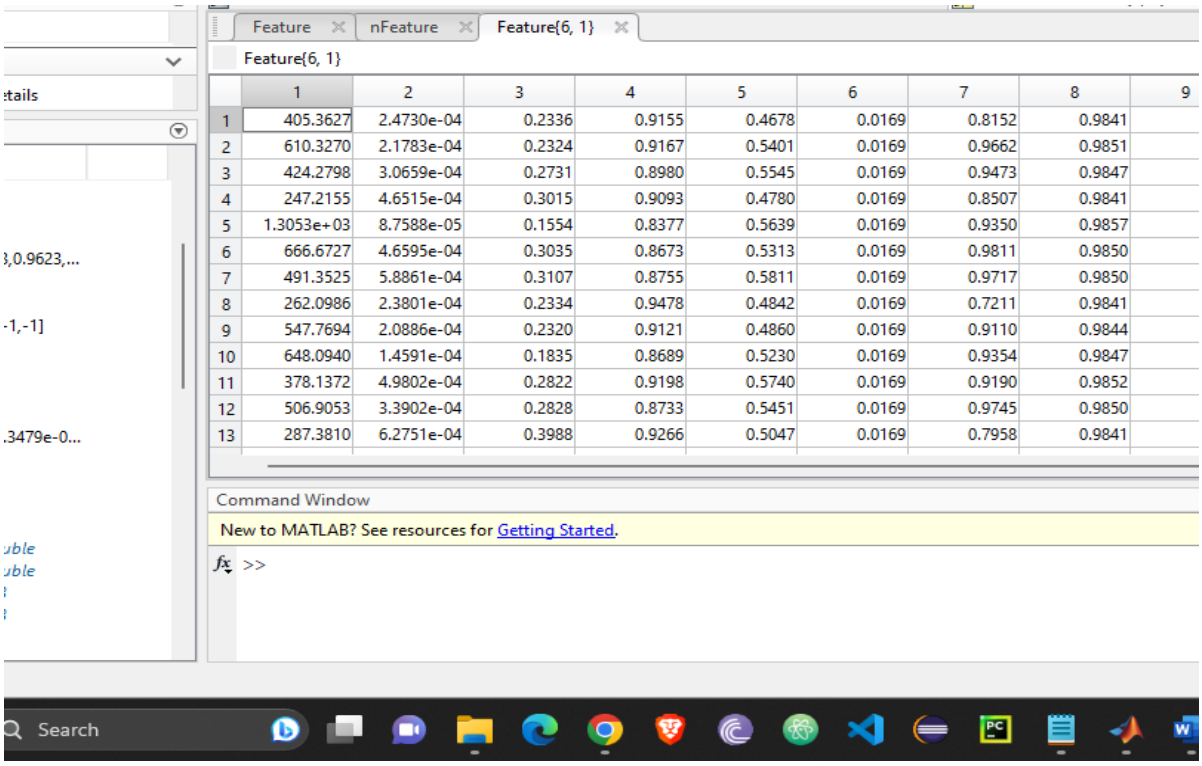


Fig 4.3: Feature Extraction

First four columns in each row of a matrix represents GLCM features such as contrast, energy, correlation and homogeneity respectively for a particular image sample. Column number 5 to 8 represents Local Binary Pattern features such as contrast, energy, correlation and homogeneity respectively for the same image sample. Each image sample feature is stored in a row of the matrix of respective class. Each image has 8 features such as GLCM contrast, GLCM energy, GLCM correlation, GLCM homogeneity, LBP contrast, LBP energy, LBP correlation, LBP homogeneity in the respective order. Machine learning model is trained on the basis of features extracted from the dataset samples which are stored in feature variable.

The results of feature extraction for all the image samples of 8 butterfly species are shown in Fig 4.4, Fig 4.5, Fig 4.6, Fig 4.7, Fig 4.8, Fig 4.9, Fig 4.10, Fig 4.11.



Feature | nFeature | Feature{6, 1}

Feature{6, 1}

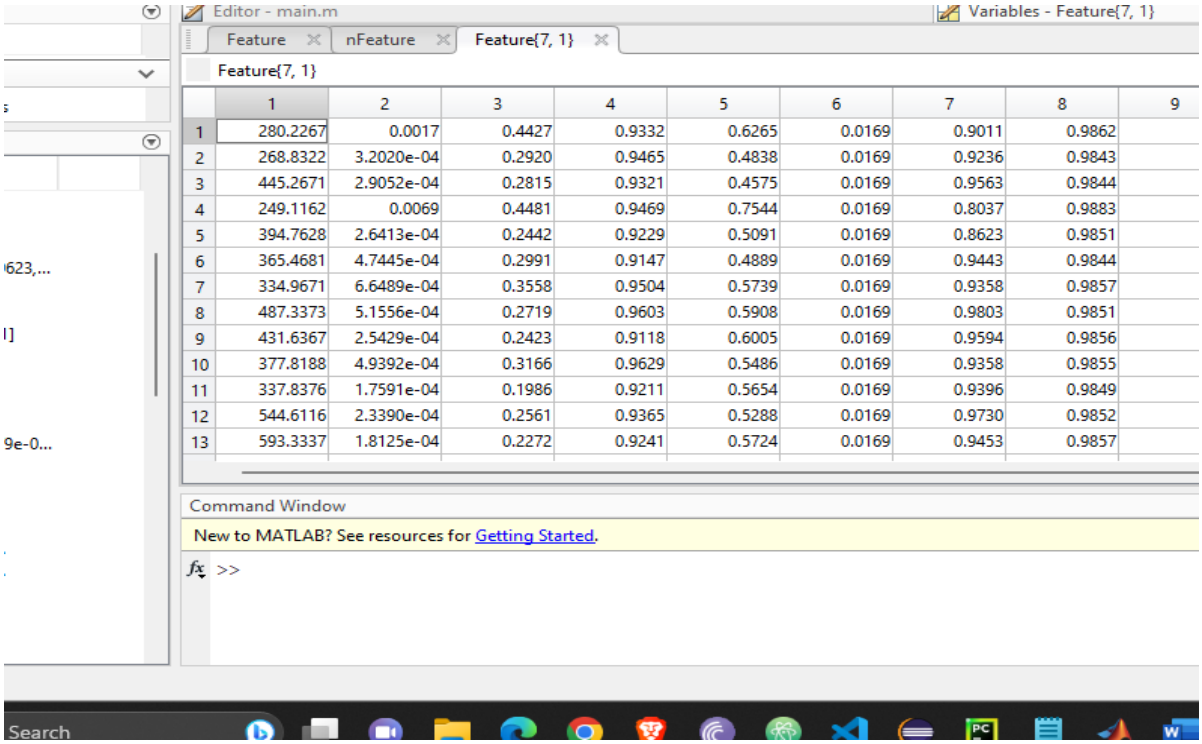
	1	2	3	4	5	6	7	8	9
1	405.3627	2.4730e-04	0.2336	0.9155	0.4678	0.0169	0.8152	0.9841	
2	610.3270	2.1783e-04	0.2324	0.9167	0.5401	0.0169	0.9662	0.9851	
3	424.2798	3.0659e-04	0.2731	0.8980	0.5545	0.0169	0.9473	0.9847	
4	247.2155	4.6515e-04	0.3015	0.9093	0.4780	0.0169	0.8507	0.9841	
5	1.3053e+03	8.7588e-05	0.1554	0.8377	0.5639	0.0169	0.9350	0.9857	
6	666.6727	4.6595e-04	0.3035	0.8673	0.5313	0.0169	0.9811	0.9850	
7	491.3525	5.8861e-04	0.3107	0.8755	0.5811	0.0169	0.9717	0.9850	
8	262.0986	2.3801e-04	0.2334	0.9478	0.4842	0.0169	0.7211	0.9841	
9	547.7694	2.0886e-04	0.2320	0.9121	0.4860	0.0169	0.9110	0.9844	
10	648.0940	1.4591e-04	0.1835	0.8689	0.5230	0.0169	0.9354	0.9847	
11	378.1372	4.9802e-04	0.2822	0.9198	0.5740	0.0169	0.9190	0.9852	
12	506.9053	3.3902e-04	0.2828	0.8733	0.5451	0.0169	0.9745	0.9850	
13	287.3810	6.2751e-04	0.3988	0.9266	0.5047	0.0169	0.7958	0.9841	

Command Window

New to MATLAB? See resources for Getting Started.

>>

Fig 4.4: Feature Extraction for Monarch



Editor - main.m | Variables - Feature{7, 1}

Feature | nFeature | Feature{7, 1}

Feature{7, 1}

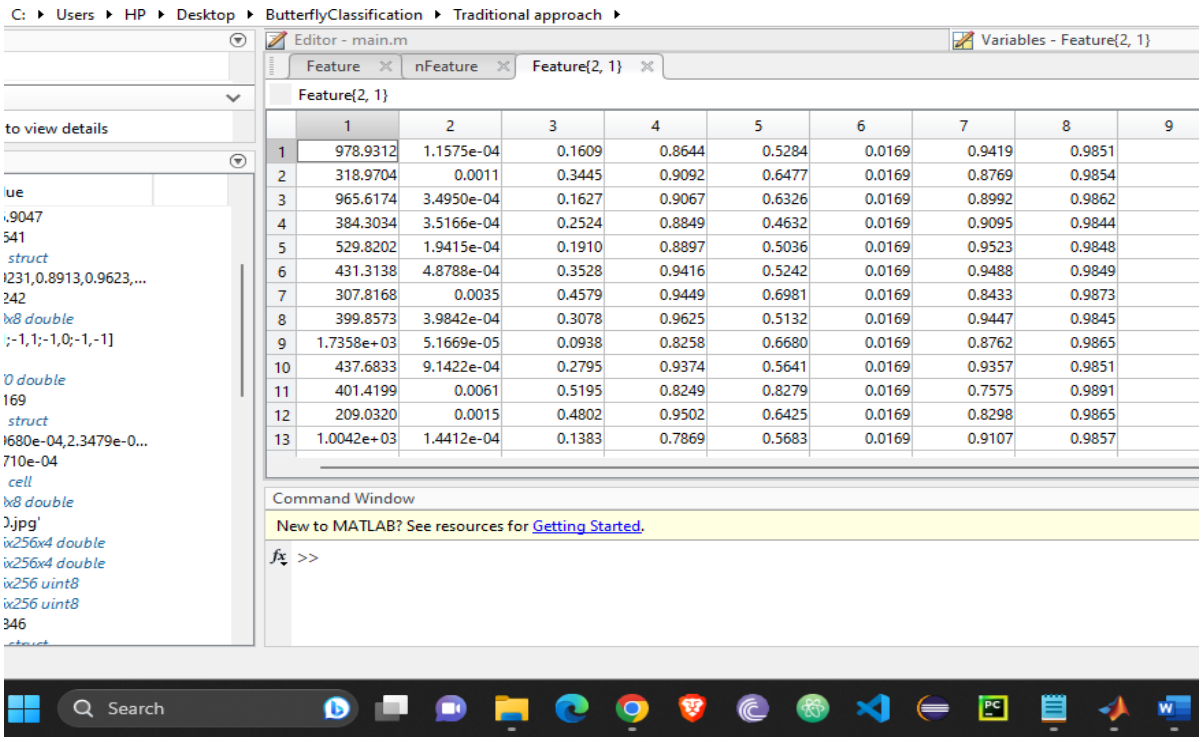
	1	2	3	4	5	6	7	8	9
1	280.2267	0.0017	0.4427	0.9332	0.6265	0.0169	0.9011	0.9862	
2	268.8322	3.2020e-04	0.2920	0.9465	0.4838	0.0169	0.9236	0.9843	
3	445.2671	2.9052e-04	0.2815	0.9321	0.4575	0.0169	0.9563	0.9844	
4	249.1162	0.0069	0.4481	0.9469	0.7544	0.0169	0.8037	0.9883	
5	394.7628	2.6413e-04	0.2442	0.9229	0.5091	0.0169	0.8623	0.9851	
6	365.4681	4.7445e-04	0.2991	0.9147	0.4889	0.0169	0.9443	0.9844	
7	334.9671	6.6489e-04	0.3558	0.9504	0.5739	0.0169	0.9358	0.9857	
8	487.3373	5.1556e-04	0.2719	0.9603	0.5908	0.0169	0.9803	0.9851	
9	431.6367	2.5429e-04	0.2423	0.9118	0.6005	0.0169	0.9594	0.9856	
10	377.8188	4.9392e-04	0.3166	0.9629	0.5486	0.0169	0.9358	0.9855	
11	337.8376	1.7591e-04	0.1986	0.9211	0.5654	0.0169	0.9396	0.9849	
12	544.6116	2.3390e-04	0.2561	0.9365	0.5288	0.0169	0.9730	0.9852	
13	593.3337	1.8125e-04	0.2272	0.9241	0.5724	0.0169	0.9453	0.9857	

Command Window

New to MATLAB? See resources for Getting Started.

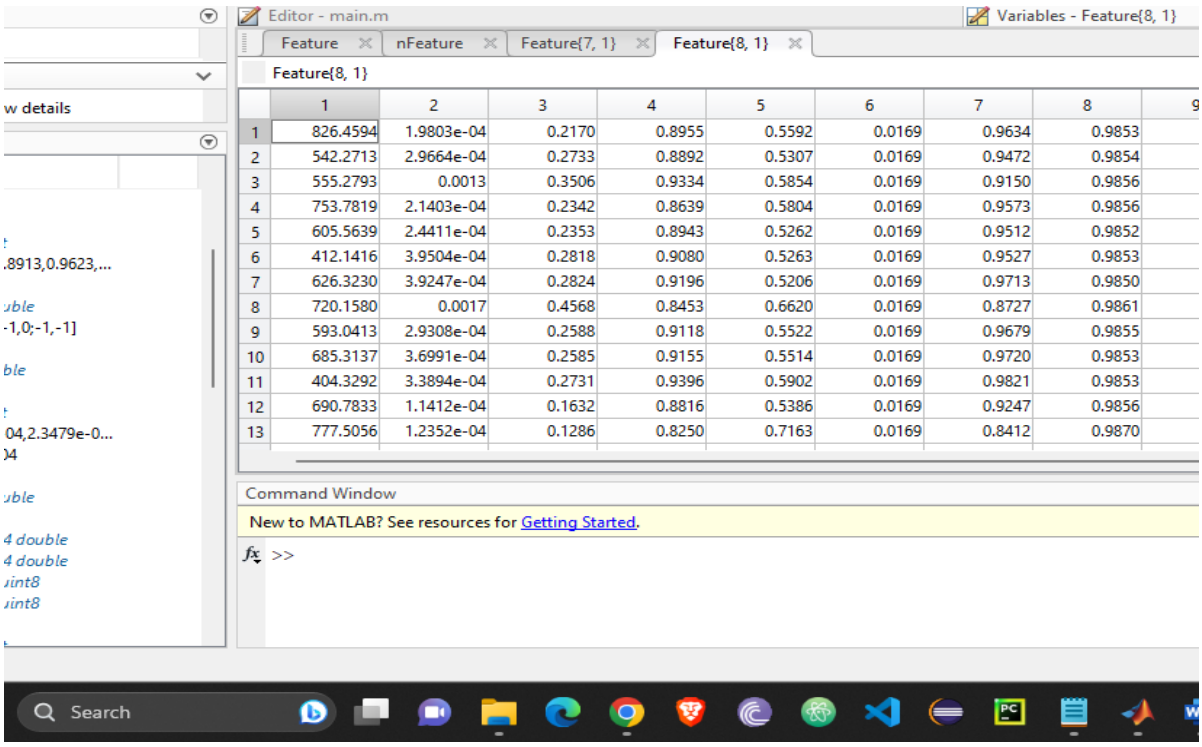
>>

Fig 4.5: Feature Extraction for Red Admiral



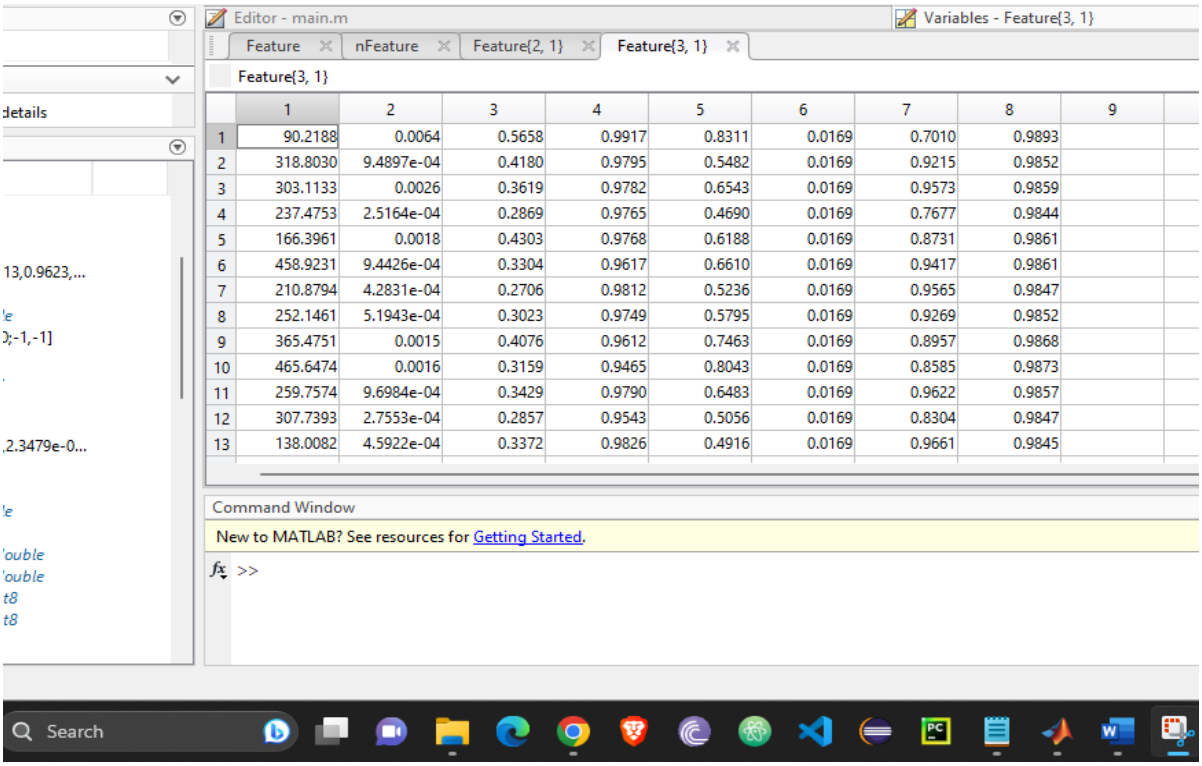
	1	2	3	4	5	6	7	8	9
1	978.9312	1.1575e-04	0.1609	0.8644	0.5284	0.0169	0.9419	0.9851	
2	318.9704	0.0011	0.3445	0.9092	0.6477	0.0169	0.8769	0.9854	
3	965.6174	3.4950e-04	0.1627	0.9067	0.6326	0.0169	0.8992	0.9862	
4	384.3034	3.5166e-04	0.2524	0.8849	0.4632	0.0169	0.9095	0.9844	
5	529.8202	1.9415e-04	0.1910	0.8897	0.5036	0.0169	0.9523	0.9848	
6	431.3138	4.8788e-04	0.3528	0.9416	0.5242	0.0169	0.9488	0.9849	
7	307.8168	0.0035	0.4579	0.9449	0.6981	0.0169	0.8433	0.9873	
8	399.8573	3.9842e-04	0.3078	0.9625	0.5132	0.0169	0.9447	0.9845	
9	1.7358e+03	5.1669e-05	0.0938	0.8258	0.6680	0.0169	0.8762	0.9865	
10	437.6833	9.1422e-04	0.2795	0.9374	0.5641	0.0169	0.9357	0.9851	
11	401.4199	0.0061	0.5195	0.8249	0.8279	0.0169	0.7575	0.9891	
12	209.0320	0.0015	0.4802	0.9502	0.6425	0.0169	0.8298	0.9865	
13	1.0042e+03	1.4412e-04	0.1383	0.7869	0.5683	0.0169	0.9107	0.9857	

Fig 4.6: Feature Extraction for Buckeye



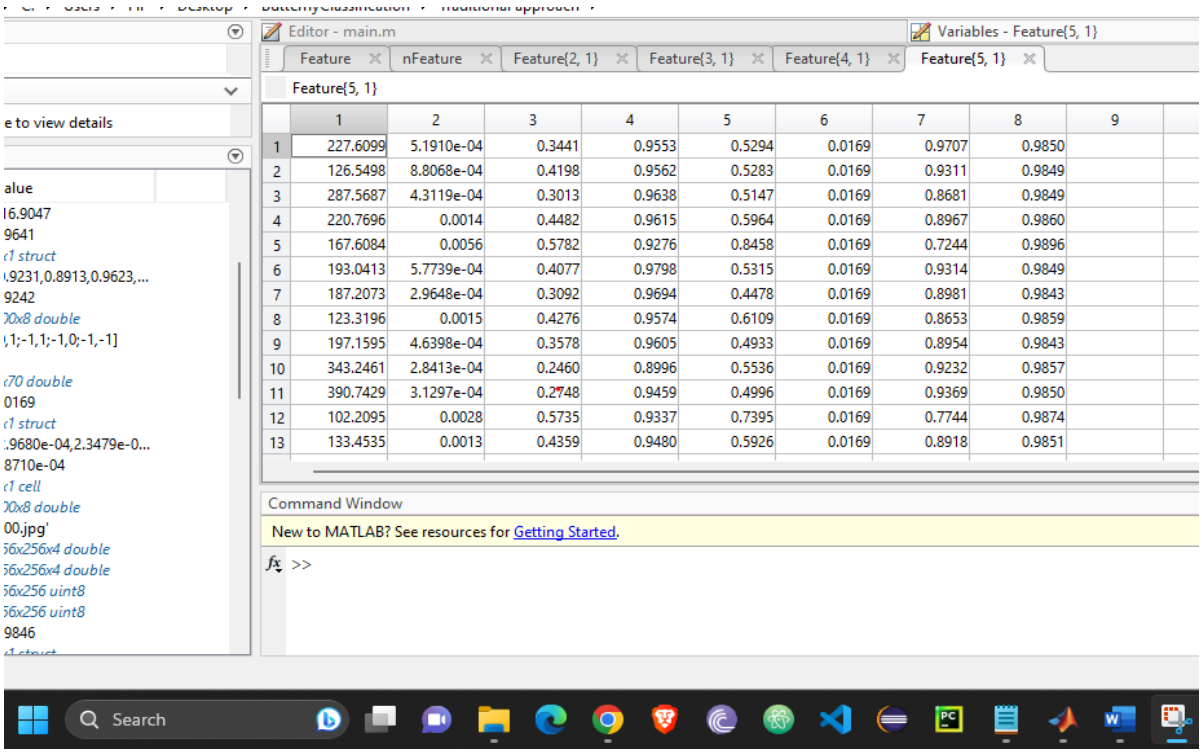
	1	2	3	4	5	6	7	8	9
1	826.4594	1.9803e-04	0.2170	0.8955	0.5592	0.0169	0.9634	0.9853	
2	542.2713	2.9664e-04	0.2733	0.8892	0.5307	0.0169	0.9472	0.9854	
3	555.2793	0.0013	0.3506	0.9334	0.5854	0.0169	0.9150	0.9856	
4	753.7819	2.1403e-04	0.2342	0.8639	0.5804	0.0169	0.9573	0.9856	
5	605.5639	2.4411e-04	0.2353	0.8943	0.5262	0.0169	0.9512	0.9852	
6	412.1416	3.9504e-04	0.2818	0.9080	0.5263	0.0169	0.9527	0.9853	
7	626.3230	3.9247e-04	0.2824	0.9196	0.5206	0.0169	0.9713	0.9850	
8	720.1580	0.0017	0.4568	0.8453	0.6620	0.0169	0.8727	0.9861	
9	593.0413	2.9308e-04	0.2588	0.9118	0.5522	0.0169	0.9679	0.9855	
10	685.3137	3.6991e-04	0.2585	0.9155	0.5514	0.0169	0.9720	0.9853	
11	404.3292	3.3894e-04	0.2731	0.9396	0.5902	0.0169	0.9821	0.9853	
12	690.7833	1.1412e-04	0.1632	0.8816	0.5386	0.0169	0.9247	0.9856	
13	777.5056	1.2352e-04	0.1286	0.8250	0.7163	0.0169	0.8412	0.9870	

Fig 4.7: Feature Extraction for Black Swallowtail



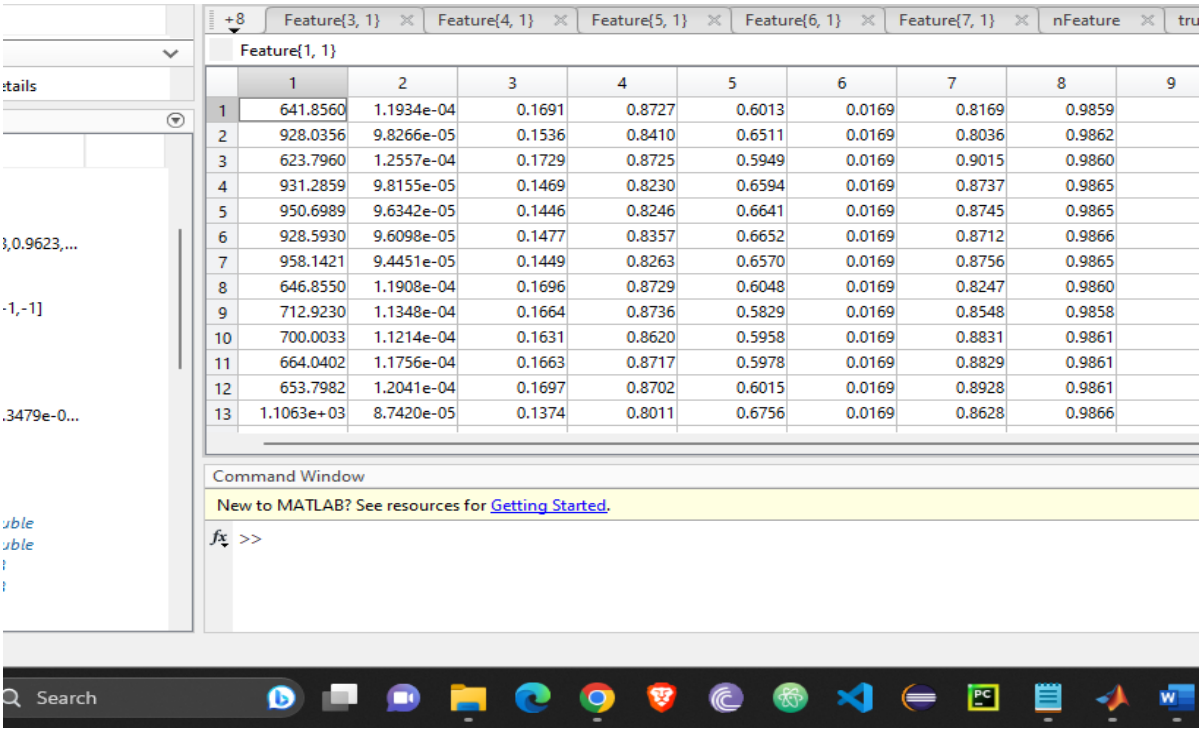
	1	2	3	4	5	6	7	8	9
1	90.2188	0.0064	0.5658	0.9917	0.8311	0.0169	0.7010	0.9893	
2	318.8030	9.4897e-04	0.4180	0.9795	0.5482	0.0169	0.9215	0.9852	
3	303.1133	0.0026	0.3619	0.9782	0.6543	0.0169	0.9573	0.9859	
4	237.4753	2.5164e-04	0.2869	0.9765	0.4690	0.0169	0.7677	0.9844	
5	166.3961	0.0018	0.4303	0.9768	0.6188	0.0169	0.8731	0.9861	
6	458.9231	9.4426e-04	0.3304	0.9617	0.6610	0.0169	0.9417	0.9861	
7	210.8794	4.2831e-04	0.2706	0.9812	0.5236	0.0169	0.9565	0.9847	
8	252.1461	5.1943e-04	0.3023	0.9749	0.5795	0.0169	0.9269	0.9852	
9	365.4751	0.0015	0.4076	0.9612	0.7463	0.0169	0.8957	0.9868	
10	465.6474	0.0016	0.3159	0.9465	0.8043	0.0169	0.8585	0.9873	
11	259.7574	9.6984e-04	0.3429	0.9790	0.6483	0.0169	0.9622	0.9857	
12	307.7393	2.7553e-04	0.2857	0.9543	0.5056	0.0169	0.8304	0.9847	
13	138.0082	4.5922e-04	0.3372	0.9826	0.4916	0.0169	0.9661	0.9845	

Fig 4.8: Feature Extraction for Cabbage White



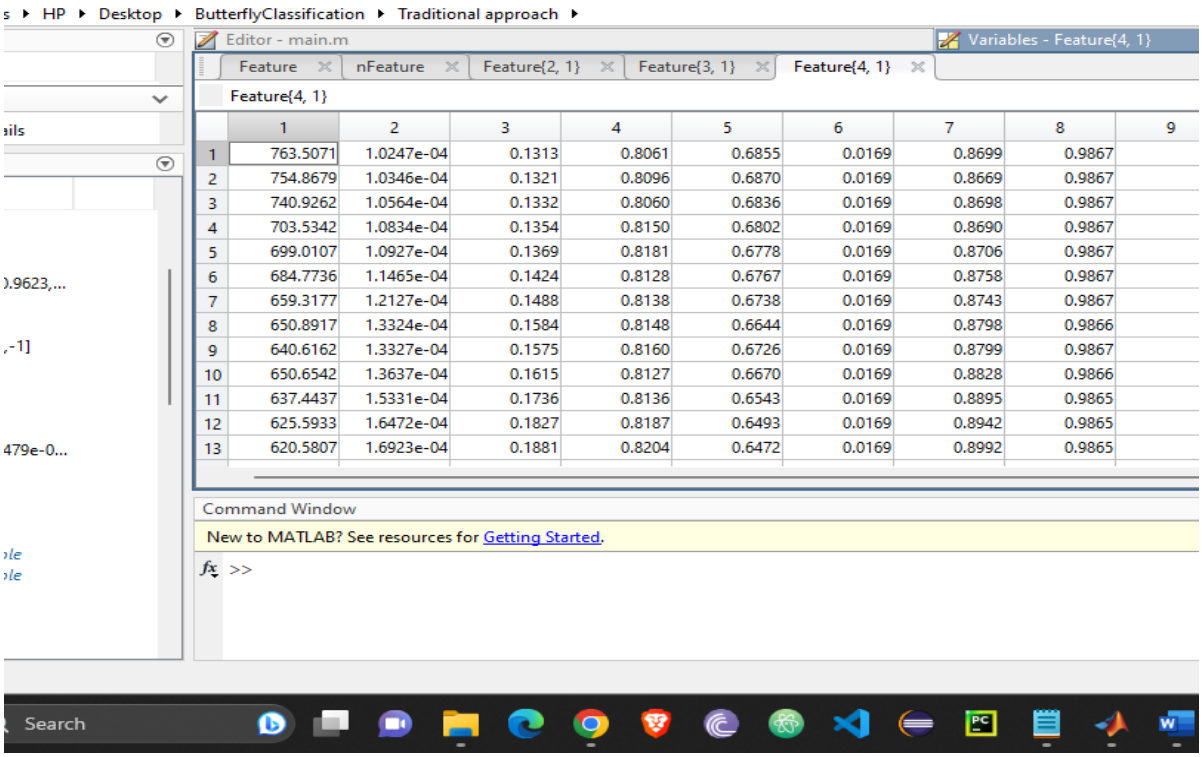
	1	2	3	4	5	6	7	8	9
1	227.6099	5.1910e-04	0.3441	0.9553	0.5294	0.0169	0.9707	0.9850	
2	126.5498	8.8068e-04	0.4198	0.9562	0.5283	0.0169	0.9311	0.9849	
3	287.5687	4.3119e-04	0.3013	0.9638	0.5147	0.0169	0.8681	0.9849	
4	220.7696	0.0014	0.4482	0.9615	0.5964	0.0169	0.8967	0.9860	
5	167.6084	0.0056	0.5782	0.9276	0.8458	0.0169	0.7244	0.9896	
6	193.0413	5.7739e-04	0.4077	0.9798	0.5315	0.0169	0.9314	0.9849	
7	187.2073	2.9648e-04	0.3092	0.9694	0.4478	0.0169	0.8981	0.9843	
8	123.3196	0.0015	0.4276	0.9574	0.6109	0.0169	0.8653	0.9859	
9	197.1595	4.6398e-04	0.3578	0.9605	0.4933	0.0169	0.8954	0.9843	
10	343.2461	2.8413e-04	0.2460	0.8996	0.5536	0.0169	0.9232	0.9857	
11	390.7429	3.1297e-04	0.2748	0.9459	0.4996	0.0169	0.9369	0.9850	
12	102.2095	0.0028	0.5735	0.9337	0.7395	0.0169	0.7744	0.9874	
13	133.4535	0.0013	0.4359	0.9480	0.5926	0.0169	0.8918	0.9851	

Fig 4.9: Feature Extraction for Crimson Patched Longwing



	1	2	3	4	5	6	7	8	9
1	641.8560	1.1934e-04	0.1691	0.8727	0.6013	0.0169	0.8169	0.9859	
2	928.0356	9.8266e-05	0.1536	0.8410	0.6511	0.0169	0.8036	0.9862	
3	623.7960	1.2557e-04	0.1729	0.8725	0.5949	0.0169	0.9015	0.9860	
4	931.2859	9.8155e-05	0.1469	0.8230	0.6594	0.0169	0.8737	0.9865	
5	950.6989	9.6342e-05	0.1446	0.8246	0.6641	0.0169	0.8745	0.9865	
6	928.5930	9.6098e-05	0.1477	0.8357	0.6652	0.0169	0.8712	0.9866	
7	958.1421	9.4451e-05	0.1449	0.8263	0.6570	0.0169	0.8756	0.9865	
8	646.8550	1.1908e-04	0.1696	0.8729	0.6048	0.0169	0.8247	0.9860	
9	712.9230	1.1348e-04	0.1664	0.8736	0.5829	0.0169	0.8548	0.9858	
10	700.0033	1.1214e-04	0.1631	0.8620	0.5958	0.0169	0.8831	0.9861	
11	664.0402	1.1756e-04	0.1663	0.8717	0.5978	0.0169	0.8829	0.9861	
12	653.7982	1.2041e-04	0.1697	0.8702	0.6015	0.0169	0.8928	0.9861	
13	1.1063e+03	8.7420e-05	0.1374	0.8011	0.6756	0.0169	0.8628	0.9866	

Fig 4.10: Feature Extraction for Blue Mormon



	1	2	3	4	5	6	7	8	9
1	763.5071	1.0247e-04	0.1313	0.8061	0.6855	0.0169	0.8699	0.9867	
2	754.8679	1.0346e-04	0.1321	0.8096	0.6870	0.0169	0.8669	0.9867	
3	740.9262	1.0564e-04	0.1332	0.8060	0.6836	0.0169	0.8698	0.9867	
4	703.5342	1.0834e-04	0.1354	0.8150	0.6802	0.0169	0.8690	0.9867	
5	699.0107	1.0927e-04	0.1369	0.8181	0.6778	0.0169	0.8706	0.9867	
6	684.7736	1.1465e-04	0.1424	0.8128	0.6767	0.0169	0.8758	0.9867	
7	659.3177	1.2127e-04	0.1488	0.8138	0.6738	0.0169	0.8743	0.9867	
8	650.8917	1.3324e-04	0.1584	0.8148	0.6644	0.0169	0.8798	0.9866	
9	640.6162	1.3327e-04	0.1575	0.8160	0.6726	0.0169	0.8799	0.9867	
10	650.6542	1.3637e-04	0.1615	0.8127	0.6670	0.0169	0.8828	0.9866	
11	637.4437	1.5331e-04	0.1736	0.8136	0.6543	0.0169	0.8895	0.9865	
12	625.5933	1.6472e-04	0.1827	0.8187	0.6493	0.0169	0.8942	0.9865	
13	620.5807	1.6923e-04	0.1881	0.8204	0.6472	0.0169	0.8992	0.9865	

Fig 4.11: Feature Extraction for Common Rose

4.6.2 kNN Classification Results

The features extracted from the GLCM and LBP approaches are fed as an input for kNN classifier. The results of kNN classification for all the image samples of butterfly species is shown in Fig 4.12, Fig 4.13, Fig 4.14, Fig 4.15, Fig 4.16, Fig 4.17, Fig 4.18, Fig 4.19.

```
clear
clc;
% Load input data from .mat file
load("C:\Users\HP\Desktop\ButterflyClassification\Traditional approach\feature1.mat")
%run('featureExtraction.m');
path = "C:\Users\HP\Desktop\ButterflyClassification\Assessement\monarch.jfif";
img = imread(path);
new_size = [256, 256];
B = imresize(img, new_size);
grayImage = im2gray(B);
nFeature(1,8) = 0;
distance = 1; % Distance between the pixels
directions = [0 1; -1 1; -1 0; -1 -1]; % Directions to consider
numGrayLevels = 256; % Number of gray levels

% Compute the GLCM
glcm = graycomatrix(grayImage, 'Offset', directions, 'NumLevels', numGrayLevels, 'Gra
contrast1 = graycoprops(glcm, 'Contrast');
contrast2 = cell2mat(struct2cell(contrast1));
contrast mean = (contrast2(1)+contrast2(2)+contrast2(3)+contrast2(4))/4;
```

```
MONARCH
>>
```

Fig 4.12: kNN Classification for Monarch

```
clear
clc;
% Load input data from .mat file
load("C:\Users\HP\Desktop\ButterflyClassification\Traditional approach\feature1.mat");
%run('featureExtraction.m');
path = "C:\Users\HP\Desktop\ButterflyClassification\Assessement\redAdmir.jpg";
img = imread(path);
new_size = [256, 256];
B = imresize(img, new_size);
grayImage = im2gray(B);
nFeature(1,8) = 0;
distance = 1; % Distance between the pixels
directions = [0 1; -1 1; -1 0; -1 -1]; % Directions to consider
numGrayLevels = 256; % Number of gray levels

% Compute the GLCM
glcm = graycomatrix(grayImage, 'Offset', directions, 'NumLevels', numGrayLevels, 'Gray
contrast1 = graycoprops(glcm, 'Contrast');
contrast2 = cell2mat(struct2cell(contrast1));
contrast mean = (contrast2(1)+contrast2(2)+contrast2(3)+contrast2(4))/4;
```

```
REDADMIRAL
>>
```

Fig 4.13: kNN Classification for Red Admiral

```
clear
clc;
% Load input data from .mat file
load("C:\Users\HP\Desktop\ButterflyClassification\Traditional approach\feature1.mat");
%run('featureExtraction.m');
path = "C:\Users\HP\Desktop\ButterflyClassification\Assessement\bucke.jpg";
img = imread(path);
new_size = [256, 256];
B = imresize(img, new_size);
grayImage = im2gray(B);
nFeature(1,8) = 0;
distance = 1; % Distance between the pixels
directions = [0 1; -1 1; -1 0; -1 -1]; % Directions to consider
numGrayLevels = 256; % Number of gray levels

% Compute the GLCM
glcm = graycomatrix(grayImage, 'Offset', directions, 'NumLevels', numGrayLevels, 'Gray
contrast1 = graycoprops(glcm, 'Contrast');
contrast2 = cell2mat(struct2cell(contrast1));
contrast_mean = (contrast2(1)+contrast2(2)+contrast2(3)+contrast2(4))/4;
```

```
BUCKEYE
>>
```

Fig 4.14: kNN Classification for Buckeye

```
clear
clc;
% Load input data from .mat file
load("C:\Users\HP\Desktop\ButterflyClassification\Traditional approach\feature1.mat"); %
%run('featureExtraction.m');
path = "C:\Users\HP\Desktop\ButterflyClassification\Assessement\swallowtail.jfif";
img = imread(path);
new_size = [256, 256];
B = imresize(img, new_size);
grayImage = im2gray(B);
nFeature(1,8) = 0;
distance = 1; % Distance between the pixels
directions = [0 1; -1 1; -1 0; -1 -1]; % Directions to consider
numGrayLevels = 256; % Number of gray levels

% Compute the GLCM
glcm = graycomatrix(grayImage, 'Offset', directions, 'NumLevels', numGrayLevels, 'GrayLi
contrast1 = graycoprops(glcm, 'Contrast');
contrast2 = cell2mat(struct2cell(contrast1));
contrast_mean = (contrast2(1)+contrast2(2)+contrast2(3)+contrast2(4))/4;
```

```
BLACK SWALLOWTAIL
>>
```

Fig 4.15: kNN Classification for Black Swallowtail

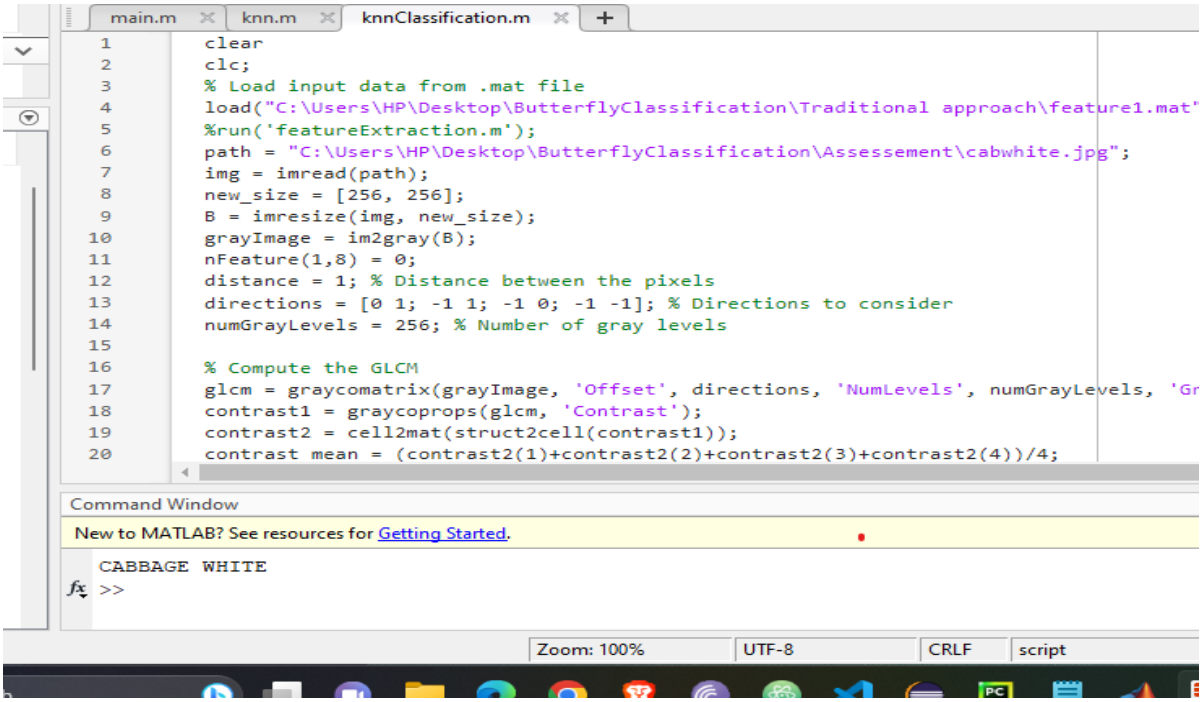


Fig 4.16: kNN Classification for Cabbage White

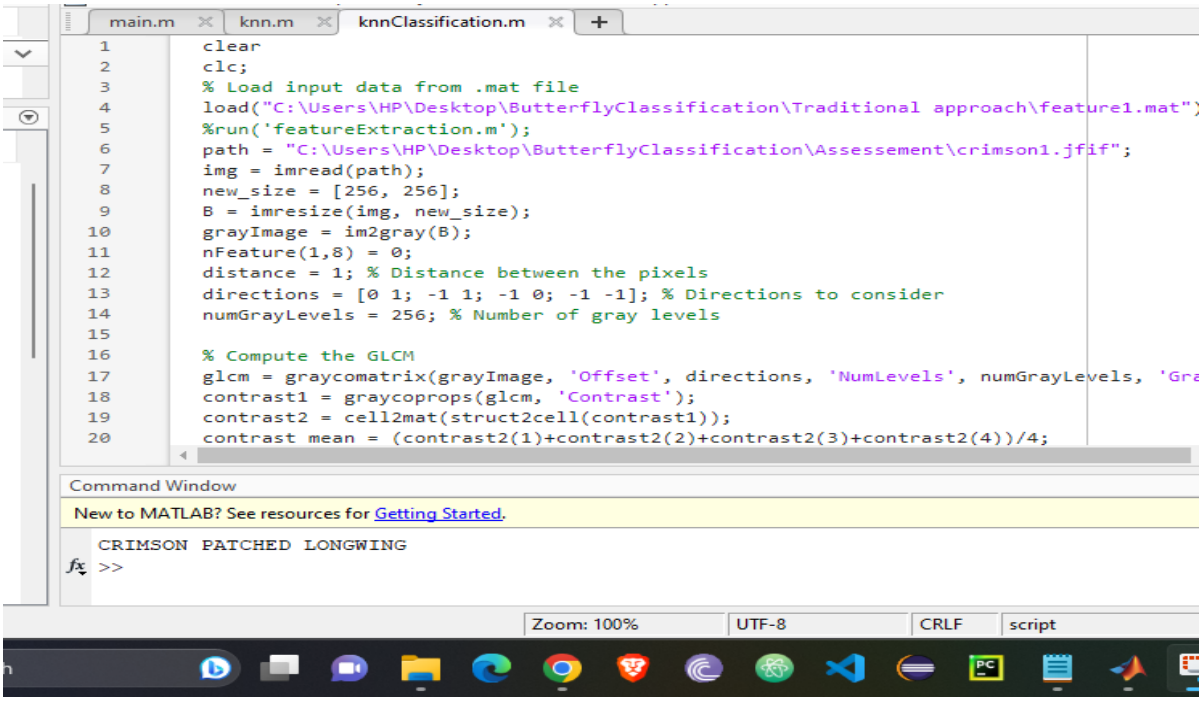


Fig 4.17: kNN Classification for Crimson Patched Longwing

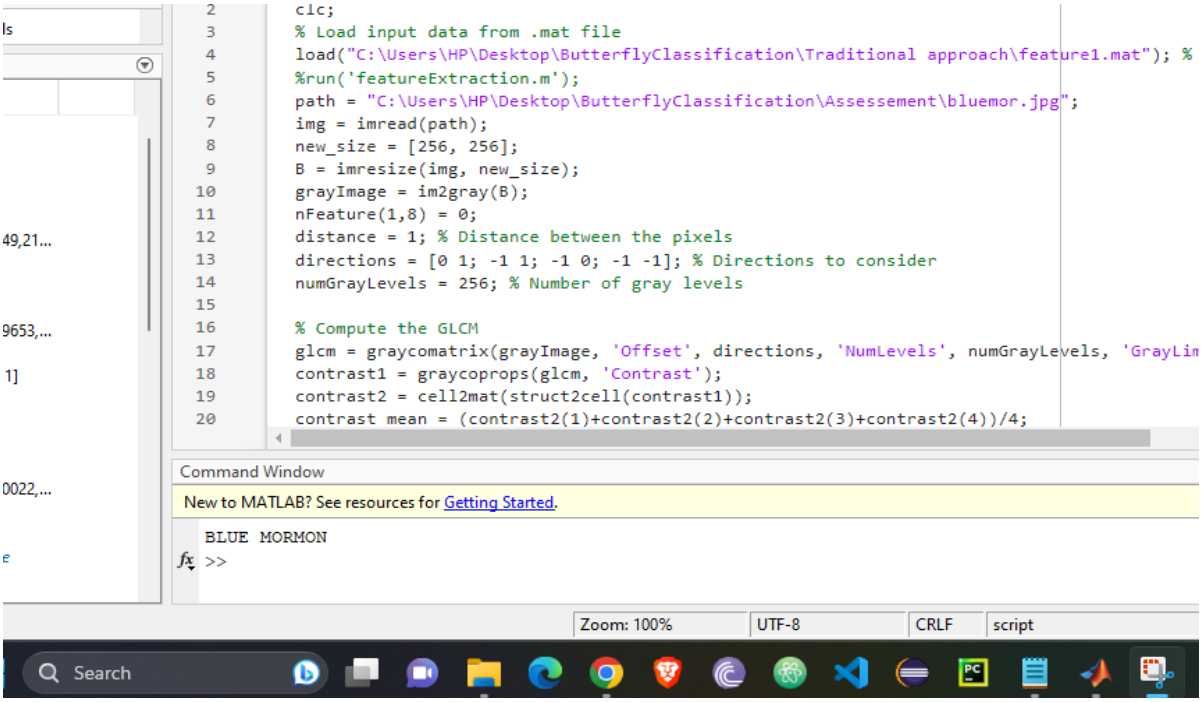


Fig 4.18: kNN Classification for Blue Mormon

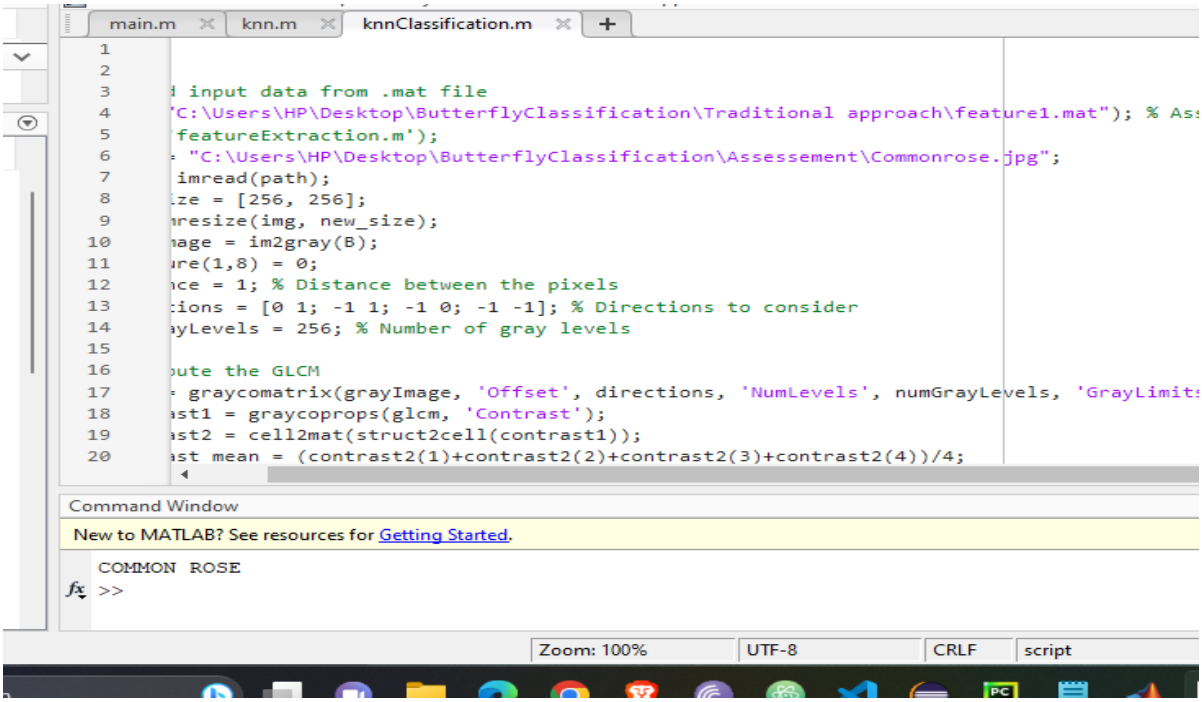


Fig 4.19: kNN Classification for Common Rose

4.6.3 Deep Learning Classification Results

The results of AlexNet classification along with the obtained confusion matrix is presented in Fig 4.20, Fig 4.21, Fig 4.22, Fig 4.23, Fig 4.24, Fig 4.25, Fig 4.26, Fig 4.27.

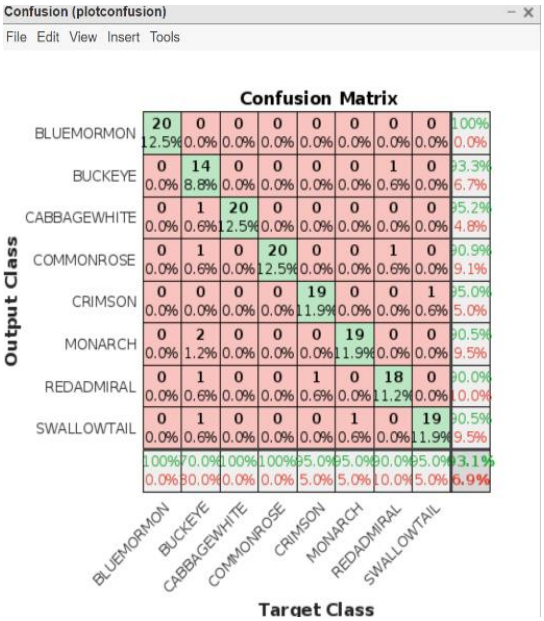


Fig 4.20: AlexNet Classification for Monarch

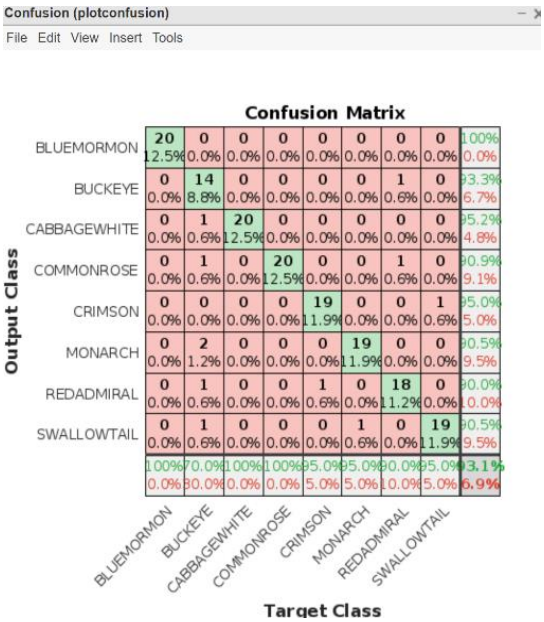


Fig 4.21: AlexNet Classification for Red Admiral

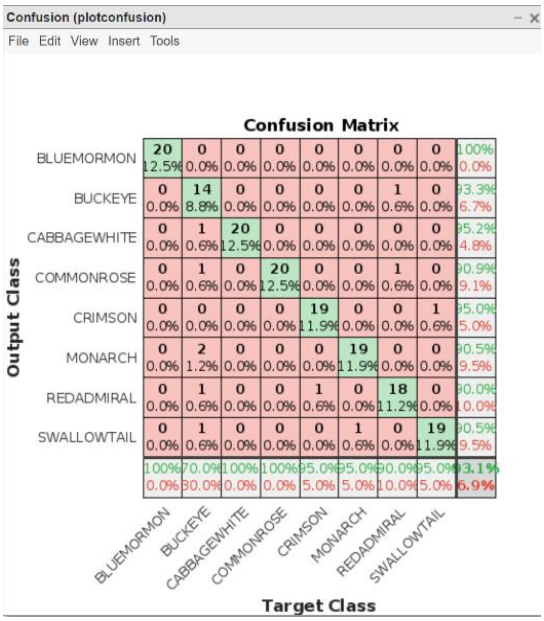


Fig 4.22: AlexNet Classification for Buckeye

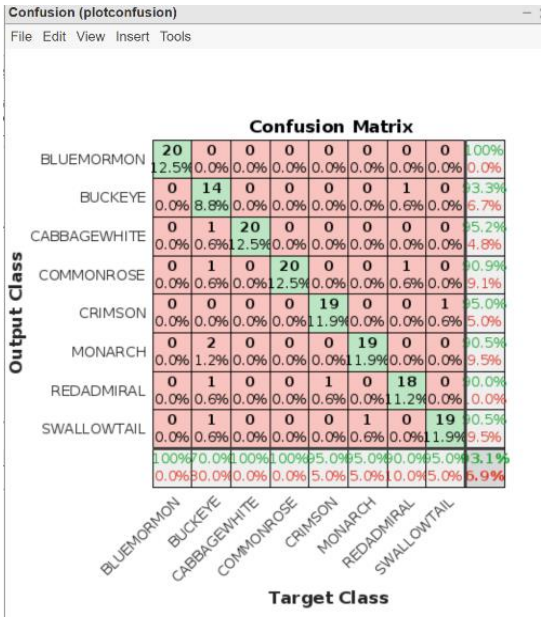


Fig 4.23: AlexNet Classification for Black Swallowtail

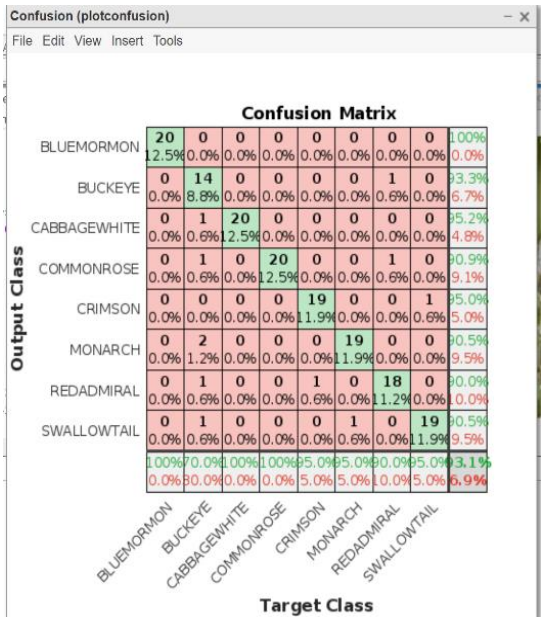


Fig 4.24: AlexNet Classification for Cabbage White

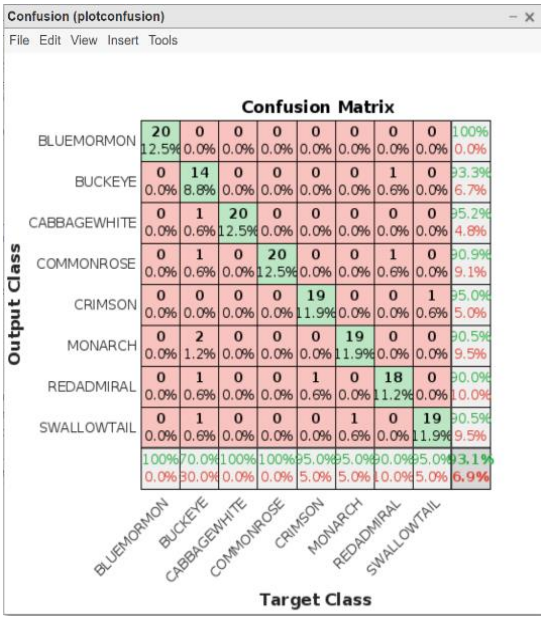


Fig 4.25: AlexNet Classification for Crimson Patched Longwing

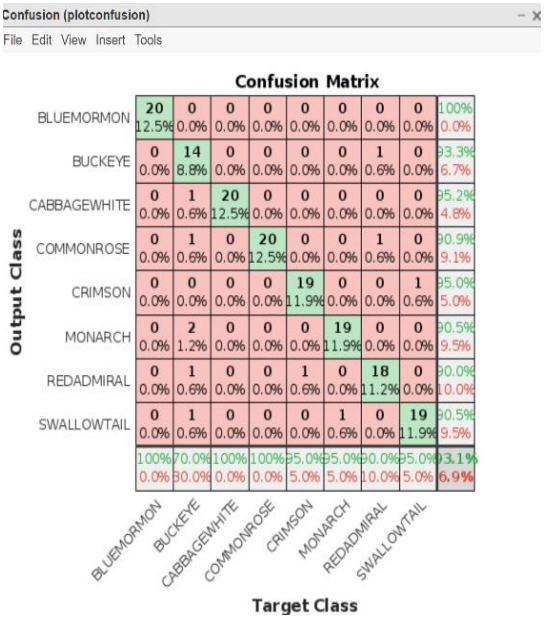


Fig 4.26: AlexNet Classification for Blue Mormon

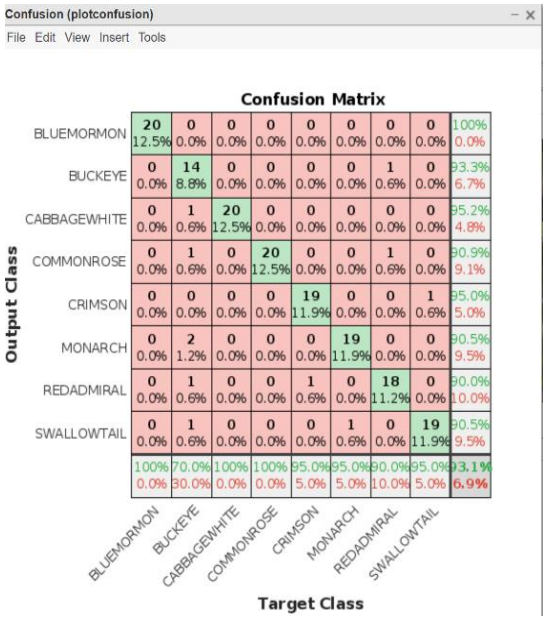


Fig 4.27: AlexNet Classification for Common Rose

4.7 Feasibility Study

A comparative study of results obtained using traditional machine learning approach and deep learning approach has been carried out. Traditional machine learning approach using feature extraction for classification of image sample using kNN classifier produced an accuracy of 89.21%. The accuracy obtained for dataset of 800 images is shown in Fig 4.28

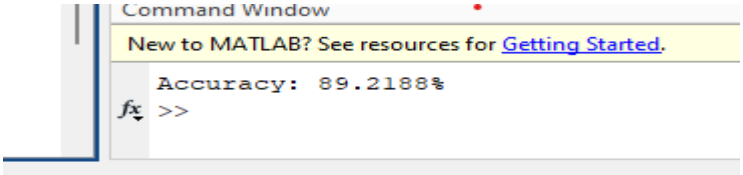


Fig 4.28: kNN accuracy

Deep learning approach for image classification using AlexNet produced an accuracy of 93.12% as shown in Fig 4.29.

Fig 4.29: AlexNet accuracy

AlexNet model accuracy varied for 93% to 93.5% on multiple runs. On analysis and reference from various internet source it was found out that since MatLab online version was adopted for AlexNet classification, the program is set with different CPU configuration on each run, which results in minute change in accuracy of the model. Deep learning approach model produced higher accuracy than the traditional approach model.

4.8 Summary

An extensive literature review has been completed. The dataset samples have undergone several image pre-processing techniques as resizing, image enhancement, and image segmentation, and property analyses have been carried out for both the experimental and proposed datasets. The functions GLCM and LBP is to extract features. A k-NN classifier is used to classify data using the collected features, and successful, accurate classification was achieved. A deep learning model, AlexNet is used for classification of butterflies resulting in good accuracy and a confusion matrix is derived. A detailed feasibility study is carried out and results are compared for both the approaches employed in this proposed work.

CHAPTER 5

EPILOGUE

In the present work, a detailed literature survey has been accomplished. Various image pre-processing methods such as resize, image enhancement and image segmentation has been applied to the dataset samples and property analysis has been derived for both experimental and proposed dataset. GLCM and LBP functions are used for feature extraction. Further, k-NN classifier is implemented for classification using extracted features and obtained classification successfully with good accuracy. A deep learning model AlexNet is implemented for butterfly classification to achieve good accuracy followed by an accurate confusion matrix for the same. A feasibility study is carried out for the approaches used for the classification of the butterflies. It is observed that the deep learning approach yields more accurate classification results than that of traditional approach.

REFERENCES

[1]. R. Andrian, D. Maharani, M.A. Muhammad and A.Junaidi, "Butterfly identification using gray level co-occurrence matrix (GLCM) extraction feature and k-nearest neighbor(KNN) classification", Register: J*urnal Ilmiah Teknologi Sistem Informasi,vol. 6, no. 1*, pp. 11-21, 2020.

[2]. Xue, A., Li, F. & Xiong, Y. Automatic Identification of Butterfly Species Based on Gray-Level Co-occurrence Matrix Features of Image Block. *J. Shanghai Jiaotong Univ. (Sci.)* 24, 220–225 (2019). https://doi.org/10.1007/s12204-018-2013-y

[3]. D.P.Mishra,T.K.Tripathy and S.Chakraborty,"CNN for Butterfly Classification", in *2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE),* 2018, pp. 200-205, doi: 10.1109/ICRIEECE44171.2018.9008419.

[4]. N. N. Kamaron Arzar, N. Sabri, N. F. Mohd Johari, A. A. Shari, Mohd R. Noordin, S. Ibrahim, “Butterfly Species Identification using Convolutional Neural Network”, in *2019 IEEE International Conference on Automatic Control and Intelligent System (I2CACIS 2019), 29 June 2019, Selangor, Malaysia,* pp. 221-224, 2019.

[5]. Ruoyan Zhao, Cuixia Li1, Shuai Ye1 and Xinru Fang, “Butterfly Recognition Based on Faster R-CNN”,

in *2018 IOP Conf. Series: Journal of Physics: Conf. Series 1176 (2019) 032048,* pp. 1-6, 2019.

[6]. Xie, J., Lu, Y.,Wu, Z. et al. “Investigations of butterfly species identification from images in natural environments” in *.Int.J. Mach. Learn. & Cyber. 12, 2431–2442 (2021)*, doi.org/10.1007/s13042-021-01322-8.

[7]. X. Chen, B. Wang and Y. Gao, "Gaussian Convolution Angles: Invariant Vein and Texture Descriptors for Butterfly Species Identification," in *2020 25th International Conference on Pattern Recognition (ICPR), 2021,* pp. 5798-5803, 2020.

[8]. X. Jia, X. Tan, G. Jin and R. O. Sinnott, "Lepidoptera Classification through Deep Learning," in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2020*, pp. 1-6, 2020.

[9]. Rajeena P. P., F.; Orban, R.; Vadivel, K.S.; Subramanian, M.; Muthusamy, S.; Elminaam, D.S.A.; Nabil, A.; Abulaigh, L.; Ahmadi, M.; Ali, M.A.S. "A Novel Method for the Classification of Butterfly Species Using Pre-Trained CNN Models", in *Electronics* **2022**, *11,* pp. 1-20, 2022.

[10]. Ayad Saad Almryad, Hakan Kutucu, "Automatic identification for field butterflies by convolutional neural networks", in *2020 Karabuk University. Publishing services by Elsevier B.V.,* pp. 189-195, 2020.

[11]. Li, Fan, and Yin Xiong. "Automatic Identification of Butterfly Species Based on HoMSC and GLCMoIB." in *The Visual Computer, vol. 34, no. 11, Springer Science and Business Media LLC*, Aug. 2017, pp. 1525–33.

[12]. Alweshah, Mohammed, et al. "The Monarch Butterfly Optimization Algorithm for Solving Feature Selection Problems." *in Neural Computing and Applications, vol. 34, no. 14, Springer Science and Business Media LLC*, July 2020, pp. 11267–81.

[13]. Kaya, Yılmaz, et al. "Evaluation of Texture Features for Automatic Detecting Butterfly Species Using Extreme Learning Machine." *in Journal of Experimental & Theoretical Artificial Intelligence, vol. 26, no. 2, Informa UK Limited*, Jan. 2014, pp. 267–81.

[14]. Tang, Hui, et al. "Deep Learning Techniques for Automatic Butterfly Segmentation in Ecological Images." in *Computers and Electronics in Agriculture, vol. 178, Elsevier BV,* Nov. 2020, p. 105739.

[15]. Reema Rodrigues, et al. "Butterfly Species Identification Using Convolutional Neural Network." in *International Journal of Research in Engineering, Science and Management Volume-3, Issue-5,* May-2020*,* p. 267

[16]. Bacanin, Nebojsa, et al. "Monarch Butterfly Optimization Based Convolutional Neural Network Design." in *Mathematics, vol. 8, no. 6, MDPI AG*, June 2020, p. 936.

[17]. Kaya, Yılmaz, et al. "Two Novel Local Binary Pattern Descriptors for Texture Analysis." in *Applied Soft Computing, vol. 34, Elsevier BV*, Sept. 2015, pp. 728–35.

[18]. Lin, Zhongqi, et al. "Increasingly Specialized Perception Network for Fine-Grained Visual Categorization of Butterfly Specimens." in *IEEE Access, vol. 7, Institute of Electrical and Electronics Engineers (IEEE)*, 2019, pp. 123367–92.

[19]. Li, Maopeng, et al. "Multi-scale Sparse Network With Cross-Attention Mechanism for Image-based Butterflies Fine-grained Classification." *in Applied Soft Computing, vol. 117, Elsevier BV*, Mar. 2019, p. 108419.

[20]. Xin, Dongjun, et al. "Fine-Grained Butterfly Classification in Ecological Images Using Squeeze-And-Excitation and Spatial Attention Modules." in *Applied Sciences, vol. 10, no. 5, MDPI AG*, Mar. 2020, p. 1681.

[21]. Lin Nie, Keze Wang et al. "Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion." *in IEEE Access* 2020, *vol. 8*, p. 124722–124733.

[22]. Zhu et al., (2017, November) "SVM and AlexNet deep learning for automatic butterfly segmentation in ecological images" in *Computers and Electronics in Agriculture*, *178*, p.105729-105739.