JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Butterflies Recognition using Enhanced Transfer Learning and Data Augmentation

Harish Trio Adityawan [1], Omar Farroq [2], Stefanus Santosa [3], Hussain Md Mehedul Islam[4], Md Kamruzzaman Sarker [5], and De Rosal Ignatius Moses Setiadi [1, *]

[1] Faculty of Computer Science, Dian Nuswatoro University, Semarang, Indonesia;
e-mail: haradityawan@gmail.com, moses@dsn.dinus.ac.id
[2] Department of Electronics Engineering, ZH College of Engineering and Technology Aligarh Muslim University, Aligarh, India; e-mail: omar.farooq@amu.ac.in
[3] Department of Civil Engineering, Politeknik Negeri Semarang, Indonesia;
e-mail: stefanus.santosa@polines.ac.id
[4] Software Engineer, The Mathworks, Inc., United States; e-mail: mehadi.cuet@gmail.com
[5] Department of Computer Science, Bowie State University, United States; e-mail: ksarker@bowiestate.edu

* Corresponding Author: De Rosal Ignatius Moses Setiadi

Abstract: Butterflies' recognition serves a crucial role as an environmental indicator and a key factor in plant pollination. The automation of this recognition process, facilitated by Convolutional Neural Networks (CNNs), can expedite this task. Several pre-trained CNN models, such as VGG, ResNet, and Inception, have been widely used for this purpose. However, the scope of previous research has been somewhat constrained, focusing only on a maximum of 15 classes. This study proposes to modify the CNN InceptionV3 model and combine it with three data augmentations to recognize up to 100 butterfly species. To curb overfitting, this study employs a series of data augmentation techniques. In parallel, we refine the InceptionV3 model by reducing the number of layers and integrating four new layers. The test results demonstrate that our proposed model achieves an impressive accuracy of 99.43% for 15 classes with only 10 epochs, exceeding prior models by approximately 5%. When extended to 100 classes, the model maintains a high accuracy rate of 98.49% with 50 epochs. The proposed model surpasses the performance of standard pre-trained models, including VGG16, ResNet50, and InceptionV3, illustrating its potential for broader application.

Keywords: Butterflies classification; Data augmentation; InceptionV3; Pre-trained CNN model; Transfer learning

Received: October, 16th 2023
Revised: November, 15th 2023
Accepted: November, 17th 2023
Published: November, 18th 2023

1. Introduction

Butterflies, a diverse group of insects within the order Lepidoptera, exhibit intricate wing patterns that are unique to each butterfly species. There are thousands of butterfly species worldwide and many of them share similar characteristics, patterns and colors[1]. This similarity poses a challenge for accurate and efficient recognition. However, the identification of butterfly species holds significant importance from both scientific and nature conservation standpoints. The Knowledge of butterfly diversity in a particular area aids experts and interested parties in preserving and managing their natural habitats. Additionally, butterflies can serve as sensitive environmental indicators, providing insights into changes in ecosystems and climate within a region. Furthermore, some butterflies can be pests for crops, while others play vital roles as pollinators for wild and cultivated plants. Identification of types of butterflies helps in recognizing beneficial and detrimental species for the ecosystem[2]. Unfortunately, there has been a drastic decrease in the number of taxonomists and researchers [3]. This necessitates the development of an automatic identification system to minimize errors in identifying butterfly species.

Butterflies, with their diverse wing patterns and shared characteristics, present a formidable challenge for accurate species identification. The decline in the number of taxonomists and researchers underscores the need for such identification. To address this challenge, an automated butterfly species identification process becomes imperative. In recent years, the Deep Neural Networks (DNNs) method, particularly Convolutional Neural Networks (CNNs), has emerged as the most efficient technique in various computer vision tasks, such as object detection and recognition, classification, and biometry[4]–[10]. CNNs are capable of achieving high accuracy in image recognition due to their complex and detailed network depth[11]. Typically, CNNs consist of four layers: convolutional, pooling, activation function, and fully connected[12]. The Convolutional Layer plays a crucial role in the CNN structure as it consists of a collection of filters or kernels designed to extract features from the image. The pooling layer reduces feature map dimensionality while preserving important data The activation function layer, which follows the convolution layer, introduces non-linearity to the output. This layer is used to limit or exaggerate the output.

CNNs can be customized by assembling these layers according to specific requirements. However, utilizing pre-trained CNN models offers several advantages, including improved model accuracy, reduced training time, and enhanced performance [13]–[17]. Several widely used pre-trained models, such as VGG, AlexNet, Xception, Inception, EfficientNet, DenseNet, MobileNet and ResNet which, can be used in various deep learning tasks [18]–[20]. However, the implementation of CNNs sometimes faces the challenge of overfitting, especially when the available dataset is insufficient. In such cases, the data augmentation process can effectively address this issue. Data augmentation is a regularization technique used to prevent the overfitting of a model by applying transformations such as padding, rotation, rescaling, flipping, cropping, and zooming to the dataset[13], [16], [21]. Previous research [13] has shown that data augmentation can effectively increase the accuracy, up to 52%.

Conventionally, research on butterfly identification has predominantly focused on a limited number of classes. For instance, one study [22] proposed the GoogleLeNet architecture to classify four types of butterflies and obtained an accuracy of 97.5% based on the test results. Similarly, another study [23] proposed a VGG-16 CNN model to identify ten butterfly species and achieved a high accuracy of 97%. Meanwhile, using transfer learning, a comparative study[13] evaluated the performance of various CNN models, including VGG16, VGG19, MobileNet, Xception, ResNet50, and InceptionV3. The research found that the InceptionV3 model yielded the highest accuracy of 94.66%. The aforementioned research example highlights a common trend: as the number of classes increases, there is a notable decrease in identification accuracy. This phenomenon is natural due to the increasing challenge of intra-class variation with the increasing number of recognized butterfly species [24]. The number of classes tested in the aforementioned studies is still significantly lower than the total number of butterfly species. Consequently, implementing the CNN model to recognize a more extensive range of classes would likely encounter significant challenges.

In recent years, the development of deep learning models has revolutionized the field of computer vision. One such model that has shown promising results is InceptionV3, as highlighted in research [13]. One notable advantage of InceptionV3 lies in its distinctive feature extraction capability, allowing it to extract features from images of varying sizes using a single level of convolution. [25]. InceptionV3 also has a high level of flexibility and accuracy compared to other architectures. InceptionV3 has proven its superior performance in terms of Top-1 class accuracy when compared to AlexNet, ResNet50, and GoogleNet [26]. This model also has a fairly good classification performance on a dataset with 20 classes [27]. The Inception V3 architecture consists of five layers for convolution, one average pooling layer, two max-pooling layers, one Fully Connecter (FC) Layer, and 11 inception modules that form an image-wise classification. Each InceptionV3 module consists of pooling layers and convolutional filters fixed as an activation function [28], [29]. While InceptionV3 has demonstrated high accuracy levels in various studies, it is important to note that these studies have primarily implemented the model on relatively limited class datasets. Consequently, the accuracy levels may vary when applying the same model to datasets with a larger number of classes [30]. Based on this background, this study aims to achieve the following objectives:

1. Analyze and enhance the InceptionV3 to classify butterfly species data and compare results with relevant literature.
2. Implement data augmentation techniques to improve accuracy and mitigate overfitting.
3. Enhance the performance of InceptionV3 to recognize 100 classes of butterfly species by reducing parameters, resulting in lighter calculations while maintaining minimal overfitting.

The remainder of this article is structured into four chapters. The first chapter discusses related work, providing insights from previous studies. The second chapter presents the proposed method, detailing the CNN model and the augmentation techniques employed. The third chapter focuses on the implementation and analysis, showcasing the results obtained from the study. Finally, the concluding section summarizes the findings and addresses whether the research objectives were successfully accomplished.

2. Related Works

Research on image recognition based on pre-trained CNN models has been conducted with various CNN models. For example, a study [31] identified seven facial emotions by comparing several pre-trained models, such as VGG-16, ResNet50, and SE-ResNet50. The result revealed that ResNet50 achieved the best performance, with a training accuracy of 99.54% and a validation accuracy of 99.47%. In addition, ResNet50 also obtained the highest precision and recall values as well as shorter training time. In another research endeavor [28], the identification of COVID-19 disease through chest X-ray images was explored by comparing ResNet50, InceptionV3, and VGG16 models. The findings indicated that VGG16 yielded the best results, achieving accuracies of 97.20%, 98.10%, and 98.30% for ResNet50, InceptionV3, and VGG16, respectively.

Several studies have been conducted in the field of butterfly image recognition. For example, in a study by [32], the authors proposed automation of butterfly species classification using a CNN model. The dataset was collected using web scraping techniques for the Tigers and Emigrants categories, predominantly found in the Indian region. The overall performance of the CNN model achieved an accuracy of approximately 88%. Another study by [22] proposed the GoogleLeNet architecture to classify four types of butterflies. The researchers utilized a dataset comprising 600 butterfly images with dimensions of 224×224 pixels, dividing it into 80% for training and 20% for testing. The test results demonstrated an accuracy of 97.5%. In a previous study [23], authors proposed the CNN VGG-16 architecture. It was implemented to identify ten butterfly species with a total dataset of 832 images, divided into 80% training and 20% validation. Each image in the dataset is resized to dimensions of 224×224 pixels. The research managed to get a high accuracy of 97%. However, when attempting to recognize butterflies with a larger number of classes, as demonstrated in another research [3], the accuracy results were less satisfactory, although not indicative of overfitting. The experimental results showed an accuracy of 80.4% and 79.5%, respectively, for training and testing on 104 classes.

The model proposed in the research [3] is VGG16, which has been proven to outperform VGG19 and ResNet. Study [13] compared the performance of models built based on transfer learning with VGG16, VGG19, MobileNet, Xception, ResNet50, and InceptionV3 architectures. The dataset used consist of 4,011 images divided into 15 classes, with an unequal number of images in each class. The dataset is divided into 1761 images for training, 1125 for validation, and 1125 for testing, all with a size of 350×350 pixels. This research obtained the highest result using the InceptionV3 model, which was 94.66%. InceptionV3 is also used in other studies, such as [25]–[27]. Research [27] recognized dog breeds in 20 and 133 classes. Each class comprises 60 images divided into 40 training and 20 testing images. The recognition accuracy for the 20 classes reached an impressive 96.7%, while for the 133 classes, the accuracy achieved was 89.5%. Another research study [25], focus on recognizing Diabetic Retinopathy images, achieving an accuracy of 90.9%. Whereas in research [26], InceptionV3 was developed to detect fatigue and sleepiness.

The incorporation of Data Augmentation (DA) in the CNN method is theoretically known to enhance accuracy. This notion is supported by empirical evidence from[14],], where the DA can increase 12% in accuracy. An improved VGG16 model was also proposed in the study. Improvements to VGG16 were carried out by reducing the network depth, specifically by reducing the number of parameters. This modification aims to address the challenges of underfitting and overfitting during training, while also reducing training time compared to other CNNs. The proposed model achieves an accuracy of 89%. The effectiveness of DA has also been observed in other studies. For instance, in [33], DA improved the recognition accuracy of handwritten images by more than 10%. Additionally, research [34] also proves that augmentation data can increase the accuracy by approximately 1-5% in four texture image datasets using the ResNet50 model.

Based on the aforementioned literature, it can be concluded that improved pre-trained models such as InceptionV3, ResNet50, and VGG16 can enhance model accuracy. Similarly, the utilization of DA can mitigate underfitting and overfitting issues, consequently improving accuracy. However, in the specific context of the butterfly dataset, the InceptionV3 model has exhibited superior performance. Therefore, this study proposes a CNN model by enhancing the InceptionV3, and combining it with DA for butterfly image recognition involving a larger number of classes. Further details regarding the design of the proposed CNN model are presented in Section 3.

3. Proposed Method

The recognition methodology showcased in Figure 1 primarily encompasses a series of steps: the ingestion of data and its subsequent augmentation, followed by the processes of training, validation, testing, and a final evaluation. However, the proposed method contributes to enhancing the pre-trained model InceptionV3, as described in Section 3.1, and data augmentation, as explained in Section 3.2. Additionally, the proposed model is designed to recognize a larger number of classes with high accuracy compared to the prior art.

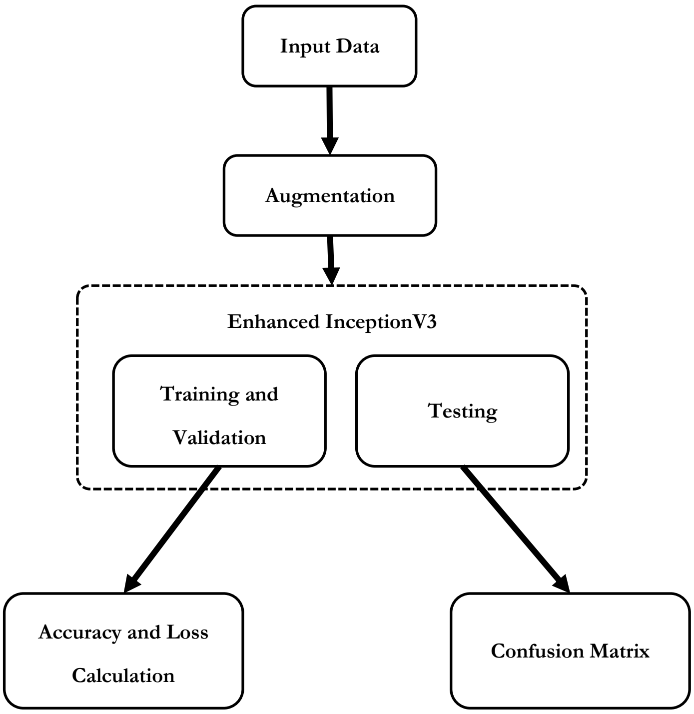

Figure 1. Proposed method.

3.1 Enhanced InceptionV3

By default, InceptionV3 has 313 layers, but in this study, the number of layers was reduced to only 279. The 279 layers of InceptionV3 are taken sequentially from the initial layer (input_4) to the 279th layer (mixed9_0). These layers are used as the initial feature extractor, consisting of convolution, activation, Batch Normalization, and average pooling layers. Furthermore, four new layers were added as learning features: flatten, dropout and two dense. The function of flattened layer is to convert multi-dimensional feature maps into single dimensions. The dropout layer in the network serves the purpose of randomly deactivating neurons during the training process. This ensures the even distribution of feature selection across all neurons. Moreover, it aids in minimizing overfitting by reducing the parameters in the model. Meanwhile, the dense layer, also known as the fully connected layer unifies nodes into a single dimension. This requires the preceding layer's output to determine features that correlate with a class, as vividly depicted in Table 1. The rationale behind the removal of these layers is to prevent the model from using the entirety of the default weight, as it may lead to excessive complexity due to the discrepancy between the default number of classes in InceptionV3 training and the requirements of the research [35]. InceptionV3 uses approximately

25 million parameters, but in this study, the number of parameters was significantly reduced to around 9.7 million. This significant reduction is an important contribution, especially for lightweight implementations, as it enhances model efficiency and mitigates overfitting with fewer classes [36].

Table 1. Proposed enhanced InceptionV3 model.

Layer (type)	Output Shape	Param
input_4 (InputLayer)	[(None, 224, 224, 3)]	0
conv2d_282 (Conv2D)	(None, 111, 111, 32)	864
batch_normalization_283	(None, 111, 111, 32)	96
activation_282 (Activation)	(None, 111, 111, 32)	0
conv2d_283 (Conv2D)	(None, 109, 109. 32)	9216
batch_nomalization_283 (Batch Normalization)	(None, 109, 109, 32)	96
activation_283 (Activation)	(None, 109, 109, 32)	0
conv2d_284 (Conv2D)	(None, 109, 109, 64)	18432
batch_normalization_284 (Batch Normalization)	(None, 109, 109, 64)	192
activation_284 (Activation)	(None, 109, 109, 64)	0
...	...	...
...	...	...
...	...	...
mixed9_0 (Concatenate)	(None, 5, 5, 768)	0
flatten	(None, 19200)	0
dense (Dense)	(None, 512)	9830912
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 100)	51300

3.2 Data Augmentation

Data augmentation is a useful preprocessing step to improve the quality of the dataset. The quality of the dataset directly influences the quality of input for both the training and testing processes. In deep learning tasks, large datasets are generally required to prevent over-fitting. Although data augmentation can often mitigate this problem, over-augmentation may sometimes lead to suboptimal results. In this study, three types of augmentation were selected, namely rotation, flip and zoom. These were selected as they tend not to significantly alter the object's shape, nor do they introduce distortion effects that could change the object's meaning or appearance within the image.

4. Implementation and Analysis

This study uses Python as a programming language and Jupyter Notebook using GPU GTX 1050 with the help of CUDA. While the Butterfly & Moths Image Classification dataset is 100 species [37], the image format specification is JPEG 24 bits with dimensions of 224×224. There are a total of 13594 images, which are divided into 12594 training images and 500 validation and testing images each. A list of 100 butterfly species contained in the dataset [37] is presented in Table 2.

The next stage is the augmentation process to produce a more optimal model and avoid overfitting. This process is carried out using the ImageDataGenerator function within the Keras API, which offers various parameters for generating augmented images. This study selected augmentation using rotation, zoom, and flip techniques, with the parameters shown in Table 3. Meanwhile, the sample augmentation results are presented in Figure 2. In addition to enhancing the augmentation function, we use the fill_mode='nearest' parameter. This function is used to fill empty areas with the value of the closest pixel. In addition, before entering the training, testing, and validation stages, all images are rescaled with the parameter rescale = 1/255. The purpose of the rescale function is to normalize image pixels and facilitate learning, accelerate convergence, uniform pixel scale, and avoid saturation. As a result of this augmentation, which is applied exclusively to training and validation images, the total dataset expands to 52,876 records.

Table 2. Butterflies class list.

No	Class/Species	No	Class/Species	No	Class/Species
1	Adonis	35	Crimson Patch	69	Orange Tip
2	African Giant Swallowtail	36	Danaid Eggfly	70	Orchard Swallow
3	American Snoot	37	Eastern Coma	71	Painted Lady
4	An 88	38	Eastern Dapple White	72	Paper Kite
5	Appollo	39	Eastern Pine Elfin	73	Peacock
6	Arcigera Flower Moth	40	Elbowed Pierrot	74	Pine White
7	Atala	41	Emperor Gum Moth	75	Pipevine Swallow
8	Atlas Moth	42	Garden Tiger Moth	76	Polyphemus Moth
9	Banded Orange Heliconian	43	Giant Leopard Moth	77	Popinjay
10	Banded Peacock	44	Glittering Sapphire	78	Purple Hairstreak
11	Banded Tiger Moth	45	Gold Banded	79	Purplish Copper
12	Beckers White	46	Great Eggfly	80	Question Mark
13	Bird Cherry Ermine Moth	47	Great Jay	81	Red Admiral
14	Black Hairstreak	48	Green Celled Cattleheart	82	Red Cracker
15	Blue Morpho	49	Green Hairstreak	83	Red Postman
16	Blue Spotted Crow	50	Grey Hairstreak	84	Red Spotted Purple
17	Brookes Birdwing	51	Hercules Moth	85	Rosy Maple Moth
18	Brown Argus	52	Hummingbird Hawk-Moth	86	Scarce Swallow
19	Brown Siproeta	53	Indra Swallow	87	Silver Spot Skipper
20	Cabbage White	54	Io Moth	88	Sixspot Burnet Moth
21	Cairns Birdwing	55	Iphiclus Sister	89	Sleepy Orange
22	Chalk Hill Blue	56	Julia	90	Sootywing
23	Chequered Skipper	57	Large Marble	91	Southern Dogface
24	Chestnut	58	Luna Moth	92	Straited Queen
25	Cinnabar Moth	59	Madagascan Sunset Moth	93	Tropical Leafwing
26	Clearwing Moth	60	Malachite	94	Two Barred Flasher
27	Cleopatra	61	Mangrove Skipper	95	Ulyses
28	Clodius Parnassian	62	Mestra	96	Viceroy
29	Clouded Sulphur	63	Metalmark	97	White Lined Sphinx Moth
30	Comet Moth	64	Milberts Tortoiseshell	98	Wood Satyr
31	Common Banded Awl	65	Monarch	99	Yellow Swallow Tail
32	Common Wood-Nymph	66	Mourning Cloak	100	Zebra Long Wing
33	Copper Tail	67	Oleander Hawk Moth		
34	Crecent	68	Orange Oakleaf		

Table 3. Augmentation Parameters.

No	Augmentation Type	Parameter
1	rotation_range	40
2	zoom_range	0.2
3	horizontal_flip	True

The initial test carried out only on fifteen classes, which is the same as the research [13], aims to compare results. The fifteen classes are presented in Figure 3. After testing, the proposed model has very good performance with only ten epochs, namely obtaining a maximum accuracy of 99.43% with the best loss of 3.7%, note Figure 4. For the record, several hyperparameters were used, namely optimizer = Adamax(learning_rate=0.001), loss = 'categorical_crossentropy', metrics = ['accuracy'], besides, we opted for ten epochs to maintain consistency with the research[13].

Figure 2. Sample Augmentation Results.

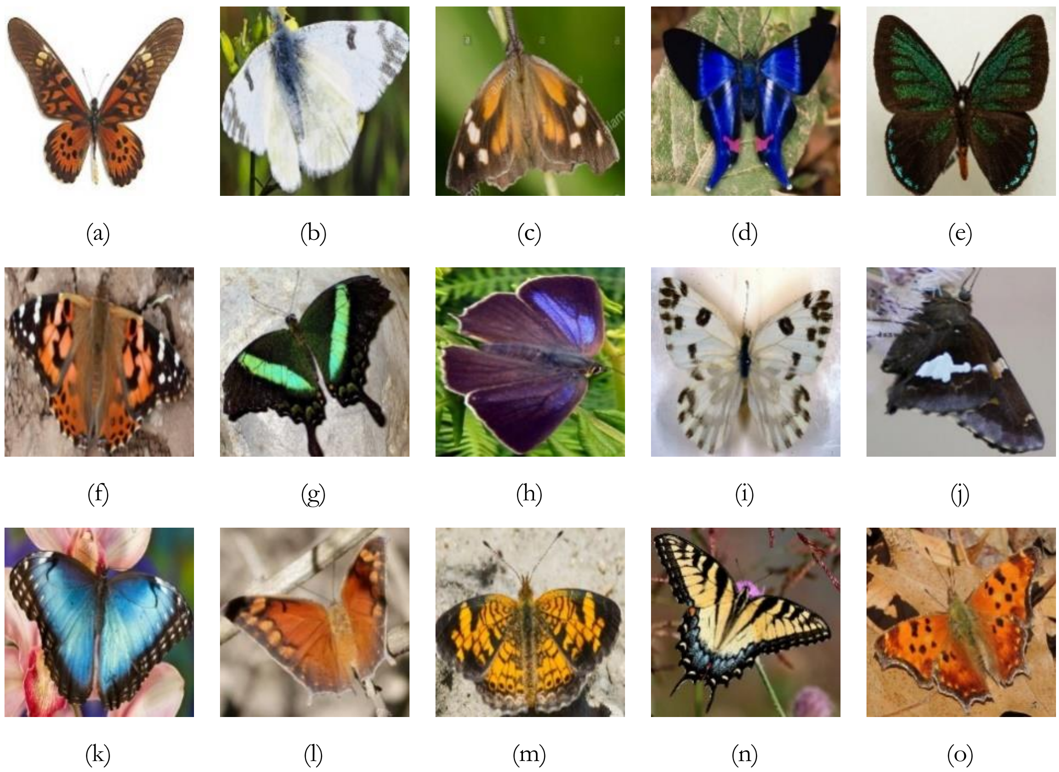

Figure 3. Fifteen species in the first test recognition{(a) African Giant Swallowtile (b) White Marble (c) American Snoot (d) Metalmark (e) Atala (f) Painted Lady (g) Banded Peacock (h) Purple Hairstreak (i) Beckers White (j) Silver Spot Skipper (k) Blue Morpho (l) Tropical Leafwing (m) Crecent (n) Yellow Shallow Tile (o) Eastern Coma}.

As previously outlined, this study included a comprehensive comparison with other pre-trained models such as VGG16, RestNet50, and the standard InceptionV3, all of which are graphically represented in Figure 5.The data indicates that the standard pre-trained model, InceptionV3, exhibits the best results, corroborating the findings published in research [13]. However, a deeper evaluation of our proposed method's performance reveals it surpasses the results mentioned in the research [13] by approximately 5%.

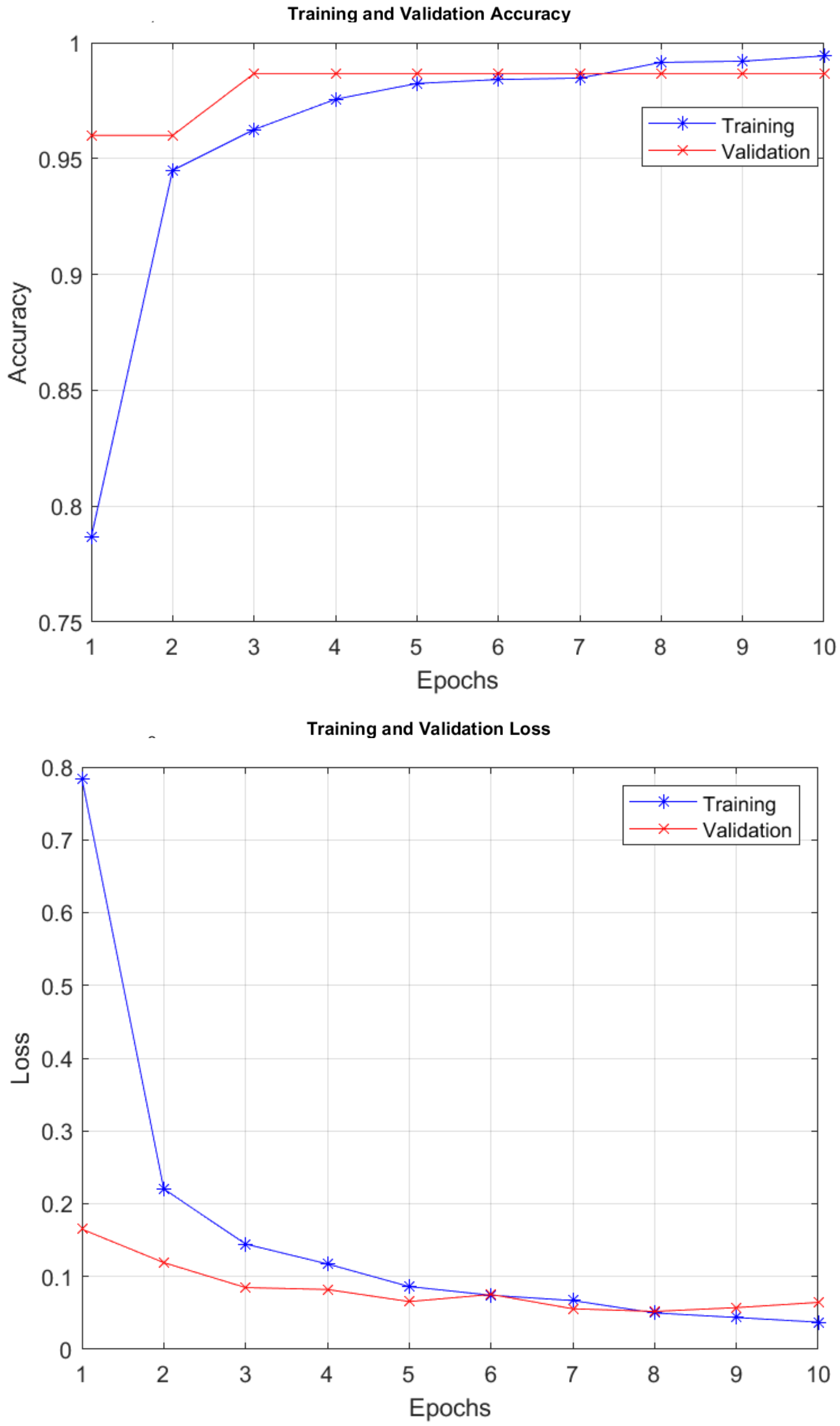

Figure 4. Accuracy and Loss Results Modified InceptionV3 (First test for fifteen class/Model 1) Butterflies Recognition

Next, the model evaluation is carried out using test data, the results are presented with a classification report and a confusion matrix. Test evaluation is only carried out on the modified InceptionV3 because it produces the best performance in validation and training. Testing on 500 test images results are presented in Figure 6. Based on the results of the confusion matrix presented, there were only three classes that experienced wrong predictions with one error each, these classes included American snoot, crescent, and purple hairstreak. Consequently, the resulting accuracy, precision, recall, and F1-score are 0.96, 0.97, 0.96, and 0.96, respectively. Where accuracy, precision, recall, and F1-score are calculated by Eq. (1)-(4), respectively[38].

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{4}$$

Where TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

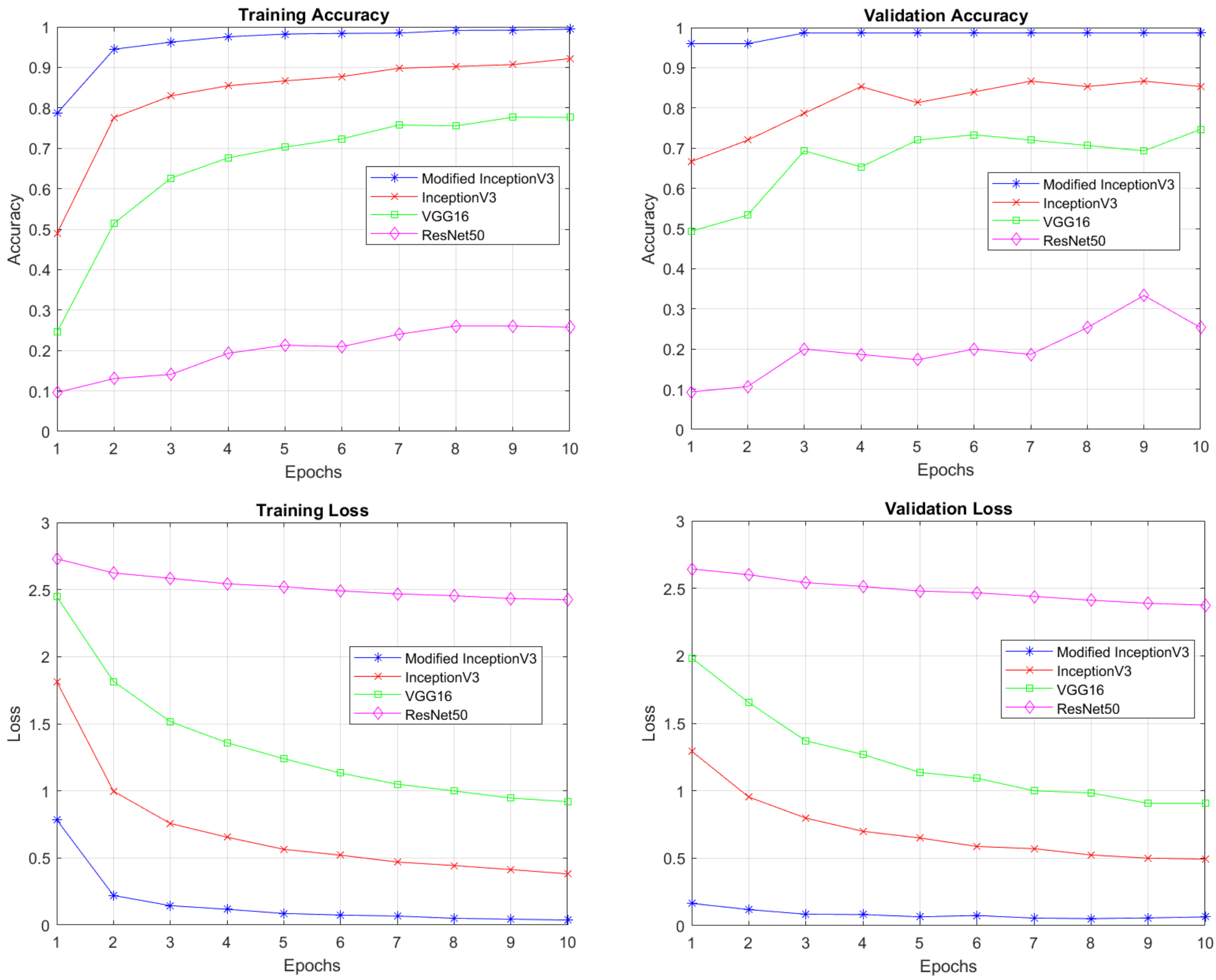

Figure 5. Comparison with pre-trained models

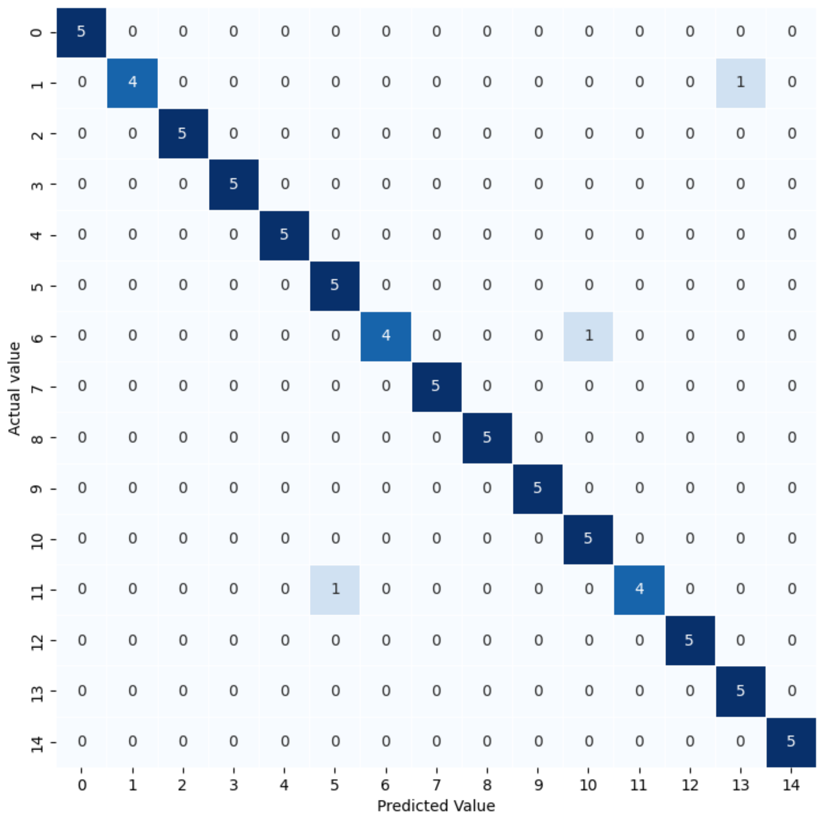

Figure 6. Confusion matrix from testing results

The training, validation, and testing results indicate that the proposed method is reliable and doesn't exhibit overfitting. This can be attributed to the reduction in the number of layers from standard InceptionV3, coupled with the addition of four extra layers, as previously detailed in Table 1. This strategy not only reduces complexity but also mitigates overfitting, especially when dealing with relatively small datasets. This helps the model to be more general and avoids the tendency to "memorize" the training data. Finally, an additional design layer is connected to the dropout layer to produce the final output.

Lastly, the testing stage is carried out only on the proposed model, namely the Modified InceptionV3. This test involves 100 classes of butterflies, aiming to ascertain the classification performance with larger datasets and more classes. Given the increased number of datasets and classes, we conducted three tests with varying epochs: 10, 20, 30, and 50. In accordance with the theory, there is a slight decrease in performance, but the results remain satisfactory, as shown in Table 4. We opted for a larger epoch than used in the 15-class recognition to enhance the accuracy performance. There appears to be a performance improvement with an increase in epochs. A 4.99% difference between the 10th and 50th epochs is observed, and the accuracy value tends to stabilize. The graphs representing the training and validation processes for 50 epochs are presented in Figure 7. Furthermore, based on the classification report and confusion matrix from the recognition of 100 classes, the resulting accuracy, precision, recall, and F1-score are 0.97, 0.98, 0.97, and 0.97, respectively. These results validate that the proposed CNN model tends to be stable and less overfitting in validation and testing.

Table 4. 100 Class Butterflies Recognition with Epoch Variation.

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
10	93.50%	92.00%	24.26%	28.75%
20	96.31%	94.40%	13.12%	28.30%
30	97.23%	96.20%	9.14%	19.56%
50	98.49%	97.80%	5.21%	9.78%

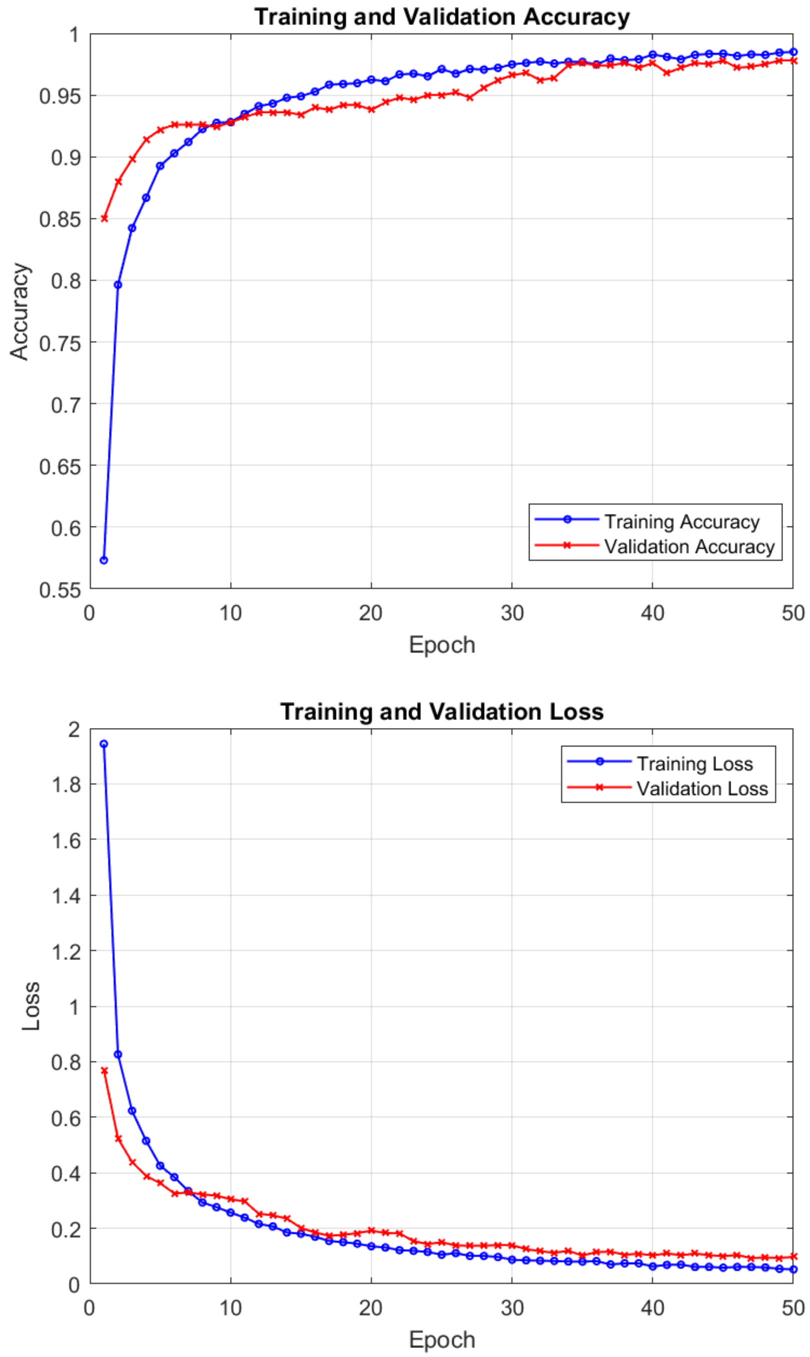

Figure 7. Accuracy and Loss Graph of Training and Validation in 100 Class Recognition (Model 2)

Figure 8 shows a comparison of the training accuracy of the models that have been proposed in butterfly recognition. The performance of the proposed method appears to be the most superior, where it is 98.49% and 99.43%, respectively, for the recognition of 15 and 100 classes. Even though the advantages seem insignificant compared to models [22], [23] these two models each only recognize a small number of classes, namely 4 and 10. Whereas Model [13] and proposed Model 1 used 15 classes, and proposed Model 1 excelled at more than 4%. Whereas in the Model [3], the recognition was carried out in 104 classes, while the proposed model 2 recognized 100 classes and excelled at 18%, even though the training process was carried out for up to 100 epochs, while the proposed model 2 only had 50 epochs. These results affirm that InceptionV3, by default, serves as a suitable pre-trained model for butterfly image recognition, and with the implementation of our proposed enhancements, the model's accuracy performance can be optimized further.

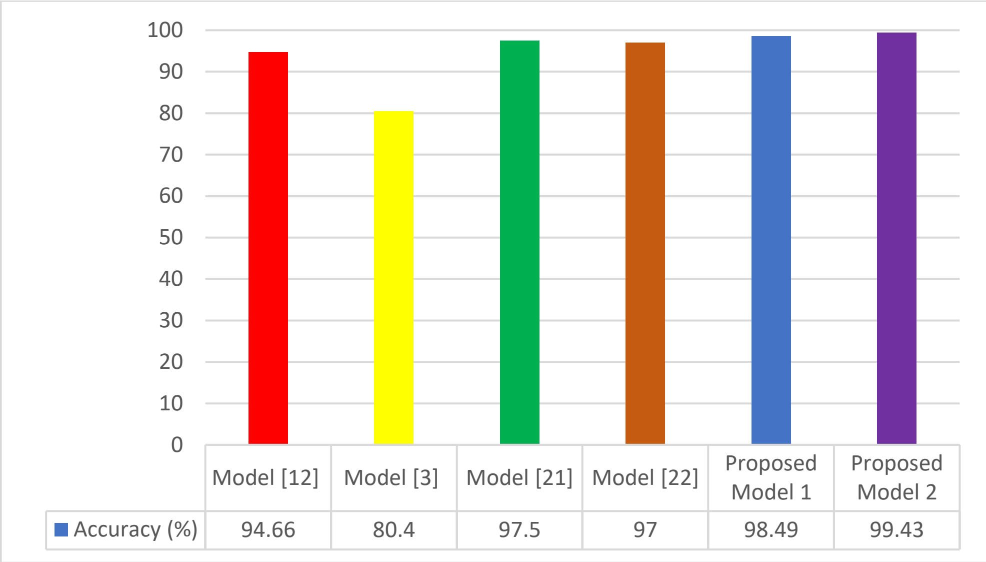

Figure 8. Comparison accuracy with related works

5. Conclusions

This research has succeeded in enhancing the InceptionV3 pre-trained model for butterfly image recognition. The InceptionV3 model was modified by reducing and adding layers to better align with the dataset's characteristics. In addition, layer reduction and data augmentation can affect more effective learning, so the accuracy is much better than the standard model. Based on the test results, the enhanced model demonstrates outstanding and stable performance in classifying butterflies across smaller and larger classes. Recognition has been successfully carried out on the largest available dataset, comprising 100 classes. With an accuracy of 99.43% and minimal overfitting, the results suggest that the proposed method is reliable and holds potential for future improvements. This method could be tested on an even larger and more comprehensive dataset, making it more realistic when applied to practical applications.

Author Contributions: Conceptualization: H.T.A. and D.R.I.M.S.; methodology: H.T.A. and D.R.I.M.S.; software: H.T.A. and H.M.M.I.; validation: All; formal analysis: All; investigation: O.F., D.R.I.M.S; resources: H.T.A. and D.R.I.M.S.; data writing—original draft preparation: H.T.A. and D.R.I.M.S.; writing—review and editing: O.F., S.S., H.M.M.I., and M.K.S; visualization: H.M.M.I.; supervision: D.R.I.M.S.; project administration: S.S; funding acquisition: All.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] J. A. Fordyce, "Butterflies of the World By Adrian Hoskins," *Q. Rev. Biol.*, vol. 95, no. 2, pp. 137–137, Jun. 2020, doi: 10.1086/709040.
[2] C. L. Boggs, W. B. Watt, and P. R. Ehrlich, *Butterflies*. University of Chicago Press, 2003. doi: 10.7208/chicago/9780226063195.001.0001.
[3] A. S. Almryad and H. Kutucu, "Automatic identification for field butterflies by convolutional neural networks," *Eng. Sci. Technol. an Int. J.*, vol. 23, no. 1, pp. 189–195, 2020, doi: 10.1016/j.jestch.2020.01.006.
[4] J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases," *Front. Plant Sci.*, vol. 10, Jul. 2019, doi: 10.3389/fpls.2019.00941.
[5] M. S. Sunarjo, H. Gan, and D. R. I. M. Setiadi, "High-Performance Convolutional Neural Network Model to Identify COVID-19 in Medical Images," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 19–30, Aug. 2023, doi: 10.33633/jcta.v1i1.8936.
[6] S. B. Imanulloh, A. R. Muslikh, and D. R. I. M. Setiadi, "Plant Diseases Classification based Leaves Image using Convolutional Neural Network," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 1–10, Aug. 2023, doi: 10.33633/jcta.v1i1.8877.

[7] L. Yang, Z. Wu, T. Xu, J. Du, and E. Wu, "Easy recognition of artistic Chinese calligraphic characters," *Vis. Comput.*, vol. 39, no. 8, pp. 3755–3766, 2023, doi: 10.1007/s00371-023-03026-2.
[8] T. A. Pham, "Effective deep neural networks for license plate detection and recognition," *Vis. Comput.*, vol. 39, no. 3, pp. 927–941, 2023, doi: 10.1007/s00371-021-02375-0.
[9] A. E. Sebastian and D. Dua, "Lung Nodule Detection via Optimized Convolutional Neural Network: Impact of Improved Moth Flame Algorithm," *Sens. Imaging*, vol. 24, no. 1, 2023, doi: 10.1007/s11220-022-00406-1.
[10] A. Khaparde, V. Deshmukh, and M. Kowdiki, *Enhanced Nature-Inspired Algorithm-based Hybrid Deep Learning for Character Recognition in Sanskrit Language*, vol. 24, no. 1. Springer US, 2023. doi: 10.1007/s11220-023-00421-w.
[11] A. P. Wibowo and D. R. I. M. Setiadi, "Performance Analysis of Deep Learning Models for Sweet Potato Image Recognition," in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2022, pp. 101–106. doi: 10.1109/iSemantic55962.2022.9920423.
[12] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental Concepts of Convolutional Neural Network," 2020, pp. 519–567. doi: 10.1007/978-3-030-32644-9_36.
[13] F. Rajeena P. P. *et al.*, "A Novel Method for the Classification of Butterfly Species Using Pre-Trained CNN Models," *Electronics*, vol. 11, no. 13, p. 2016, Jun. 2022, doi: 10.3390/electronics11132016.
[14] Z.-P. Jiang, Y.-Y. Liu, Z.-E. Shao, and K.-W. Huang, "An Improved VGG16 Model for Pneumonia Image Classification," *Appl. Sci.*, vol. 11, no. 23, p. 11185, Nov. 2021, doi: 10.3390/app112311185.
[15] Y. Ge, T. Zhang, H. Liang, Q. Jiang, and D. Wang, "A novel technique for image steganalysis based on separable convolution and adversarial mechanism," *Electron.*, vol. 10, no. 22, pp. 1–15, 2021, doi: 10.3390/electronics10222742.
[16] M. M. H. Matin, A. Khatun, M. G. Moazzam, and M. S. Uddin, "An Efficient Disease Detection Technique of Rice Leaf Using AlexNet," *J. Comput. Commun.*, vol. 08, no. 12, pp. 49–57, 2020, doi: 10.4236/jcc.2020.812005.
[17] R. F. Mattins, M. V. R. Sarobin, A. A. Aziz, and S. Srivarshan, "Object detection and classification of butterflies using efficient CNN and pre-trained deep convolutional neural networks," *Multimed. Tools Appl.*, no. 0123456789, Nov. 2023, doi: 10.1007/s11042-023-17563-4.
[18] R. Rashmi, U. Snekhalatha, A. L. Salvador, and A. N. J. Raj, "Facial emotion detection using thermal and visual images based on deep learning techniques," *Imaging Sci. J.*, pp. 1–14, Apr. 2023, doi: 10.1080/13682199.2023.2199504.
[19] S. Zhao, Z. Gong, and D. Zhao, "Traffic signs and markings recognition based on lightweight convolutional neural network," *Vis. Comput.*, 2023, doi: 10.1007/s00371-023-02801-5.
[20] S. Karthi, L. R. Sudha, and M. Navaneetha Krishnan, "A Novel Hybrid Optimization Enabled Densenet for Covid-19 Classification using CT Images," *Sens. Imaging*, vol. 24, no. 1, pp. 1–27, 2023, doi: 10.1007/s11220-023-00434-5.
[21] R. Mehta and K. K. Singh, "Deep convolutional neural network-based effective model for 2D ear recognition using data augmentation," *Imaging Sci. J.*, pp. 1–18, May 2023, doi: 10.1080/13682199.2023.2206763.
[22] N. N. Kamaron Arzar, N. Sabri, N. F. Mohd Johari, A. Amilah Shari, M. R. Mohd Noordin, and S. Ibrahim, "Butterfly Species Identification Using Convolutional Neural Network (CNN)," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Jun. 2019, no. June, pp. 221–224. doi: 10.1109/I2CACIS.2019.8825031.
[23] S. N. Hegde, P. Sindhura, R. Rodrigues, and B. Student, "Identification of Butterfly Speciecs Using Vgg-16," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 2, no. 05, pp. 1–5, 2020, [Online]. Available: www.irjmets.com
[24] A. Parashar, R. S. Shekhawat, W. Ding, and I. Rida, "Intra-class variations with deep learning-based gait analysis: A comprehensive survey of covariates and methods," *Neurocomputing*, vol. 505, pp. 315–338, Sep. 2022, doi: 10.1016/j.neucom.2022.07.002.
[25] M. T. Hagos and S. Kant, "Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset," May 2019, [Online]. Available: http://arxiv.org/abs/1905.07203
[26] Y. Zhao, K. Xie, Z. Zou, and J.-B. He, "Intelligent Recognition of Fatigue and Sleepiness Based on InceptionV3-LSTM via Multi-Feature Fusion," *IEEE Access*, vol. 8, pp. 144205–144217, 2020, doi: 10.1109/ACCESS.2020.3014508.
[27] P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn, and W. Kusakunniran, "Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning," in *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct. 2019, pp. 1–5. doi: 10.1109/ICITEED.2019.8929955.
[28] S. Guefrechi, M. Ben Jabra, A. Ammar, A. Koubaa, and H. Hamam, "Deep learning based detection of COVID-19 from chest X-ray images," *Multimed. Tools Appl.*, vol. 80, no. 21–23, pp. 31803–31820, Sep. 2021, doi: 10.1007/s11042-021-11192-5.
[29] T. Vijayan, M. Sangeetha, and B. Karthik, "Efficient analysis of diabetic retinopathy on retinal fundus images using deep learning techniques with inception V3 architecture," *J. Green Eng.*, vol. 10, no. 10, pp. 9615–9625, 2020.
[30] L. Ali, F. Alnajjar, H. Al Jassmi, M. Gocho, W. Khan, and M. A. Serhani, "Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures," *Sensors*, vol. 21, no. 5, p. 1688, Mar. 2021, doi: 10.3390/s21051688.
[31] D. Theckedath and R. R. Sedamkar, "Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks," *SN Comput. Sci.*, vol. 1, no. 2, p. 79, Mar. 2020, doi: 10.1007/s42979-020-0114-9.
[32] T. Surabhi, B. Sachin, and C. Advait, "Deep Convolutional Neural Networks for Automated Butterfly Species Recognition and Classification," in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Aug. 2023, pp. 778–783. doi: 10.1109/ICIRCA57980.2023.10220696.
[33] A. Susanto, I. U. Wahyu Mulyono, C. Atika Sari, E. Hari Rachmawanto, D. R. I. M. Setiadi, and M. K. Sarker, "Handwritten Javanese script recognition method based 12-layers deep convolutional neural network and data augmentation," *IAES Int. J. Artif. Intell.*, vol. 12, no. 3, p. 1448, Sep. 2023, doi: 10.11591/ijai.v12.i3.pp1448-1458.
[34] L. Nanni, M. Paci, S. Brahnam, and A. Lumini, "Comparison of different image data augmentation approaches," *J. Imaging*, vol. 7, no. 12, 2021, doi: 10.3390/jimaging7120254.
[35] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023, doi: 10.3390/computation11030052.

[36] H. D. Prasetyo, H. Triatmoko, Nurdiansyah, and I. N. Isnainiyah, "The Implementation of CNN on Website-based Rice Plant Disease Detection," in *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, Nov. 2020, pp. 75–80. doi: 10.1109/ICIMCIS51567.2020.9354329.
[37] "Butterfly & Moths Image Classification 100 species | Kaggle." https://www.kaggle.com/datasets/gpiosenka/butterfly-images40-species (accessed Jul. 23, 2023).
[38] F. Mustofa, A. N. Safriandono, A. R. Muslikh, and D. R. I. M. Setiadi, "Dataset and Feature Analysis for Diabetes Mellitus Classification using Random Forest," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 41–48, Sep. 2023, doi: 10.33633/jcta.v1i1.9190.