JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

CONVOLUTIONAL NEURAL NETWORK APPROACH FOR IDENTIFICATION OF BUTTERFLY SPECIES

Renuka Chalwa[#1], Ashvini Dhage[#2], Akshada Mhaske[#3], R.L.Nandargi[#4], B.C. Walimbe[#5]

Department Of Electronics and Telecommunication Engineering
SKNCOE, SPPU, Pune, Maharashtra, India,
[1]renukachalwa@gmail.com
[2]ashudhage70@gmail.com
[3]mhaskeakshada491@gmail.com
[4]rajshekhar.nandargi@gmail.com
[5]bhakti.lomte@gmail.com

Abstract-**There are thousands of butterfly species in the environment. Classification and identification is very important for scientists for study of species. There are many old techniques available for identification of butterflies but they are not effective enough and are time consuming. We need faster and more reliable methods. In the proposed system, a model for butterfly identification using deep learning will be developed. The proposed system will use a database of various butterfly species. For classification and identification of butterfly species, neural network technology like CNN and ANN will be used.**

Keywords: **Vgg16, CNN, Vgg19, deep learning , butterfly species**

I. INTRODUCTION

Butterfly research is among the most important studies in the field of biology, and it is being conducted all over the world. Lepidopterists are researchers and students who conducted this observation. The scientific name of the butterfly is Lepidoptera. Lepidoptera is an Animalia order that includes moths and butterflies. The scales that cover the entire body, including the bodies, wings, and proboscis, are one of the most intriguing features of the lepidoptera. The lepidoptera family includes a huge number of species, each with its own unique wing shape and colour, making identification tricky. As a result, a tool must be developed in tandem to lessen the pressure of routine butterfly identification. The targeted butterfly species must be manually captured by trap during the traditional butterfly identification process. This not only exhausts time and energy, but it is also damaging to the butterfly because it will be confined in tiny spaces for an extended period of time in order to identify differences in the body and wing. Not just that, the attributes of the butterfly have to be categorised using the encyclopaedia, which is large and not easily accessible.

Distinguishing between butterfly species necessarily involves expertise and time, both of which are not always available, but with the development of software that identifies butterfly species by extracting features from images, the need for experts will be reduced. There are two major issues in existing computer vision-based butterfly species identification . For starters, collecting the butterfly dataset is difficult, identifying butterflies is time-consuming job for entomologists, and the number of butterflies included in the butterfly dataset is insufficient. Second, the butterfly images used in training are all pattern images with obvious morphological traits, rather than ecological images of butterflies in surroundings. Further to that, the differences between the two images are obvious, finding it challenging to combine research and production, and the detection performance is low.
The difficulty of carrying an encyclopaedia can be overcome by using the database that is included in the application, thanks to the introduction of a useful mobile application. Since smartphones are now a daily accessory for humans, people would every time manage to keep a smartphone with themselves. It's helpful for them because they don't have to keep in mind to carry specific equipment or tools while going out for research purpose .

The goal of this project is to propose a framework for recognising digital image of butterflies and classifying them using a trained dataset. Butterfly identification is useful for a lot of reasons. One of it's main motives is to survey and record butterflies. Butterfly monitoring is carried out in camps, where the identified butterfly is recorded and the species is counted . The introduction of an automated system to reduce the burden of routine identification is required to assist in the education department.

II. LITERATURE SURVEY:

Ayad Saad Almryad et al. [1] present the Automatic identification for field butterflies by convolutional neural networks (CNN). This system collected 44,659 images of 104 different butterfly species taken with varying positions of butterflies, the shooting angle, butterfly distance, occlusion, and background complexity in the field in Turkey. Since many species have a few image samples, we constructed a field-based dataset of 17,769 butterflies with 10 species. CNNs were used for the identification of butterfly species. Comparison and evaluation of the experimental results obtained using three different network structures are conducted. Experimental results on 10 common butterfly species showed that vgg16, vgg19, and ResNet achieved a training and testing accuracy of 80.4%, 77.6%, 84.8%, and 79.5%, 77.2%, 70.2%, respectively.

Omer Faruk Ertugrul et al. [2] presented Law's texture energy measure method to identify butterfly species as an alternative to conventional diagnostic techniques and other image processing methods. Mean, standard deviation, and entropy of filtered images were used as a texture feature set of the butterflies. The best suitable features were used for classification with kNN, SVM, and ELM, which were also optimized for the butterfly dataset, with 99.26%, 98.16%, and 99.47%, respectively. These findings suggest that the ELM algorithm and Law's texture energy technique are feasible and excellent for identifying and classifying butterfly species.

Badrul Aiman Bakri et al. [3] presents a computer vision study on how computers can make high-level comprehension from the input of digital image and videos. Utilizing the latest Image Processing technique can identify the correct butterfly species with high accuracy by using layers of a node in a Convolutional Neural Network (CNN). The work process starts with data acquisition, pre-processing, analysing and understanding digital images, and making assumptions of the high complication data from the real world in producing numerical information that machines can comprehend to form conclusions. The project is developed using TensorFlow in Ubuntu operating system. The interface is in HTML connected to the Python script via Flask. The experiment results show that CNN can identify with 92.7% of absolute accuracy with learning saturation (overfitting) of 500 cycles. While testing results show 62.5 percent of accuracy in predicting new datasets.

Keanu Buschbacher et al. [4] propose DeepABIS based on the concepts of the successful Automated Bee Identification System (ABIS), which allowed mobile field investigations, including species identification of live bees in the field. Deep-ABIS features three critical advancements. First, DeepABIS reduces the training of the system by employing automated feature generation using deep convolutional networks (CNN). Second, DeepABIS enables participatory sensing scenarios utilizing mobile smartphones and a cloud-based platform for data collection and communication. Third, DeepABIS is adaptable and transferable to other taxa beyond Hymenoptera, i.e., butterflies, flies, etc. Current results show identification results with an average top-1 accuracy of 93.95% and a top-5 accuracy of 99.61% applied to the data material of the ABIS project. They are adapting DeepABIS to a butterfly dataset showing morphologically difficult to separate populations of the same species of butterfly yields identification results with an average top-1 accuracy of 96.72% and a top-5 accuracy of 99.99%.

Nur Nabila Kamaron Arzar et al. [5] presents the study of butterfly species identification using image processing technique, and Convolution Neural Network (CNN) is proposed. This research focuses on GoogLeNet, a pre-trained model of CNN architecture. Four species of butterflies commonly found in Asia: Black Veined Tiger, Chocolate Grass Yellow, Grey Pansy, and Plain Lacewing used in this research. The testing conducted reflected 97.5% overall identification accuracy on one hundred and twenty images of four types of butterflies.

Aijiao Tan, Guoxiong Zhou et al. [6] proposed FCM-KM and Mask R-CNN fusion to realize the location and recognition of butterflies by robot vision system in a complex environment, rapid fine-grained classification of butterflies. First, an adaptive image enhancement algorithm based on fuzzy sets optimized by FOA was used to realize the adaptive fuzzy enhancement of butterfly images in image pre-processing. Then, the K-Means clustering algorithm optimized by dynamic population butterfly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, was used to determine the optimal clustering number K instead of manual parameter tuning. Finally, while effectively segmenting the butterfly images, the SoftMax in Mask R-CNN was used to classify the butterfly images. The recognition accuracy of the trained model in the verification set was 83.62%. To verify the feasibility and effectiveness of the model in a complex environment, the rapid _ne-grained classification method of butterflies based on the FCM-KM and Mask R-CNN fusion was compared with CNN, Resnet, and original Mask R-CNN. The experiment results show that there is a 3% increase in accuracy than the existing system.

Ruoyan Zhao et al. [7] uses the Faster R-CNN algorithm for butterfly recognition and describe the process from butterfly dataset selection and butterfly dataset processing to butterfly classification. The experimental results show that the butterfly automatic recognition system based on the Faster R-CNN deep learning framework can realize automatic detection and species identification of butterfly photos in the ecological environment. The average classification accuracy can reach 70.4%.

Xie Juanying et al. [8] built a butterfly image dataset composed of all species of butterflies in Monograph of Chinese butterflies with 4270 standard pattern images of 1176 butterfly species and 1425 butterfly images from the living environment of 111 species. They use Faster R-CNN's deep learning technique to develop an automatic butterfly identification system, including butterfly position detection in images from living environments and species recognition. To construct the training subset for Faster R-CNN, nine methods are adopted to amplify the images in the training subset, including the turning of up and down, and left and right, rotation with different angles, adding noises, blurring, and contrast ratio adjusting, etc. Three kinds of network structure-based prediction models are trained. The MAP (mean average prediction) criterion is used to evaluate the performance of the predictive models. The experimental results demonstrate that our Faster R-CNN-based butterfly automatic identification system performs well. Its worst MAP is up to 60%, and it can simultaneously detect the positions of more than one butterfly in one image from a living environment and can recognize their species as well.

Jan Carlo T. Arroyo [9] presents convolutional neural networks and transfer learning for classifying Coleoptera specimens. The images used as a dataset in this study were gathered from previous research work and various repositories. Four classes were used to train the convolutional neural network, with the Buprestid, Carbide, Cerambycid, and Coccinellidae families. Since the dataset was somewhat imbalanced, images were pre-processed to augment the dataset and minimize the probability of overfitting. Transfer learning was implemented by using the InceptionV3 pre-trained model. The final layer was retrained using the new dataset while retaining its prior knowledge base. After the training and validation of the new model, an average of 97% classification accuracy was attained.

Juanying Xie et al. [10] present some investigations have undertaken in this field over the past two years and show the results we have obtained. They propose a new partition and augmentation technique for the highly unbalanced ecological butterfly database. We found that Retina Net is, so far, the best deep learning algorithm to tackle butterfly species identification based on butterfly images taken in natural environments. The best result we obtained was 79.7% in terms of map (mean average precision). This is the best result compared to the state-of-the-art studies in this field on the same database so far.

III. Methodology:

Block diagram of the proposed system is shown in Fig.1

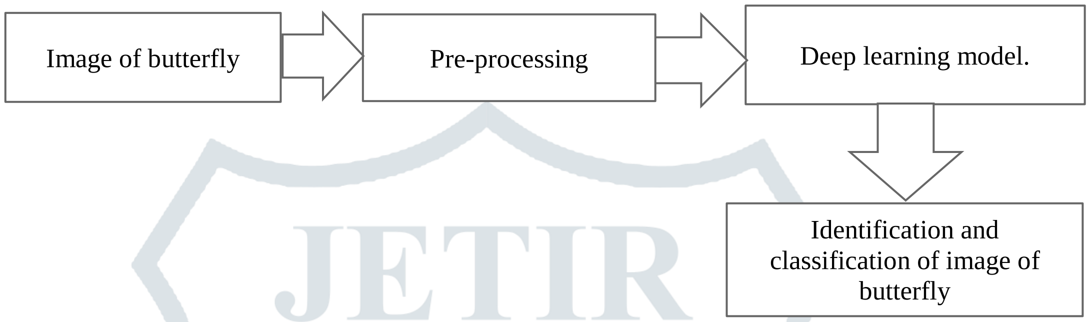


Fig. 1 Block diagram demonstrates the development process flow for butterfly identification Before pre-processing can begin, an image must be acquired. Following that, training will take place to train using the CNN model. The proposed system uses butterfly images from the collected database as input. Following pre-processing, butterfly images will be fed into the deep learning model, which has already been trained using 70% of the database. This model will both identify and categorise butterfly species. The accuracy percentage will be calculated after the training process. Finally, the identified butterfly image will be identified, and a detailed image will be displayed.

Fig. 1 Dataset:

The images were gathered from Kaggle. The dataset is made up of two parts: Train and Test. Set of validation data for 50 butterfly species. All images are in.jpg format and measure 224 X 224 X 3. The train set is made up of 4955 images that have been divided into 50 subdirectories, one for each species. The test set contains 250 images divided into 50 subdirectories, with 5 test images for each species. The validation set consists of 250 images divided into 50 subdirectories, each with five validation images.

Pre-processing

Cropping, resizing to a reasonable size, histogram equalisation for eliminating illumination variance, noise reduction, thresholding, converting to a binary or grayscale image, and so on may be required. The colour format of the input image is RGB. The RGB image is converted into a grayscale image for further processing.

$$Gray(G) = 0.30R + 0.69G + 0.11B \quad (1)$$

Data Augmentation.

Data augmentation is a method of creating new training data from existing training data. This is accomplished by applying domain-specific techniques to examples from the training data, resulting in new and distinct training examples.

The most well-known type of data augmentation is image data augmentation, which entails creating transformed versions of images in the training dataset that belong to the same class as the original image. Transforms encompass a wide range of image manipulation operations such as shifts, flips, zooms, and much more.

The goal is to add new, plausible examples to the training dataset. This refers to variations of the training set image that the model is likely to see. A horizontal flip of a cat photo, for example, may make sense because the photo could have been taken from the left or right. A vertical flip of a cat photo makes no sense and is probably inappropriate given that the model is very unlikely to see an upside-down cat photo.

As a result, it is obvious that the specific data augmentation techniques used for a training dataset must be selected with care and within the context of the training dataset and knowledge of the problem.

Modern deep learning algorithms, such as CNN, can learn features that are independent of their location in the image. Nonetheless, augmentation can help with this transform invariant approach to learning by assisting the model in learning features that are also transform invariant, such as left-to-right to top-to-bottom ordering, light levels in photographs, and more.

Generally, image data augmentation is only applied for the training dataset, not the validation or test dataset. This is distinct from data preparation tasks like image resizing and pixel scaling, which must be carried out consistently across all datasets that interact with the model.

Training and testing Using CNN and Vgg16

CNNs are a type of Neural Network that has proven effective in picture identification. CNNs are a type of multi-layer feed-forward neural network. CNNs are made up of filters, kernels, or neurons with learnable weights, specifications, and biases. Each filter takes some inputs, performs convolution, and then optionally adds non-linearity. Figure 4 depicts an example of a CNN architecture. CNN has Convolutional, Pooling, Rectified Linear Unit (RLU), and Fully Connected layers.

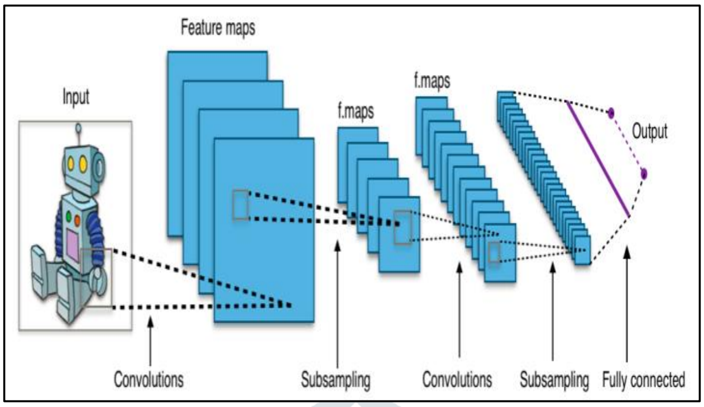


Fig 3.2 Architecture of CNN

- Convolutional Layer

The convolutional layer is the fundamental building block of a Convolutional Network, performing the majority of the computational heavy lifting. The Convolution layer's primary function is to extract features from the input data, which is an image. By learning image features from small squares of the input image, convolution preserves the spatial relationship between pixels. A set of learnable neurons is used to convolute the input image. This generates a feature map or activation map in the output image, which is then fed as input data to the next Convolutional layer. It is denoted mathematically as

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] (3.1)$$

where f represents the input image and h represents our kernel The indexes of the result matrix's rows and columns are denoted by m and n, respectively.

RLU Layer

Gated Recurrent Units (RLUs) are non-linear activation functions that work on multi-layer deep learning. In this layer, we replace all negative values in the filtered image with zeros. This function is only enabled when the node input exceeds a certain threshold. As a result, when the input is less than zero, the output is zero. When the input exceeds a certain threshold, however, it has a linear relationship with the dependent variable. This means it can ramp up the speed of a training data set in a deep neural network faster than other activation functions – this is made to prevent summing to zero.

$f(x) = max(0, x)$ (3.1)

Pooling Layer

The pooling layer minimises the dimensionality of every activation map while retaining the most critical information. The input images are separated into non-overlapping rectangles. A non-linear operation, of that kind as average or maximum, is used to down sample each region. This layer, which is typically placed between Convolutional layers, achieves better generalisation, faster convergence, and is resistant to translation and distortion.

The max-pooling layers are very simple and do not learn anything. They simply take a kkkk region and output a single value that is the maximum value in that region. For example, if their input layer is a NNNN layer, they will output a NkNkNkNk layer because the max function reduces each kkkk block to a single value.

- Flatten Layer

We should now have a pooled feature map after completing the previous two steps. Now we're going to flatten our pooled feature map into a column.

- Fully Connected Layer

The primary aim of using FCL is to use these features to classify the input image into different classes depending on the training dataset. The final pooling layer is regarded as FCL, which feeds the features to a classifier that employs the SoftMax activation function. The sum of the Fully Connected Layer's output probabilities is 1. This is accomplished by employing the SoftMax as the activation function. The SoftMax function takes an arbitrary vector of real-valued scores and squashes it to a vector of values between zero and one that sum to one.

- Vgg16

The architecture of the vgg16 model is depicted in Fig.3.3, which includes 13 convolutional layers, 2 fully connected layers, and 1 SoftMax classifier. Karen Simonyan and Andrew Zisserman introduced the VGG-16 architecture in 2014. Karen and Andrew built a

16-layer network with convolutional and fully connected layers. For simplicity, only 33 convolutional layers were layered on top of each other.

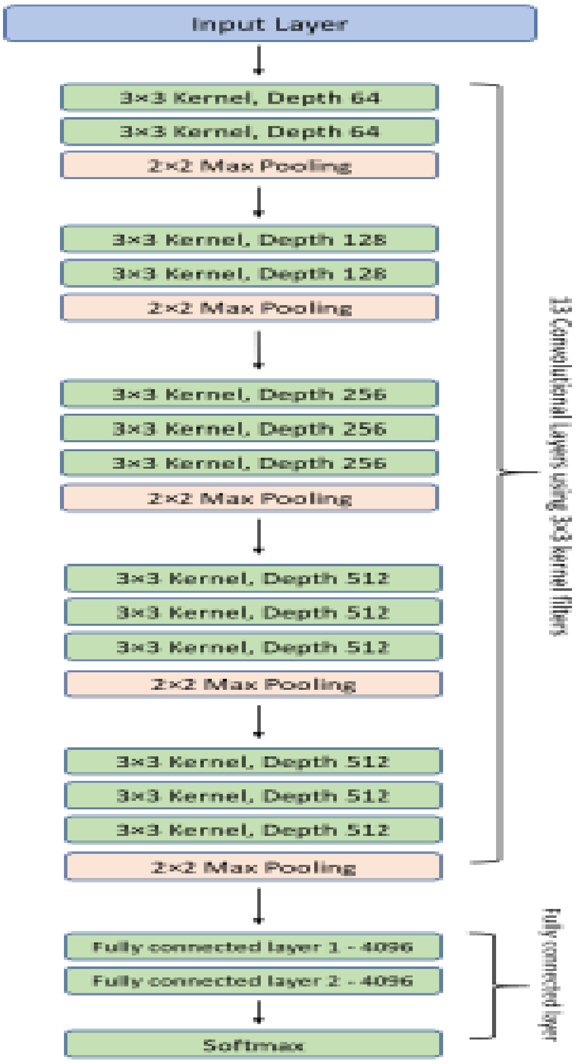


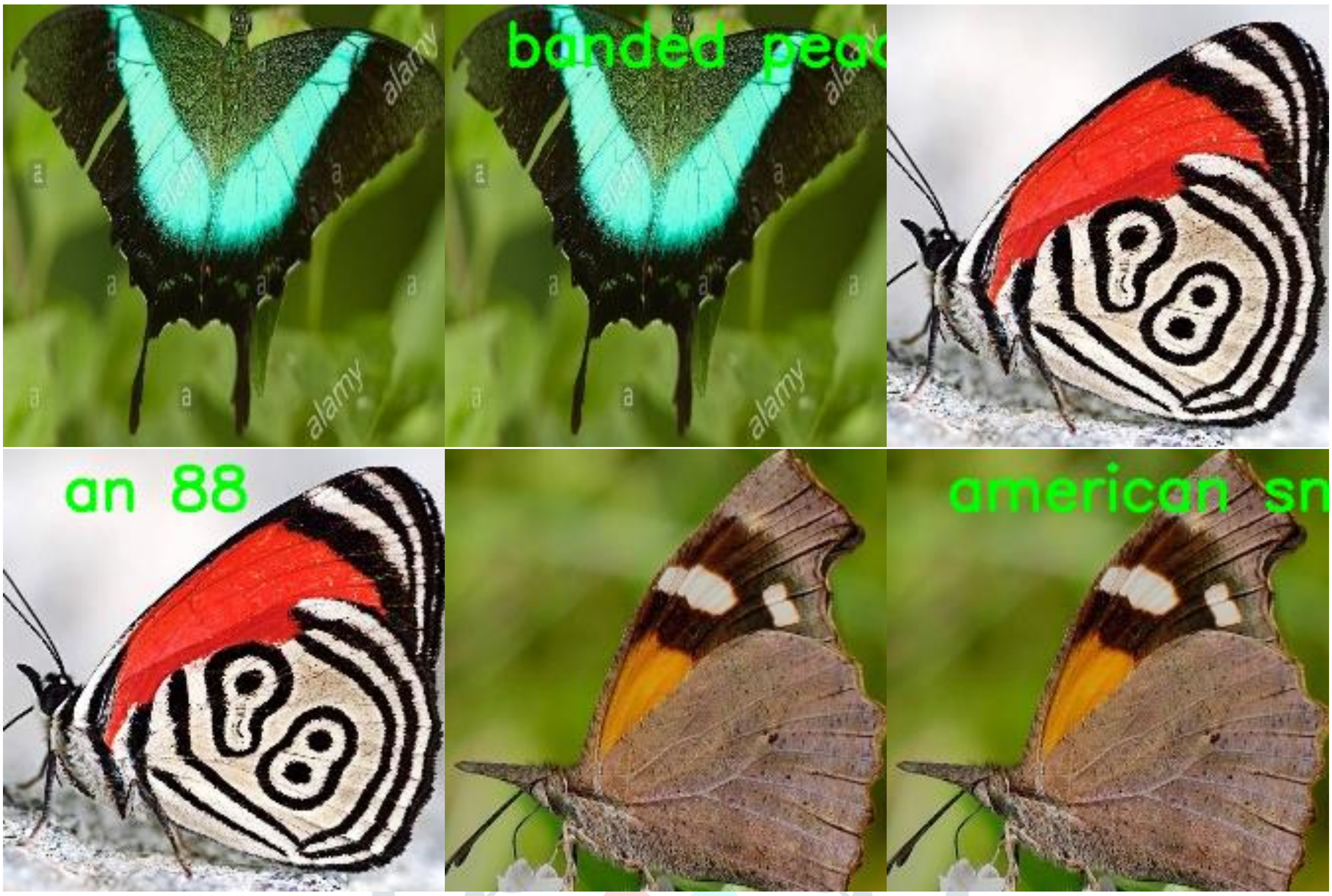
Fig. 3.3 VGG-16 model architecture

The following is the precise structure of the VGG-16 networks shown in Fig.3.3:

- The first and second convolutional layers are made up of 64 feature kernel filters with a filter size of 33. The dimensions of the input image (RGB image with depth 3) change to 224x224x64 as it passes through the first and second convolutional layers. The outcome is afterward passed to the max pooling layer with a stride of 2.
- The third and fourth convolutional layers are made up of 124 feature kernel filters with a filter size of 33. Following these two layers is a max pooling layer with stride 2, and the resulting output is 56x56x128.
- The fifth, sixth, and seventh layers are convolutional layers with kernel sizes of 33 and 33, respectively. All three make use of the 256 features. Following these layers is a max pooling layer with stride 2.
- Layers eight through thirteen are two sets of convolutional layers with kernel size 3x3. All of these convolutional layer sets have 512 kernel filters. Following these layers is a max pooling layer with a stride of 1.

- The fourteenth and fifteenth layers are fully connected hidden layers with 4096 units each, followed by a SoftMax output layer (sixteenth layer) with 1000 units

Result and Discussion:




indra swallo
grey hairstre
gold banded
copper tail

CONCLUSION:

In this system we proposed a framework design and implementation for butterfly detection and identification system using convolutional neural network approach. This semester, we collected the dataset from online source. First, we segment out the region of Interest (ROI) i.e., Butterfly from the image. In next phase we will train the model over the dataset for 50 species of the butterflies and evaluate the performance of the system using accuracy and loss. In this system, Vgg19 algorithms shows the highest accuracy among all classifier.

REFERENCES

[1] Ayad Saad Almryad, Hakan Kutucu, Automatic identification for field butterflies by convolutional neural networks, International Journal of Engineering Science and Technology, Volume 23, Issue 1, 2020, pp. 189-195.

[2] Ertugrul, Omer & Kaya, Yilmaz & Kayci, Lokman & Tekin, Ramazan. (2015). A Vision System for Classifying Butterfly Species by using Law's Texture Energy Measures. International Journal of Biomedical Data Mining. 1. 16-24.

[3] Ahmad, Zaaba & Hatim, Shahirah. (2021). Butterfly Family Detection and Identification Using Convolutional Neural Network for Lepidopterology. International Journal of Recent Technology and Engineering. 8. 636. 10.35940/ijrte.B1099.0982S1119.

[4] Keanu Buschbacher, Dirk Ahrens, Marianne Espeland, Volker Steinhage, "Image-based species identification of wild bees using convolutional neural networks", Ecological Informatics, Volume 55, 2020.

[5] Arzar, Nur & Sabri, Baity & Mohd Johari, Nur Farahin & Amilah, Anis & Noordin, Mohd & Ibrahim, Shafaf. (2019). Butterfly Species Identification Using Convolutional Neural Network (CNN). 221-224. 10.1109/I2CACIS.2019.8825031.

[6] A. Tan, G. Zhou and M. He, "Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion," in IEEE Access, vol. 8, pp. 124722-124733, 2020, doi: 10.1109/ACCESS.2020.3007745.

[7] Zhao, Ruoyan & Li, Cuixia & Ye, Shuai & Fang, Xinru. (2019). Butterfly Recognition Based on Faster R-CNN. Journal of Physics: Conference Series. 1176. 032048. 10.1088/1742-6596/1176/3/032048.

[8] Ayad Saad Almryad, Hakan Kutucu, "Automatic identification for field butterflies by convolutional neural networks", Engineering Science and Technology, an International Journal, Volume 23, Issue 1, 2020, Pages 189-195.

[9] Arroyo, Jan Carlo. (2021). Coleoptera Classification Using Convolutional Neural Network and Transfer Learning. International Journal of Engineering Trends and Technology. 69. 1-5. 10.14445/22315381/IJETT-V69I5P201.

[10] Xie, Juanying & Lu, Yinyuan & Wu, Zhaozhong & Xu, Shengquan & Grant, Phil. (2021). Investigations of butterfly species identification from images in natural environments. International Journal of Machine Learning and Cybernetics. 12. 10.1007/s13042-021-01322-8.