GRASSHOPPER ANALYSIS USING CNN ALGORITHM

BY

PRIYADHARSHINI.R

(19PCS012)

Project Report Submitted

In partial fulfillment of the requirements for the Award of

Master's Degree in Computer Science

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN (Deemed to be University)

COIMBATORE – 641043

MAY- 2021

GRASSHOPPER ANALYSIS USING CNN ALGORITHM

BY

PRIYADHARSHINI.R

(19PCS012)

Project Report Submitted

In partial fulfillment of the requirements for the Award of

Master's Degree in Computer Science

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN (Deemed to be University)

COIMBATORE – 641043

Signature of the

Head of the Department **Signature ofSupervisor**

Viva-voce Examination Held on ________________

Signature of Examiners

ACKNOWLEDGEMENT

I would like to express my sincere thanks to **God Almighty**, for his constant love and grace that he has showed upon me, which kept me in good health, and sound mind without which my project would not have reached a successful end.

I would like to express my deep sense of reverential gratitude and sincere thanks to **Prof.S.P.Thyagarajan, D.Sc,PhD,M.D,Chancellor**, Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatore, for the opportunity given to me for undertaking this study and for providing all the needed facilities during the course of my study.

I owe my great deal of gratitude to **Dr. Premavathy Vijayan, M.Sc., M.Ed., Dip.Spl.Edn., M.Phil., Ph.D., Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my gratitude to **Dr. S. Kowsalya, Registrar, M.Sc., M.Phil., Ph.D** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I would express my boundless thanks to **Dr. K. Udaya Chandrika, M.Sc., M.Phil., Ph.D., Dean, School of Physical Sciences & Computational Sciences,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr. Vasantha Kalyani David, M.Sc., M.Phil(Maths)., M.Phil(CS)., Ph.D, Professor and Head, Department of Computer Science** for support and encouragement to complete the project.

I express my heart full gratitude to my esteemed mentor **Dr. D.Mathivadhani, M.C.A., M.Phil., Ph.D., Senior Technical Assistant,** Department of Computer Science, for imparting the tremendous assistance and well-timed support for triumph of our project with guidance and constant supervision as well as for providing necessary resources for the project and also for her support in completing the project.

I am grateful to the project coordinator **Dr. G. Sudhamathy, M.C.A., Ph.D., Assistant Professor,** Department of Computer Science, who was instrumental in granting me the facilities required for doing project.

Finally, yet importantly, I would like to thank my **parents, family members and friends** for their kind inspiration, support, encouragement, blessings and prayers, which were instrumental in the successful completion of the project.

I have great pleasure in expressing my deep sense of gratitude to all other teaching and non-teaching staff members of the Department of Computer Science, who stood behind the screen for the completion of the project.

I would extend my hearty thanks to one and all that helped me directly or indirectly for the successful completion of my project.

ABSTRACT

Classification of insect species is a particularly difficult challenge because of the high degree of similarity of the appearance between distinct species. Growing interest in conservation and biodiversity increased the demand for accurate and consistent identification of biological objects, such as insects, at the level of individual or species. In the present study, we propose a novel approach based on a convolutional neural network (CNNs) to identify grasshopper species. Views are adopted on image capture, feature extraction, classification methods and the tested datasets. A new automatic identification system using photographic images has been designed to recognize different species of grasshoppers. The automatic classification system integrates multiple image processing tools to extract the geometry, morphology, and texture of the images. This challenge has been tackled by computer scientists as well as by biologists themselves. In this study, CNN were developed to discriminate different species of grasshopper based on differences in their wing structure.

CONTENTS

INTRODUCTION

In today world they are a million types of insects present. They are different colours, shapes, sex and life stage which makes the task of insect classification and recognition challenging. Insect identification is difficult as it requires a detail understanding of insect taxonomy as well as the morphological characteristics terms. Only experts such as taxonomists and trained technicians can identify taxa accurately. However, the number of taxonomists and other identification experts has drastically decreased. Computer vision is a technology with the potential to make complete automation of the task possible. This is because it can fully utilize the huge potential offered by information technology, instead of relying on users to compare specimens to images or illustration (Yaakob and Jain, 2012[1]). Building accurate knowledge of the identity, the geographic distribution and the evolution of living species is essential for a sustainable development of humanity as well as for biodiversity conservation (Martineau *et al.,* 2016[2]). The identification process is also necessary for monitoring the spread of pollution and disease vectors and to identify areas of biodiversity (Gaston and O'Neill, 2004; Rua *et al.,* 2009[3]).

Orthoptera is one of the largest orders of insect having two suborder Cealifera (short horned grasshoppers and Ensifera (Long horned grasshoppers). Grasshoppers are one of the most important components of the terrestrial ecosystem. They are providing a wide range of environmental and ecological services by supporting other predators and parasites in the natural food chain. Some grasshoppers are distinctive in nature and cause serious damage to various wild and agricultural crops (Joshi *et al.,* 1999). They are also important components of the food chain for many birds and mammals (Capinera *et al.,* 1997[4]). Grasshoppers are difficult to identify because the features that distinguish them (e.g., shape, wing venation, wing colour etc.) are complicated and subtle.

The development of digital image processing technology and pattern recognition techniques, improved methods have been suggested for automatically identifying species based on the feature extraction of image shapes (Arbuckle *et al.,* 2001; Larios *et al.,* 2007[5]). In recent years, deep learning (DL) and convolutional neural networks (CNNs) have emerged as the most effective approaches to a range of problems in automated classification (LeCun *et al.,* 2015; Schmidhuber, 2015[6]) and computer vision is one of the fields where these techniques have had a transformative impact. CNNs have been particularly successful in image classification tasks, where large labelled training sets are available for supervised learning. Sophisticated CNNs and transfer learning have been used successfully in recent years to improve the classification of some biological image data sets

These methods are helpful in alleviating the “taxonomy crisis” In this research, we present a new methodology for the identification of different taxonomic groups to the species level for grasshoppers. We designed a simple and effective algorithm (pre-process solution) and defined a range of new features that use pattern recognition with convolutional neural networks (CNNs).

1.1Problem defination

Almost whatever the approach employed, there are some common significant issues associated with reliable automated species identification. The quality of training sets are highlighted, the nature of errors in identification, how to scale the process up to differentiate among larger numbers of species and how to deal with species that a system has not been trained to identify.

SYSTEM SPECIFICATION

2.1 Hardware Specification

Processor:INTEL(R) Celeron(R)N4020 CPU@1.10GHz

RAM-4.00 GB

System type:64-bit operating system, x64-based processor

2.2Software specification

Operating system: Windows 10

Software : Python 3.7 – Anaconda3-5.3.1- Windows-X86_64

2.3 ABOUT THE SOFTWARE

2.3.1 PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018. Python can be used on a server to create web applications.

Python features a dynamic type system and automatic memory management. It supports multiprogramming paradigms , including object oriented, imperative, functional and procedural, it also has a comprehensive standard library. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (Including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter(), map(), and reduce() functions; list comprehensions, dictionaries, sets and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document *The Zen of Python* (*PEP 20*), which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python features include:

- Easier access to debuggers through a new breakpoint() built-in
- Simple class creation using data classes
- Customized access to module attributes
- Improved support for type hinting
- Higher precision timing functions
- Readable and Maintainable Code
- Multiple Programming Paradigms
- Compatible with Major Platforms and Systems
- Robust Standard Library
- Many Open Source Frameworks and Tools
- Simplify Complex Software Development
- Adopt Test Driven Development

2.3.2 Python packages

cv2

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. **cv2**. imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

TensorFlow

TensorFlow is a python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of **TensorFlow**.

Matplotlib.Pyplot

Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots and so on. Matplotlib is a graphics package for data visualization in Python. It is well integrated with NumPy and Pandas. The pyplot module mirrors the MATLAB plotting commands closely. Hence, MATLAB users can easily transit to plotting with Python.

SYSTEM DEVELOPMENT

Dataset description

Table 1. Details of species used in this study

S.No	Species	Family	Subfamily	Tribe
1.	*Leva indica*	Acrididae	Gomphocerinae	Arcypterini
2.	*Acrotylus patruelis*	Acrididae	Oedipodinae	Acrotylini
3.	*Trilophidia annulata*	Acrididae	Oedipodinae	Trilophidiini
4.	*Spathosternum prasiniferum*	Acrididae	Spathosterninae	Spathosternini

Acridids such as *Leva indica*, *Acrotylus patruelis, Trilophidia annulate* and *Spathosternum prasiniferum* were dominant species and found in all the zones and seasons of Coimbatore.

(i) Training sets

The quality of training sets is important in obtaining reliable identifications. Such sets need to comprise high quality images of specimens that have been accurately identified, to be reasonably large, and, while avoiding particularly aberrant specimens, to capture sufficient of the breadth of morphological variation exhibited by individuals of each species. Establishing larger training sets can be difficult where some of the species to be included are rare, and specimens may be problematic to obtain; this problem is exacerbated in situations where any marked variation in the size of training sets for different species may reduce the accuracy of identifications (Boddy *et al.,*2001; Morris *et al.,* 2001[7]).

(ii) Errors in identification

Different approaches to automated species identification may tend to lead to different error patterns. For example, different pre-processing methodologies (histogram stretch, wavelet transforms, edge extraction etc.) lead to different error patterns in the DAISY identifier (O'Neill, 2000[8]).Empirically these error patterns have been found to be essentially orthogonal.

(iii) Scaling

If sufficiently reliable automated identifications can be made among a few species, the issue remains as to how to scale the process up to differentiate among large numbers. This is a non-trivial problem. First, in general, increasing the number of species amounts to increasing the breadth of taxa under consideration. Different taxa often require different sets of features to be examined to generate species-level identifications, which requires different kinds of species identifier .

(iv) Novel species

The development and testing of systems for automated species identification has almost exclusively been concerned with closed assemblages, in which a training set is established for a set of species, and the system is tested using other specimens of these same species. Identification of novel species is probably the most serious challenge to the development of automated species identification systems. The problem can therefore be seen differently along granularity in the taxonomy .

Methodology

Collection of grasshoppers

Grasshoppers were collected randomly at different areas of Coimbatore (11.0603° N, 76.9458° E) Tamil Nadu, India . The specimens were caught using the insect sweep net and the handpicking methods.

Identification of grasshoppers

Collected Grasshoppers were identified based on morphological characteristics using the manual, Identified specimens were sorted out species-wise and counted.The grasshopper is identified by plot into graph.

Wing preparation

In this study, forewings were used to study the identification features of grasshoppers. The wings were removed from the body of adult grasshoppers using dissecting needles, scalpel and forceps. The wings thus removed were placed on a glass slide and examined under a microscope to describe the venation pattern of species.

Image Processing

Each wing image was photographed with the same scale using a digital camera attached to phase-contrast stereomicroscope (consistent magnification 10X/22). The digital images of grasshopper wings were used in data analysis.

CNN Algorithm

A Convolutional Neural Network (ConvNet/cnn) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. CNNs are used for image classification and recognition because of its high accuracy. The cnn follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed One of the main parts of Neural Networks is Convolutional neural networks (cnn). They are made up of neurons with learnable weights and biases. Each specific neuron receives numerous inputs and then takes a weighted sum over them, where it passes it through an activation function and responds back with an output. Image processing using CNN, for detection the grasshopper is quite powerful for recognize the image. Researchers are trying to identify the insects using image processing and different algorithm where a few researchers got success in this type of work. Orthoptera is one of the largest order of insects commonly known as the grasshoppers, crickets, and katydids. .Grasshopper identification is hard because of their immense species diversity and the significant variation within and between species.

The CNN-based automated identification systems clearly develop a powerful high-level understanding of the tasks with which they are challenged. If that understanding of taxonomic identification tasks could be communicated effectively to biologists, it could replace traditional identification keys. Perhaps the best way to do this would be through visualizations of morphospace that are optimized for identification. In other words, morphospace as it is organized by a CNN trained on a relevant identification task. It is not difficult to imagine ways in which such approaches might generate identification tools that would be highly efficient and very appealing to humans.

Convolutional layer

This is the first layer of CNN used to extract the various features from the input images.The dataset extract the equal and same size for all images.

Pooling layer

The pooling layer used to predefined the size of all images.This calculate the average size of the image.This is also used to calculate the predefined size in the image selection.

Activation layer

The important in CNN is the activation function is used to decides which in formation of the model should be in forward and which is not.

Dense layer

To complete the model will feed the last output tensor from the convolutional base into one or more dense layer to perform classification.Dense layer take vector as input while the current output is a 3D tensor

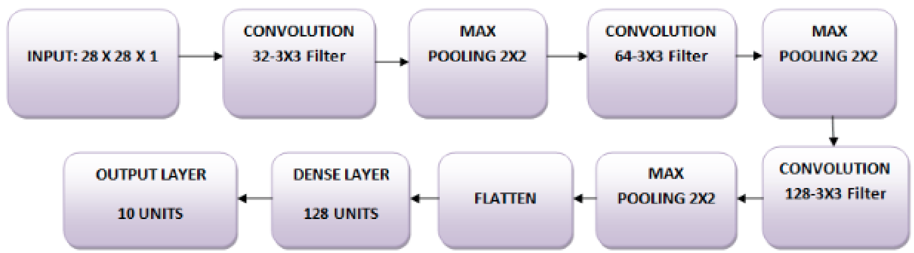
Flattern layer

The flattern layer the 3D output to 1D then add one or more dense layer for output.

```
Model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(128, (3, 3), padding="same",
activation='relu', input_shape=(169, 785, 3)),
                    tf.keras.layers.Conv2D(64, kernel_size=(4, 4), padding="same",
activation='relu'),
                    #
                    tf.keras.layers.Conv2D(32, kernel_size=(4, 4), padding="same",
activation='relu'),
                    tf.keras.layers.MaxPool2D(2,2),
                    #
                    tf.keras.layers.Conv2D(32, kernel_size=(4, 4), padding="same",
activation='relu'),
                    tf.keras.layers.MaxPool2D(2,2),
                    ##
                    tf.keras.layers.Flatten(),
                    ##
                    tf.keras.layers.Dense(512, activation='relu'),
                    ##
                    tf.keras.layers.Dense(1, activation='softmax')
                    ])
```

the 6 lines of code below define the convolutional base using a common pattern a stack of Conv2D and MaxPooling2D layers.

As input, a CNN takes tensors of shape (image_height, image_width, color_channels), ignoring the batch size.The dimensions, color_channels refers to (R,G,B). In this example, you will configure our CNN to process inputs of shape (32, 32, 3).



model.summary()

Model: "sequential_1" ____

Layer (type) Output Shape Param #

==

==== conv2d_4 (Conv2D) (None, 169, 785, 128) 3584 ____ conv2d_5 (Conv2D) (None, 169, 785, 64) 131136 ____ conv2d_6 (Conv2D) (None, 169, 785, 32) 32800 ____ max_pooling2d_2 (MaxPooling2 (None, 84, 392, 32) 0 ____ conv2d_7 (Conv2D) (None, 84, 392, 32) 16416 ____ max_pooling2d_3 (MaxPooling2 (None, 42, 196, 32) 0 ____ flatten_1 (Flatten) (None, 263424) 0 ____ dense_2 (Dense) (None, 512) 134873600 ____ dense_3 (Dense) (None, 1) 513

==

==== Total params: 135,058,049 Trainable params: 135,058,049 Non-trainable params: 0
____

The output of every Conv2D and MaxPooling2D layer is a 3D tensor of shape (height, width, channels). The width and height dimensions tend to shrink as you go deeper in the network. The number of output channels for each Conv2D layer is controlled by the first argument (e.g., 32 or 64[9]). Typically, as the width and height shrink, you can afford (computationally) to add more output channels in each Conv2D layer.

CONCLUSION

Species identification of grasshopper is a fundamental part of recognizing and describing biodiversity. Rapid and reliable identification of insects, either to species or to higher taxonomic groups, is important in many contexts. Convolutional Neural Network (CNN) will be used to do the image recognition and the algorithm will provide an optimal architecture for image recognition. This approach also will help a taxonomist to recognize the insect from ecosystems.There is no doubt that CNN algorithm for alleviating the taxonomy problems. Understanding of taxonomic identification tasks could be communicated effectively to biologists, it could replace traditional identification keys. Building accurate knowledge of the identity is essential for a sustainable development of humanity as well as for biodiversity conservation. The identification process is also necessary for monitoring the spread of pollution and disease vectors and to identify areas of biodiversity.

SCOPE FOR FUTURE ENHANCEMENT

Implementation of AI technique will be used to discriminate different species of grasshopper based on differences in their wing structure. Effective feature extraction from AI tools provides a convenient and efficient method for developing fully automated taxonomic identification systems. Application of AI to routine identification clearly develop a powerful high-level understanding and start thinking about completely new ways in which algorithm could help boost systematic research.

BIBLIOGRAPHY

1. Cigliano, M.M.,Braun, H. and Eades, D.C. and Otte D(2017) Orthoptera species file online. Version (5.0)http://orthoptera.speciesfile.org
2. Donahue J., Hendricks L.A., Guadarrama S,. Rohrbach M., Venugopalan S., Saenko K., Darrell T. (2015). Long-term recurrent convolutional networks for visual recognition and description. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE. p. 2625–2634.
3. Kasinathan, T., Singaraju, D. and Uyyala, S.R. (2020). Insect classification and detection in field crops using modern machine learning techniques, Information Processing in Agriculture, https://doi.org/10.1016/j.inpa.2020.09.006
4. LeCun Y., Bengio Y., Hinton G. (2015). Deep learning. Nature 521:436–444.
5. O'Neill MA (2000) DAISY: a practical tool for semi-automated species identification. In: MacLeod N (ed) Automated taxon identification in systematics: theory, approaches, and applications. CRC Press/Taylor & Francis Group, Boca Raton/Florida, pp 101–114.
6. Ponttuset,J.;Arbelaez, P.; Barron, J.; Marques, F.; Malik, J. (2016).Multiscale combinatorial grouping for image segmentation and object proposal generation. IEEE Trans. Pattern Anal. Mach. Intell.39, 128–140.

7. Shishodia, M. S., Chandra, K. and Gupta, S. K. (2010). An Annotated checklist of Orthoptera (Insecta) from India. Rec. Zool. Surv. India, Occ. Paper. 314 : 1-366.

8. Tandon, S. K and Hazra, A. K (1998). Faunal diversity in India Orthoptera, pp183-188. ENVIS Center. Zoological Survey of India Kolkata.

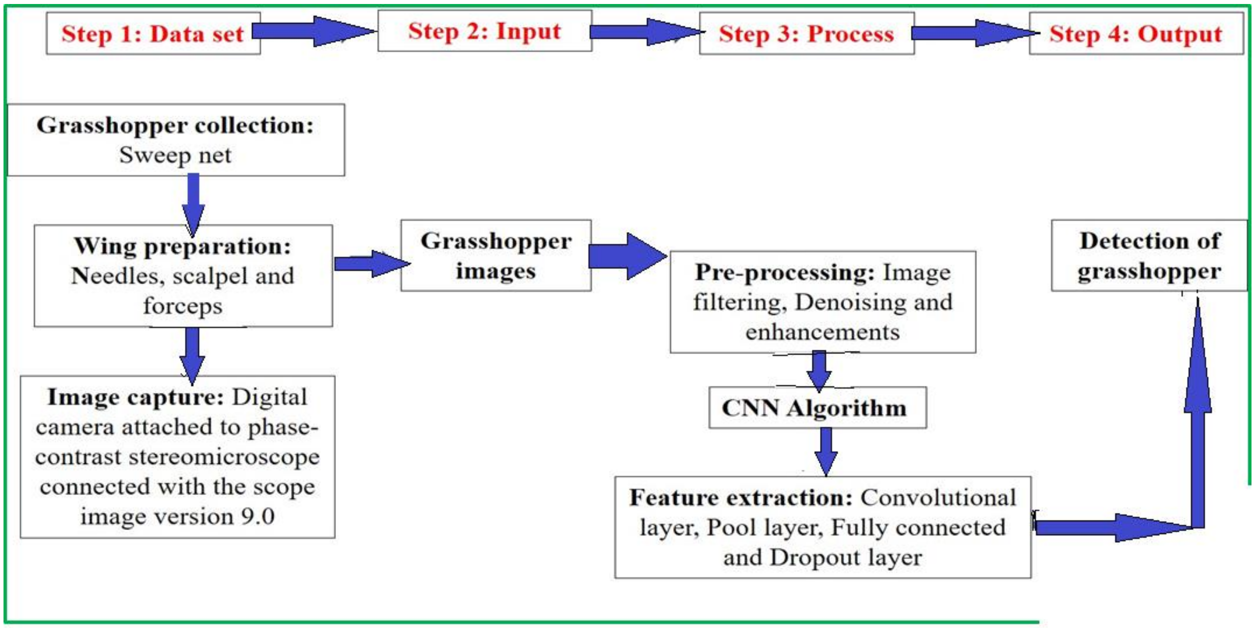
9. Theivaprakasham, H, (2020). Identification of Indian butterflies using Deep ConvolutionalNeural Network, *Journal of Asia-Pacific* Entomology.doi: https://doi.org/10.1016/j.aspen.2020.11.015

10. Valan, M., Makonyi, K., Mak A., Vondrac D. and Ronquist, F. (2019). Automated Taxonomic Identification of Insects with Expert-Level Accuracy Using Effective Feature Transfer from Convolutional Networks *Syst. Biol.* 68(6):876–895.

11. Xu K., Ba J., Kiros R., Cho K., Courville A., Salakhudinov R., Zemel R., Bengio Y. (2015). Show, attend and tell: neural image caption generation with visual attention. Lile, France: International Machine Learning Society (IMLS). p. 2048–2057.

APPENDIX

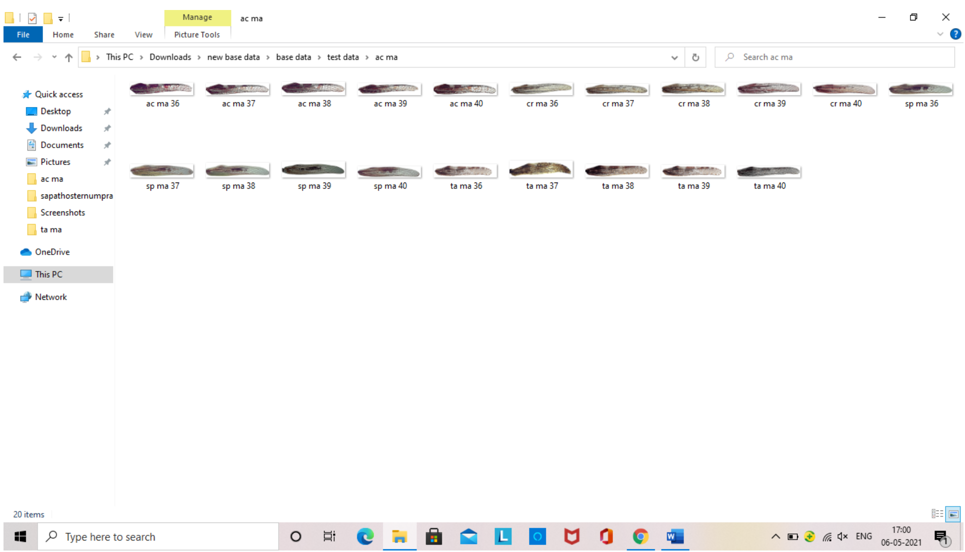
Data flow diagram


Step 1: Data set
Step 2: Input
Step 3: Process
Step 4: Output
Grasshopper collection: Sweep net
Wing preparation: Needles, scalpel and forceps
Image capture: Digital camera attached to phase-contrast stereomicroscope connected with the scope image version 9.0
Grasshopper images
Pre-processing: Image filtering, Denoising and enhancements
CNN Algorithm
Feature extraction: Convolutional layer, Pool layer, Fully connected and Dropout layer
Detection of grasshopper

Sample dataset:

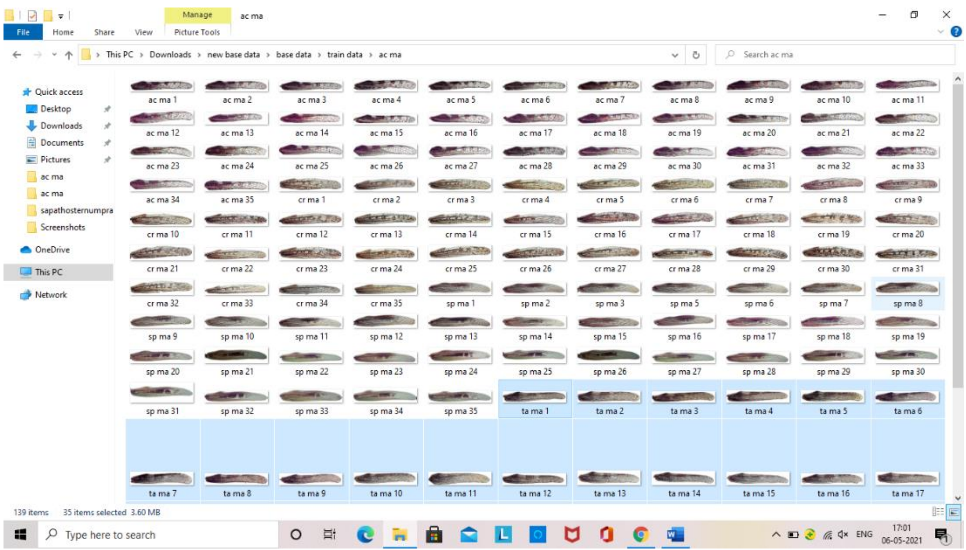
Test dataset

The test data set is consist of 20 images out of 200 images



Train dataset

The train dataset is consist of 139 images out of 200 images.



Valid dataset

The valid dataset consist of 40 images out of 200 images

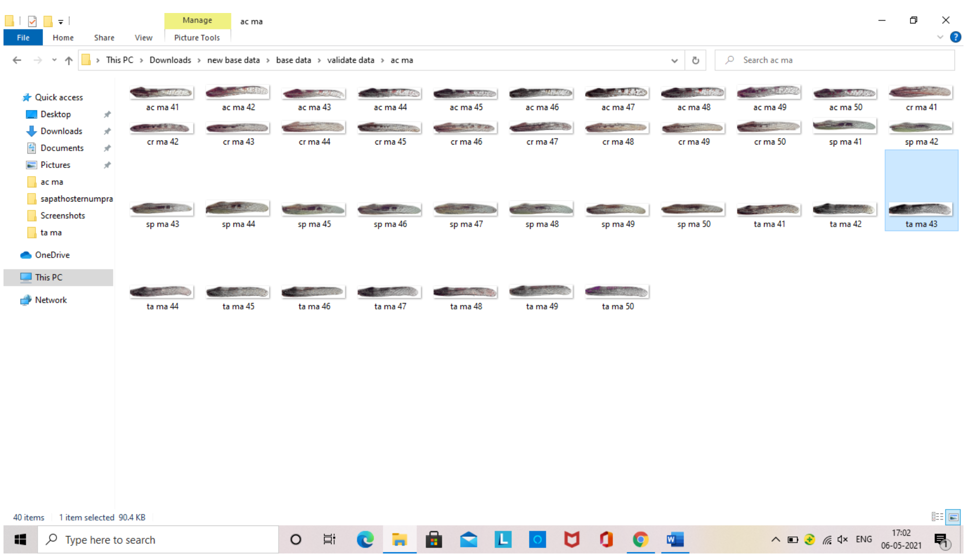
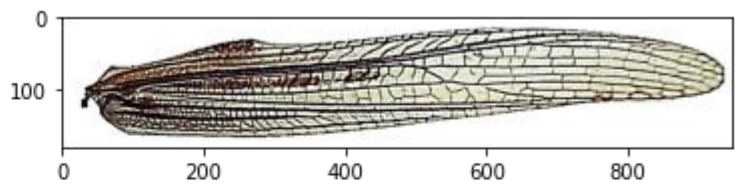
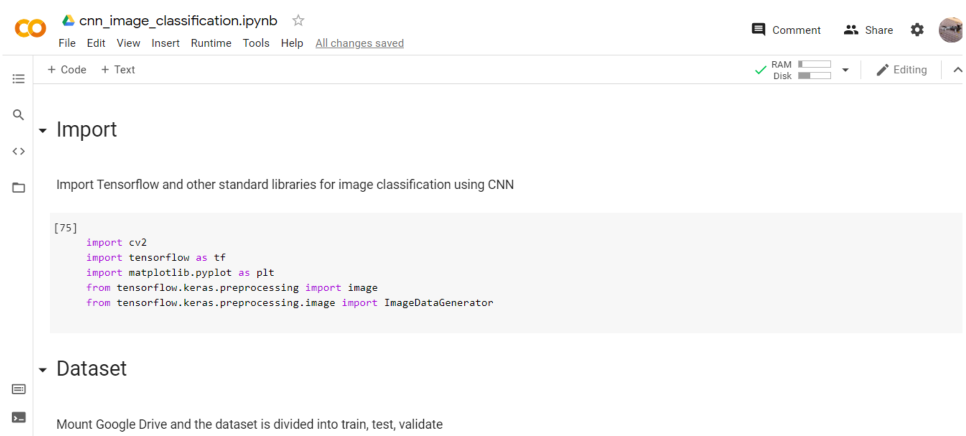
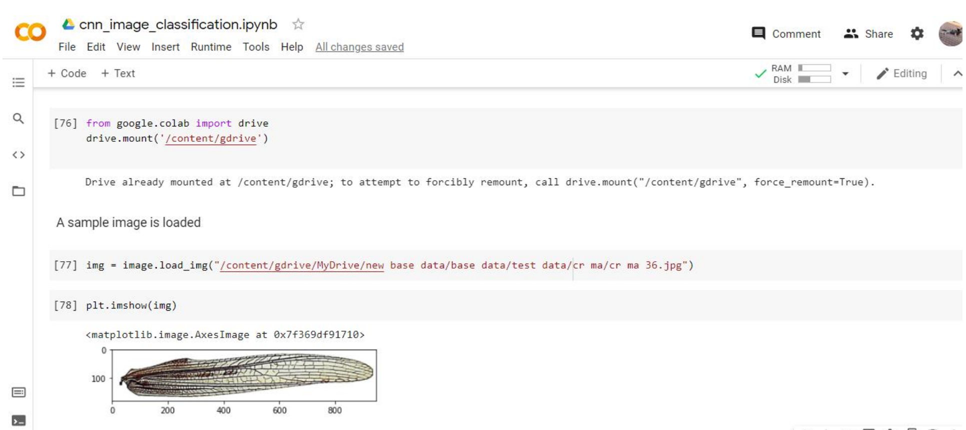


Image plotted in graph



SCREENSHOTS





cnn_image_classification.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
cv2.imread("/content/gdrive/MyDrive/basedata/train/cr ma/cr ma 2.jpg")
```

```
array([[[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
```

cnn_image_classification.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[80] cv2.imread("/content/gdrive/MyDrive/basedata/train/cr ma/cr ma 2.jpg").shape
```

```
(169, 785, 3)
```

ImageDataGenerator will lable the data based on the directory and also rescale the data so that all the pixel values are between 0 to 1

```
[81] train = ImageDataGenerator(rescale=1/255)
     validation = ImageDataGenerator(rescale=1/255)
```

Creating train dataset

```
[82] train_dataset = train.flow_from_directory('/content/gdrive/MyDrive/new base data/base data/train data',
                                              target_size=(169, 785),
                                              batch_size=5)
```

```
Found 139 images belonging to 4 classes.
```

Creating Validation dataset

cnn_image_classification.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[83] validation_dataset = train.flow_from_directory('/content/gdrive/MyDrive/new base data/base data/validate data/',
                                              target_size=(169, 785),
                                              batch_size=5)
```

```
Found 40 images belonging to 4 classes.
```

While importing data to neural network the labels should be numeric class_indices show the corresponding categorical label

```
[84] train_dataset.class_indices
```

```
{'ac ma': 0, 'cr ma': 1, 'sp ma': 2, 'ta ma': 3}
```

```
[85] train_dataset.classes
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3], dtype=int32)
```

cnn_image_classification.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Model

Training Model

```
[86] model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(128, (3, 3), padding="same", activation='relu', input_shape=(169, 785, 3)),
                                    tf.keras.layers.Conv2D(64, kernel_size=(4, 4), padding="same", activation='relu'),
                                    #
                                    tf.keras.layers.Conv2D(32, kernel_size=(4, 4), padding="same", activation='relu'),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    #
                                    tf.keras.layers.Conv2D(32, kernel_size=(4, 4), padding="same", activation='relu'),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    ##
                                    tf.keras.layers.Flatten(),
                                    ##
                                    tf.keras.layers.Dense(512, activation='relu'),
                                    ##
                                    tf.keras.layers.Dense(1, activation='softmax')
                                    ])
```

cnn_image_classification.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[88] model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 169, 785, 128)     3584
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 169, 785, 64)      131136
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 169, 785, 32)      32800
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 84, 392, 32)       0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 84, 392, 32)       16416
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 42, 196, 32)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 263424)            0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               134873600
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 513
=================================================================
Total params: 135,058,049
Trainable params: 135,058,049
```

RESULTS & DISCUSSION

Automated entomology has been around for decades. Machine learning-based image processing models have been developed for accurategrasshopper classification and identification.The result accurancy is 80%.