SCHOOL OF INFORMATION AND TECHNOLOGY

SOFT COMPUTING - SWE2019

FINAL REVIEW DOCUMENT

TOPIC : IMAGE CLASSIFICATION USING CNN

By : P R Lalith Sagar

18MIS0045

Faculty : Prof. Subhashini R

Slot : C1+TC1

Image Classification using Convolutional Neural Network

Abstract:

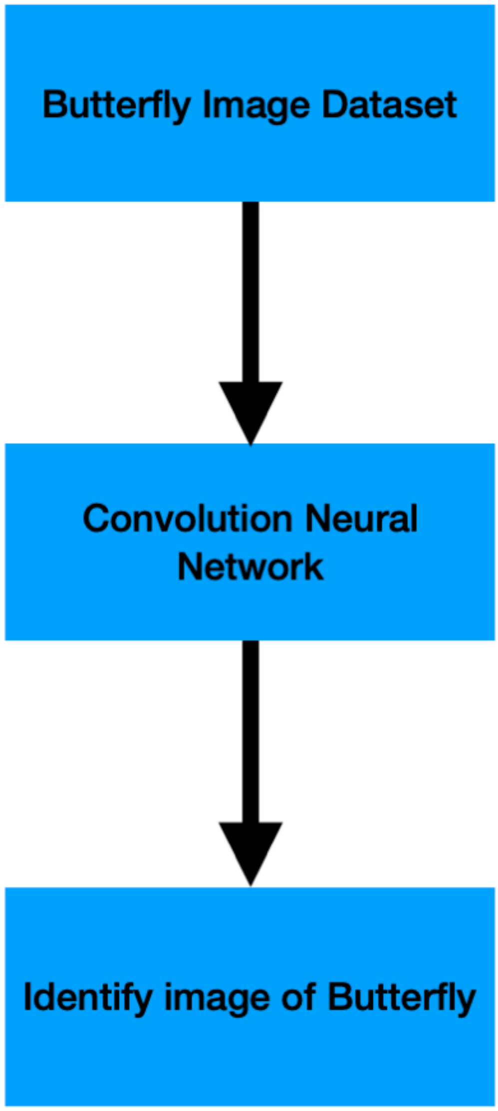
Butterflies are the most essential from of ecosystem, aesthetic, educational, economic, scientific and intrinsic value in our human world. One of the popular significance of butterflies is as a model to understand the effect of habitat loss and environmental change. Current approach in image processing for butterfly identification is not efficient due to complicated butterfly shapes. Gathering, recognizing and archiving specimen images physically is tedious and costly for entomologist. Hence, the need to have an application that can accelerate the process using a technique that easy to understand will definitely solve the problems.

Nowadays, dataset consist a lot of noise or too small to suit to latest application. Thus, a study of butterfly species identification using image processing technique and Convolution Neural Network (CNN) is proposed. This research mainly focuses on GoogLeNet which is a pre-trained model of Convolutional Neural Network (CNN) architecture. Four species of butterflies which that are commonly found in Asia which are 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.

Introduction:

The process of Image classification is used for labelling images according to predefined set of categories. The process of image classification is mainly based on Supervised learning. An image classification model is fed with a set of images within a specific category. Based on the set which is fed, the algorithm will learn which class or category the test images belong to and then it can predict the correct class of future image inputs, and this can also measure how accurate the predictions are.

Convolutional neural networks (CNN) are one of the special architectures of artificial neural networks which was proposed by Yann LeCun in the year 1988. The initial versions of Convolutional Neural networks are called as LeNet, used for many applications such as to recognize the handwritten digits, in banking applications and to read zip codes. Later the CNN found its place in image classification process. CNN help to analyse visual imagery in image classification process.



The Convolutional Neural Network based deep neural system is widely used in the medical filed. CNN also used in decoding facial recognition to identify each and every face in a picture and to identify unique features.CNN is widely used for document analysis. This is not just useful for handwriting analysis but also plays a major role in recognizers. It can also be used to fight against the climatic changes and to understand the various reasons for the sudden and drastic climatic changes. Some other applications include historic and environmental collections, grey areas and advertising.

Image of butterfly will go through resizing and cropping to 224x224 pixels. To classify image that has been resized, a pre-trained GoogLeNet model is used which produces the final outcome of the butterfly classification subsequently. After process of GoogLeNet completed, accuracy percentage will be calculated using confusion matrix. Lastly, identified image of butterfly and habitat information will be displayed.

Dataset specification:

- https://www.kaggle.com/veeralakrishna/butterfly-dataset
- This data set consist of butterfly images. This data set is used for training and testing process, eight hundred and thirty-two butterfly images are used to test the accuracy of whole system. The images of the butterflies will pass through the first architectural layer, which is a convolutional layer in which all features are extracted and the result is transferred to the next layer. After the completion of entire process, classification accuracy for all species of butterfly is calculated.

Literature Survey:

Paper 1:
Image Classification Using Convolutional Neural Network
Author(s):
Muthukrishnan Ramprasath, M.Vijay Anand, Shanmugasundaram Hariharan
Literature Review:
In this paper the authors used deep learning alorightm to achieve their expected results in the fields like computer vision.They also used a machine learning algorithm , Convolutional Neural Network to automate the process of classification of images.The defined system uses Digit of MNIST data set to classify grayscale images.The grayscale images used by them for training requires more computational ability to classify them.By training these images using Convolutional Neural Network,the authors were able to achieve 98% accuracy in the experiment they did.The method described by the authors in this paper has high accuracy in classifying images.The proposed system uses CNN for implementation.
Algorithm:
1.Batch size =128
2.No of classes 10, Number of epochs = 5
3.Dimensions of input image should be 28 x 28.
4.Images are retrieved from MNIST dataset.
5.Variables: X=test data set (10000,28,28,1), Train data set (60000,28,28,1)
6.The models should be created and compiled.
7.Train the network.

Paper 2:
Title:
Butterfly Species Identification Using Convolutional Neural Network (CNN)
Author(s):
Nur Nabila Kamaron Arzar, Nurbaity Sabri, Nur Farahin Mohd Johari, Anis Amilah Shari, Mohd Rahmat Mohd Noordin, Shafaf Ibrahim
Literature Review:
Authors of this paper made a study on butterfly species identification using image classification and proposed Convolutional Neural Networks method.The authors mainly focused on GoogLeNet ,which is a pretrained model of CNN architecture.The authors used four species as a dataset.The four species are namely Black Veined Tiger, Chocolate Grass Yellow, Grey Pansy and Plain Lacewing.They performed testing on 120 images of four species and the experiment resulted with 97.5 accuracy on identification.The main objective of this paper is to propose a method for identification of butterfly species and to evaluate the performance of classification methodology. For this study GoogLeNet, a pretrained model is used. GoogLeNet model is defined to use a number of smaller convolutional kernels to decrease the quantity of neurons and parameters.They used confusion matrix to calculate the accuaracy percentage. Confusion matrix used by authors has four components which are True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN). Accuracy percentage for all species of butterfly images is calculated as:
Accuracy % = (Total TRUE value / Total images testing) * 100
The authors succeeded their research as they obtained 97.5% over 120 images of butterfly. They came to a conclusion that the have proposed a efficient method for classification of butterfly species and also said that they are going to add more species in future.

Paper 3:
Title:
Research on Image Classification Model Based on Deep Convolutional Neural Network
Author(s):
Mingyuan Xin and Yong Wang
Literature Review:
Authors made an analysis on error Back Propagation algorithm and proposed a new training algorithm of depth neural network maximum interval minimum classification error.In addition, they analyzed cross entropy and M3CE and combined them to achieve better results. They used MNIST and CIFAR-10 as standard databases.

Finally, they tested the proposed M3 CE-CEc on these deep learning databases. The study showed that M3 CE can improve the cross-entropy, and it is very effective to the cross-entropy . M3 CE-CEc has achieved better results in both these standard databases.

The proposed method has mainly three parts image preprocessing, image feature extraction and classifier.In this paper for the defined data set the random generation method(ZCA process) is used to avoid the color of artificial classification.Feature extraction is done by time-frequency composite weighting.The authors also wrote a note on application of deep convolution neural network in image classification. In this paper, the evaluation index calculated based on image recognition effect,which is divided into three parts: the overall classification accuracy,classification accuracy of different categories, and classification time consumption.The authors were able to obtain 80% accuracy rate ,which seemed to be same among different species.They also stated that accuracy percentage increases as clarity of the image increases.

Paper 4:

Title:

Automatically Designing CNN Architectures Using Genetic Algorithm For Image Classification

Author(s):

Yanan Sun, Mengjie Zhang, Gary G. Yen and Jiancheng Lv

Literature Review:

In this paper the authors proposed automaticCNN architecture design method by using genetic algorithms for performing classification of images.The main advantage of this method is that the intended users do not need the knowledge of CNN architecture. The proposed algorithm is tested on widely used image classification datasets, by comparing to the state-of-the-art peer competitors covering eight manually-designed CNNs, seven automatic+manually tuning and five automatic CNN architecture design algorithms. The study results shows that the proposed algorithm outperforms the existing automatic Convolutional Neural Network architecture design algorithms in terms of classification accuracy, parameter numbers and computational resources. The proposed algorithm presents the comparable classification accuracy to the best one from manually-designed and automatic+manually tuning CNNs,by taking less computational resources.

The authors used six algorithms in the proposed methodology. To evaluate the performance of the proposed algorithm authors performed a series of experiments on image classification tasks. The main aim of this paper is to propose an automatic architecture design algorithm for CNNs by using the genetic algorithm which is capable of discovering the best Convolutional Neural Network architecture in solving image classification problems for the users who have no knowledge in tuning CNN architectures. The authors has been successfully achieved the goal by designing a new encoding strategy for the genetic algorithm to encode arbitrary depths of CNNs, incorporating the skip connections to promote deeper CNNs to be produced during the evolution and developing a parallel as well as a cache component to significantly accelerate the fitness evaluation given a limited computational resource.

Paper 5:

Image Classification Based on the Kohonen Network and the Data Space Modification

Author(s):

Volodymyr Gorokhovatskyi, Iryna Tvoroshenko

Literature Review:

In this paper, the authors have proposed a solution for visual object recognition using a methodology which involve classification of image key points based on the training phase of Kohonen neural network. The Kohonen network can be able to recognize clusters in data and also can establish the proximity of classes in this case. Therefore, it is possible to enhance the understanding and interpretation of data structures to refine the network model by changing the existing rules for the process of identifying objects. In turn this paper also describes the results drawn from a comparative study on developed training methods and classification of images for given butterfly dataset for numerous parameters involved in the kohonen classification network. The methods which involve the formation of a system of centres and also uses data convolutional turned up some best results for the taken samples and the authors interpreted that the use of convolution increases the speed and accuracy of data processing compared to all other methods and options. The main contribution of the research carried out in this paper is the enhancement of classification methods using Kohonen network by using a newly descripted data space based on data convolution which helps to achieve high data classification accuracy and efficiency and makes this possible to use in real time application. The other practical importance of work carried in this paper is achieved by obtaining software models for the purpose of assessing the effectiveness of classifiers in the computer vision system.

Paper 6:

Title:

Going Deeper with Contextual CNN for Hyperspectral Image Classification

Author(s):

Hyungtae Lee, Heesung Kwon

Literature Review:

In this paper they describe a novel deep convolutional neural network that is deeper than other existing deep networks for hyperspectral image classification. Unlike current state-of-the-art approaches in CNN-based hyperspectral image classification the proposed network, called contextual deep CNN. The proposed approach is tested for three benchmark datasets: the Indian Pines dataset, the Salinas dataset and the University of Pavia dataset. Performance comparison shows an enhanced classification performance of the approach over the current state-of-the-art on the three datasets. They randomly sample a certain number of pixels from hyperspectral image for training and use the rest to evaluate performance of proposed network.The proposed network contains approximately 1000K parameters which is learned from several hundreds of training pixels from material category. They firstly set a base learning rate as 0.001. The base learning rate is decreased to 0.0001 after 33,333 iterations and is further reduced to 0.00001 after 66,666 iterations. To learn the network, the last argmax layer replaced by a softmax layer commonly used for learning convolutional layers. The 1st, 2nd, and 9th convolutional layers are initialized from a zero-mean. Authors of this paper use 1600 images for trainings and 6904 for testing. In this proposed work they built a fully convolutional neural network with a total of 9 layers, which is much deeper than other existing convolutional networks for HSI classification.The multiscale filter bank consists of three convolutional filters with different sizes: two filters (3×3 and 5×5) are used to exploit local spatial correlations while 1×1 is used to address spectral correlations The classification performance also shows that the proposed network with two different learning modules performs the one with only one module, which supports the effectiveness of the residual learning incorporated into the proposed network.

Paper 7:

Title:

A Study on CNN Transfer Learning for Image Classification

Author(s):

Mahbub Hussain, Jordan J. Bird, and Diego R. Faria

Literature Review:

Many image classification models have been introduced to help issue of recognition accuracy. image classification is one of the problems in Computer Vision field with the large variety of practical applications. Examples include: object recognition for robotic manipulation, pedestrian or obstacle detection for autonomous vehicles, among others. A lot of attention has been associated with Machine Learning, specifically neural networks such as the Convolutional Neural Network (CNN) winning image classification competitions. These work proposes the study and investigation of such a CNN architecture model. the model is evaluated, and the results are compared to some state-of-the-art approaches. Proposed method includes three tests The author used three tests to answer the following questions: Does Transfer Learning help improve the accuracy of a CNN? Does the no.of epochs (training steps) improve accuracy? Does the number of images per class in a dataset influence accuracy? Does the type of image in a dataset effect the accuracy? A series of tests were conducted to determine the usability of such a technique and whether it could be applied to different sets of data. In turn, we could prove the usefulness of Transfer Learning asthe results from the tests proved retraining the Inception-v3 model on the CIFAR-10 dataset resulted in better results compared to that stated in the previous state-of-the-art works, whereby authors did not use Transfer Learning and instead used a CNN trained on the same dataset (CIFAR-10) from scratch

Paper 8: (2020)

Automatic identification for field butterflies by convolutional neural networks

Author(s):

Ayad Saad Almryad, Hakan Kutucu

Literature Review:

In this paper, the authors have proposed an automated methodology for identification and classification of butterfly species using neural networks. They collected around forty thousand images of different butterfly species taken with different shooting angle, positions of butterflies and different background complexity. They used Convolutional Neural Network (CNN) to identify the butterfly species. This paper also provides a comparative study of the experimental results obtained from using three different network structures. In this study, various convolution neural network based on learning transfer methods are used such as VGG16, VGG19 and ResNet50 and fine- tuned transfer learning methods are utilised for classification process of taken butterfly images. Here, nine hundred sample images were used for each and every class. Again, each class is divided into two parts automatically into training part and testing part. Since data of training and testing process would be changed in each and every run, the training part is performed for five times for each dataset. Average accuracy values for hundred epochs are interpreted in this paper. VGG16, VGG19 and ResNet50 deep learning models are ran on GPU to shorten the time for learning and testing process. The loss and accuracy curves obtained are also interpreted in this paper. The

Paper 9: (2019)
Butterfly Recognition Based on Faster R-CNN
Author(s):
Ruoyan Zhao, Cuixia Li, Shuai Ye and Xinru Fang
Literature Review:
In this paper, the entire process starting from butterfly dataset collection to dataset processing for recognition and classification of butterfly species using Region based Convolutional Neural Network is described briefly. The task of detecting and classification of butterflies involve identification of the location of butterflies accurately in a given input image. The R-CNN algorithm helps to find out area of the image that contains the object in which we are interested in and then normalise the area into the required format for CNN input. Fast R-CNN mainly solves two major problems which include it proposes a regional recommendation network (RCN) to produce required regions and the other benefit is it makes Fast R-CNN and RPN to share parameters by alternate training phase. R-CNN uses a set of Convolutional, Relu and pooling layers for feature extraction. In this proposed methodology, an input image of arbitrary size let us say P*Q is scaled to a fixed size image let us say M*N and the sent to the network. Convolutional layers contain nearly thirteen convolutional layers, thirteen relu and four pooling layers. RPN, first passes by convolution of 3*3 and then generates regression offset values respectively and then calculates the proposals. The ROI pooling layer uses these generated proposals to extract features and send these features are sent to full-connection and softmax networks for classification of butterfly species. The authors interpreted that experimental results shows that automatic butterfly recognition system based on Regional recommendation network has an average accuracy of 70.4%. The authors made some conclusions that during the testing process, one must ensure that the sample is sufficient and also the ensure the randomness of samples to achieve high classification as well as identification accuracy.

Paper 10: (2019)
Butterfly Family Detection and Identification Using Convolutional Neural Network for Lepidopterology
Author(s):
Badrul Aiman Bakri, Zaaba Ahmad, Shahirah Mohamed Hatim
Literature Review:

Lepidopterology is a branch of entomology (study of insects) focuses on scientific study of butterfly's families and moths. The authors made a study on latest image classification technique which aims to acquire high accuracy by using Convolutional Neural Network. They proposed a frame design for butterfly detection and identification system using CNN. They interpreted that the accuracy of this method is 92.7% from training phase and 62.5% from actual system implementation results. The steps in the proposed methodology include data collection by mining the butterfly images from google, pre-processing, understanding and interpreting the images and finally making assumptions of the highly complicated data that be compared by using machines to draw some conclusions. In pre-processing, the images are being segmented and improving the quality of images for better identification of butterflies. After Image pre-processing, the next step is recognition process which integrates CNN to identify the visual image for recognition. CNN does not require manual feature extraction process which in turn eliminates the need for human effort. The project proposed in this paper is developed using Tensorflow in Ubuntu OS and interface is designed using HTML connected to Python script. The experimental results in this paper show that CNN can identify the butterflies with more than 90% accuracy with some learning saturation of five hundred cycles while testing phase results show 62.5% of accuracy in the process of predicting new datasets.

(Year, Authors)	Methodology	Advantages	Issues	Metrics used
(2018, Muthukrishnan Ramprasath, M.Vijay Anand, Shanmugasundaram Hariharan)	Deep learning,feature extraction by LBP and Convolutional Neural Network, SVM for image classification.	1.It provides the accuracy rate 98%. 2.It is suitable for grayscale images. 3.Provide accurate results of classification of images.	1.Not suitable for colored images. 2.Can't classify large size images. 3.Image segmentation can't be done.	1.Accuracy 2.Precision 3.Recall

(Year, Authors)	Methodology	Advantages	Issues	Metrics used
(2019, Nur Nabila Kamaron Arzar, Nurbaity Sabri, Nur Farahin Mohd Johari, Anis Amilah Shari, Mohd Rahmat Mohd Noordin, Shafaf Ibrahim)	GoogLeNet (pre- trained model of CNN),Confusion matrix for accuracy calculation,Soft max classifier for image classification	1.It is specially designed for identification of butterfly species. 2. Overall classification accuracy successfully achieves 97.5%. 3.Performance of classification of images can be found	1.This method can identify only four species butterflies. 2.This limited only 120 images.	1.Accuracy 2.Performance 3. True Positive (TP) 4.False Negative (FN) 5.False Positive (FP) 6.True Negative (TN)
(2019, Mingyuan Xin and Yong Wang)	Error back propagation algorithm, depth neural network for maximum interval minimum classification error, cross entropy and M3CE.	1. Recognition rate of this method is generally the same among different species. 2.This method obtained 80 % accuracy rate. 3.As the clarity of the image increases, feature extraction becomes easier.	1.Accuracy depends on the type of classifier we choose. 2. Its accuracy on the test set is only 69.47%	1.Accuracy 2.Errorness 3.Recall 4.Inception Score
(2020, Yanan Sun, Mengjie Zhang, Gary G. Yen and Jiancheng Lv)	Convolutional neural networks, genetic algorithms, optimization, evolutionary deep learning.	1.No need of expertise knowledge in CNN to classify images. 2.This method requires less computational resources.	1.Fitness evaluation of CNN's is relatively low	1.hypervolume 2.Inverted generational distance (IGD) 3.Precision

(Year, Authors)	Methodology	Advantages	Issues	Metrics used
(2020, Volodymyr Gorokhovatskyi, Iryna Tvoroshenko)	Kohonen Network is used for Image classification by introducing a dataspace based on system of etalon centres and data convolution.	The proposed methodology helps to ensure high classification efficiency percentage with high performance and makes it possible to use in a wide range of real- time applications.	Since they are using Kohonen network, it is difficult to predict what input weights to be used. Further performance improvements in the proposed methodology can be achieved using supervised training approaches.	1. Classification efficiency 2. Accuracy 3. Classification error
(2018, Hyungtae Lee, Heesng Kwon)	stochastic gradient descent (SGD), residual learning, CNNs architectures	they built a fully convolutional neural network with a total of 9 layers, which is much deeper than other existing convolutional networks for HIS classification.	To evaluate how the proposed network performs for pixels near boundaries between different classes	1.Accuracy 2.Precision 3.recall

(Year, Authors)	Methodology	Advantages	Issues	Metrics used
(2018, Mahbub Hussain, Jordan J. Bird, and Diego R. Faria)	CNN , Machine Learning, Inception-v3 models, Caltech Face model	this provides a solid basis to advance the use of Transfer Learning in not only the model presented but other deep neural networks	possibility would be combining a CNN with a Long-Short Term Memory (LSTM), this may be multifaceted, but in theory should help achieving better results and efficiency.	1.Accuracy 2.Precision 3.recall
(2020, Ayad Saad Almryad, Hakan Kutucu)	Deep Convolutional Neural Networksbased on VGG (VGG16 & VGG19) andResNet (ResNet-50) architecturesand Softmaxfunction is used to calculate the accuracy.	This convolutional neural network has very little dependence on pre- processing and it has high accuracy among all algorithms that predicts images.	The proposed methodology did not achieve complete success for both training and testing data because some images have few problems like position of butterflies, shooting angle and some background complexity.	1. Test Accuracy 2. Train Accuracy 3. Precision 4. Success Rate

(Year, Authors)	Methodology	Advantages	Issues	Metrics used
(2019, Ruoyan Zhao, Cuixia Li, Shuai Ye and Xinru Fang)	Whole process frombutterfly dataset collection to butterfly classification is done using Faster Region based Convolutional Neural Network (R- CNN)	The scheme quickly generates candidate regions and it integrates feature extraction, regression and classification into a network, which improves the classification speed.	The accuracy level achievedin this schemeis lower compared to others. The scheme involves many steps and difficult to implement.	1. Object detection time 2. Recognition 3. Accuracy
(2019, Badrul Aiman Bakri, Zaaba Ahmad, Shahirah Mohamed Hatim)	Convolutional Neural Network is used for identification and extracting features of butterflies and Data cleaning, Image compressing techniques are used for pre-processing the image	The performance and accuracy of proposed identification system has high accuracy of 92.7% since the image collected is ran through 500 cycles or epochs of training.	The proposed system is able to detect the butterfly species but it is highly depended on the image captured and the position of the butterfly. This methodology is limited to smaller dataset.	1. Accuracy 2. Precisio 3. Recall 4. Performac e 5. Consistec y

Convolutional Neural Networks for Image Classification

What Is Image Classification?

Image classification is the process of labeling images according to predefined categories. The process of image classification is based on supervised learning. An image classification model is fed a set of images within a specific category. Based on this set, the algorithm learns which class the test images belong to, and can then predict the correct class of future image inputs, and can even measure how accurate the predictions are.

This process introduces multiple challenges, including scale variation, viewpoint variation, intra-class variation, image deformation, image occlusion, illumination conditions and background clutter.

The Use of Convolutional Neural Networks for Image Classification

The CNN approach is based on the idea that the model function properly based on a local understanding of the image. It uses fewer parameters compared to a fully connected network by reusing the same parameter numerous times. While a fully connected network generates weights from each pixel on the image, a convolutional neural network generates just enough weights to scan a small area of the image at any given time. This approach is beneficial for the training process the fewer parameters within the network, the better it performs. Additionally, since the model requires less amount of data, it is also able to train faster.

Additional Data Transformation for training:

Deep neural networks are greatly improved with the addition of more training data. When more training data is not available, transformations to the existing training data which reflect the variation found in images can synthetically increase the training set size. In the previous Imagenet classification system, three types of image transformations were used to augment the training set. The first was to take a randomly located crop of 224x224 pixels from a 256x256 pixel image capturing some translation invariance. The second was to flip the image horizontally to capture the reflection invariance. The final data transformation was to add randomly generated lighting which tries to capture invariance to the change in lighting and minor color variation. We add additional transformations that extend the translation

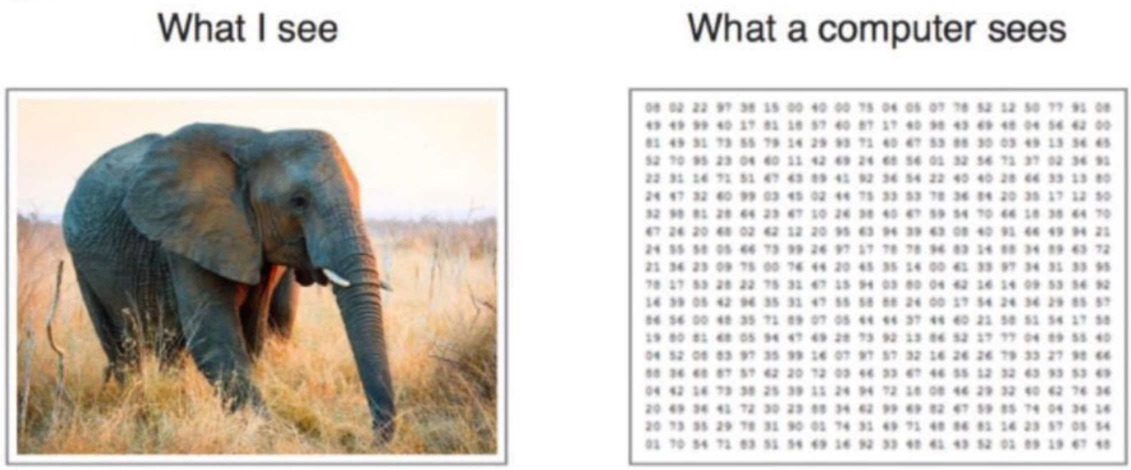
Deep Learning for Image Classification

Deep learning, a subset of Artificial Intelligence (AI), uses large datasets to recognize patterns within input images and produce meaningful classes with which to label the images. A common deep learning method for image classification is to train an Artificial Neural Network (ANN) to process input images and generate an output with a class for the image. The challenge with deep learning for image classification is that it can take a long time to train artificial neural networks for this task. However, Convolutional Neural Networks (CNNs) excel at this type of task.

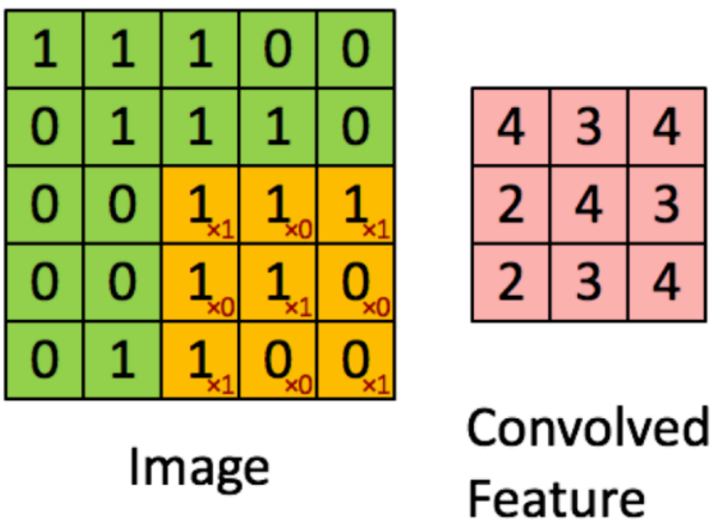
Image Classification with Convolutional Neural Networks :

Convolutional neural networks (CNN) is a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification.
Instead of the image, the computer sees an array of pixels. For example, if image size

is 300 x 300. In this case, the size of the array will be 300x300x3. Where 300 is width, next 300 is height and 3 is RGB channel values. The computer is assigned a value from 0 to 255 to each of these numbers. This value describes the intensity of the pixel at each point.



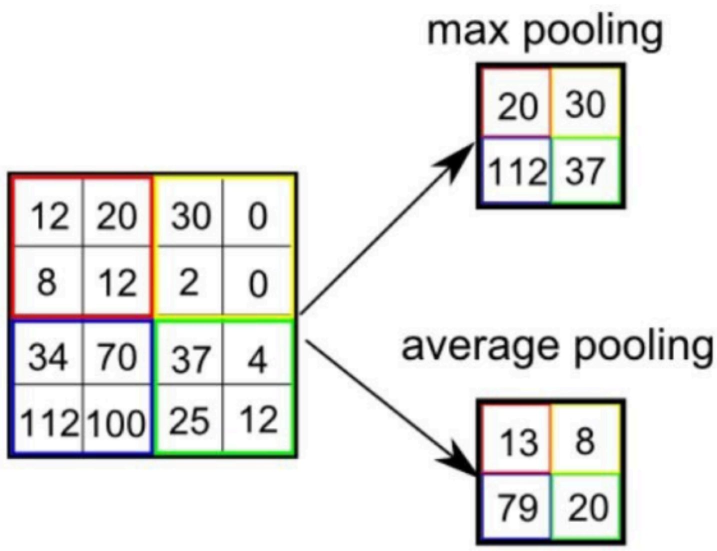
The Convolution layer is always the first. The image (matrix with pixel values) is entered into it. Imagine that the reading of the input matrix begins at the top left of image. Next the software selects a smaller matrix there, which is called a filter (or neuron, or core). Then the filter produces convolution, i.e. moves along the input image.



The filter's task is to multiply its values by the original pixel values. All these multiplications are summed up. One number is obtained in the end. Since the filter has read the image only in the upper left corner, it moves further and further right by 1 unit performing a similar operation. After passing the filter across all positions, a matrix is obtained, but smaller then a input matrix.

The nonlinear layer :
Is added after each convolution operation. It has an activation function, which brings nonlinear property. Without this property a network would not be sufficiently intense and will not be able to model the response variable.

The pooling layer :
Follows the nonlinear layer. It works with width and height of the image and performs a downsampling operation on them. As a result the image volume is reduced.



Types of Pooling

ImageDataGenerator has the following arguments:
1. rotation_range — which is used for random rotations, given in degrees in the range from 0 to 180
2. width_shift_range — which is shown in fraction of total width, used for random horizontal shifts
3. height_shift_range — which is the same as width_shift_range, but with height
4. shear_range — shear intensity, used for linear mapping that displaces each point in a fixed direction

5. zoom_range — use for random zooming
6. horizontal_flip — unlike other arguments has boolean type, used for randomly flipping inputs horizontally
7. fill_mode — can be “constant”, “reflect”, “wrap” or “nearest” as in this case; indicates the method of filling the newly formed pixels.
This architecture was made on the principle of convolutional neural networks. It consists of 3 groups of layers, where the convolution layers (Conv 2D) alternate with the nonlinear layers (Relu) and the pooling layers (Max Pooling 2D)
The author assembled and trained the CNN model to classify photographs of cars and elephants. The author tested that this model works really well with a small number of photos. The author determined that 10 epochs are enough for a successful training of the mode

Methodologies

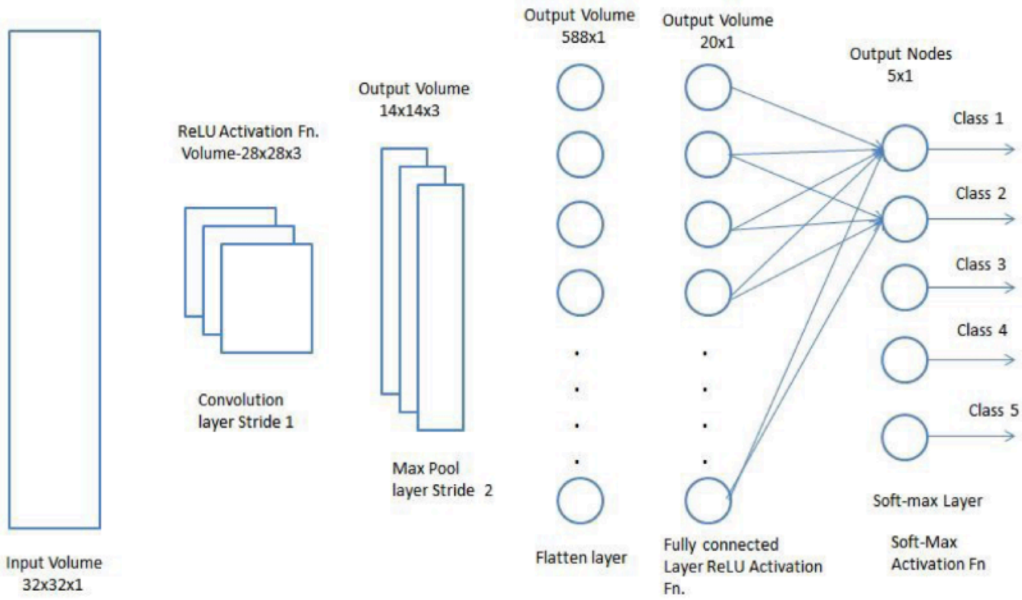
Images of Butterflies:

A total of eight hundred and thirty-two images of butterflies were collected from online database and some of it were captured using camera. The size of images obtained are from different kind of sizes but then resize to 128x128 pixels.

Convolutional Neural Network (CNN):

In this project, for training and testing process, eight hundred and thirty-two butterfly images are used to test the accuracy of whole system. The images of the butterflies will pass through the first architectural layer, which is a convolutional layer in which all features are extracted and the result is transferred to the next layer. The next layer would be a pooling layer that reduces image volumes without data loss and the max pooling layer is used to get most maximum values of each divided region. The next layer is a fully connected layer that connects this layer directly to each node in the previous layer. Fully connected layer transforms the data dimension to allow data to be linearly classified. It modifies the input image into vector resulting output in array size and the number then apply to Softmax to change to
probability. For softmax layer, the result that obtained from fully- connected layer will be processed to an array of probabilities for each of the category. The maximum probability is the class that it predicts. Last three layers of GoogLeNet with names 'loss3- classifier', 'prob', and 'output' replaced with ‘fullyConnectedLayer’, with

number of classes four as there are four types of butterflies, ‘softmaxLayer’ and ‘classificationLayer’.



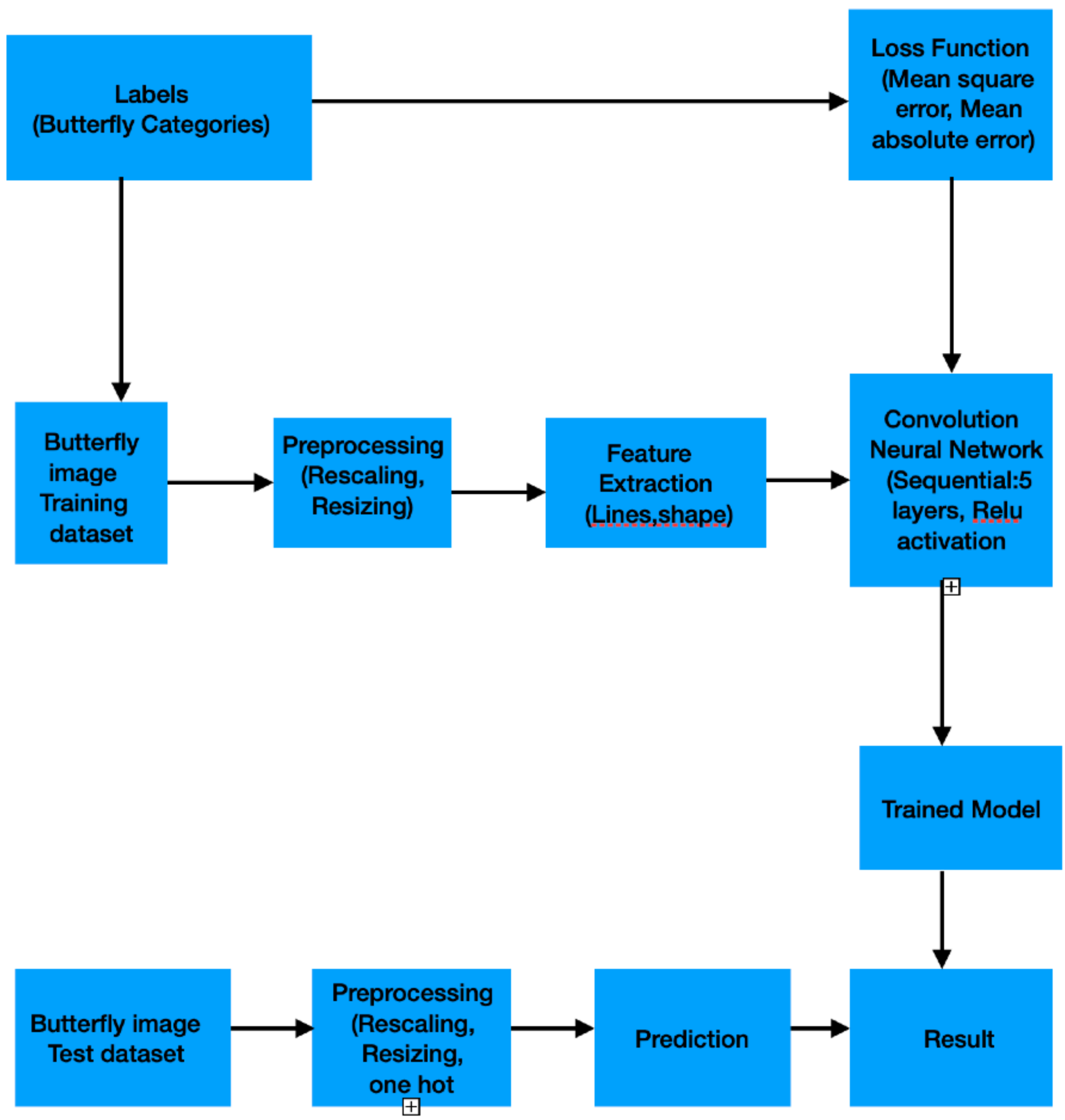
<u>Accuracy Percentage:</u>

The accuracy percentage is calculated using confusion matrix. Confusion matrix consists of four components which are True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN). TP is when it predicted yes for the example for the class of the butterfly. TN is when it predicted no for the class of the butterfly. FP is when it predicted yes but class of butterfly is wrong. FN is when it predicted no but butterfly class is true. True value is retrieved from each row and column that have the same species of butterflies. False value is retrieved from each row and column with different species of butterflies. Classification accuracy for all species of butterfly is calculated using

$$\text{Accuracy (\%)} = \frac{\text{Total TRUE value}}{\text{Total images testing}} \times 100$$

General Architecture:

Block diagram of Image classification using Convolution neural network:



Software Requirement Specifications

- Python & google colab is used for the free availability of 12 GB of GPU that can be used for quickly training the dataset.
  Tensorflow is used for neural networks applications.
- Keras is for defining the 2D conv layers

Implementation and Source code:

```
from google.colab import drive
drive.mount('/content/drive',force_remount=True)

import os
os.chdir('/content/drive/MyDrive/leedsbutterfly')

! ls

! unzip 'butterfly-dataset.zip'

from pandas import DataFrame
Categories_Dictionary={'001': 'Danaus_plexippus','002': 'Heliconius_charitonius',
'003': 'Heliconius_erato','004': 'Junonia_coenia','005': 'Lycaena_phlaeas',
'006': 'Nymphalis_antiopa','007': 'Papilio_cresphontes','008': 'Pieris_rapae',
'009': 'Vanessa_atalanta','010': 'Vanessa_cardui'}
Categories = []
Filenames = os.listdir('/content/drive/My Drive/leedsbutterfly/images/')
for Filename in Filenames:
    Category = Filename.split(".")[0]
    Categories.append(Categories_Dictionary[Category[0:3]])
DF=DataFrame({'Filename':Filenames,'Category':Categories})
DF.head(10)

os.chdir('/content/drive/My Drive/')
if os.path.exists('NewDataset'):
  ! rm -r NewDataset
os.makedirs('NewDataset')

os.chdir('/content/drive/My Drive/NewDataset')
```

```
Directories=('train','valid','test')
for i in Directories:
  os.makedirs(i)

Unique_Categories=DF['Category'].unique()

for i in Directories:
  os.chdir('/content/drive/My Drive/NewDataset/'+i)
  for j in Unique_Categories:
    os.mkdir(j)

from sklearn.model_selection import train_test_split
X_train , x_test , Y_train , y_test = train_test_split(DF['Filename'] ,
DF['Category'] ,test_size = 0.2)

import shutil
for i in range(len(X_train)):
  Source='/content/drive/My Drive/leedsbutterfly/images/'+X_train.iloc[i]
  Destination='/content/drive/My Drive/NewDataset/train/'+Y_train.iloc[i]
+'/'+X_train.iloc[i]
  print('Processing image: ',i+1,'/',len(X_train))
  shutil.copy(Source,Destination)

X_valid , x_test , Y_valid , y_test = train_test_split(x_test, y_test,test_size = 0.1)
print('Size of Validation set = ',len(X_valid))
print('Size of Test set = ',len(x_test))

for i in range(len(X_valid)):
  Source='/content/drive/My Drive/leedsbutterfly/images/'+X_valid.iloc[i]
  Destination='/content/drive/My Drive/NewDataset/valid/'+Y_valid.iloc[i]
+'/'+X_valid.iloc[i]
  print('Processing image: ',i+1,'/',len(X_valid))
  shutil.copy(Source,Destination)

for i in range(len(x_test)):
  Source='/content/drive/My Drive/leedsbutterfly/images/'+x_test.iloc[i]
  Destination='/content/drive/My Drive/NewDataset/test/'+y_test.iloc[i]
+'/'+x_test.iloc[i]
```

```
  print('Processing image: ',i+1,'/',len(x_test))
  shutil.copy(Source,Destination)

import tensorflow as tf
Device_Name=tf.test.gpu_device_name()
if Device_Name!='/device:GPU:0':
  raise SystemError('GPU device not found')

import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten,Dense
from keras.optimizers import RMSprop

model = Sequential()
model.add(Conv2D(32, (5,5), activation = 'relu', input_shape=(128,128,3)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dropout(0.25))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(10, activation='softmax'))
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.0001,decay=1e-6),
              metrics=['accuracy','mae','mse'])
model.summary()

from keras.preprocessing.image import ImageDataGenerator
import os

Batch_Size_Train=5
```

```
Batch_Size_Valid=2

Train_Path='/content/drive/My Drive/NewDataset/train'
Train_Size=0
for i in os.listdir(Train_Path):
  Train_Size+=len(os.listdir(Train_Path+'/'+i))
print(Train_Size)

Valid_Path='/content/drive/My Drive/NewDataset/valid'
Valid_Size=0
for i in os.listdir(Valid_Path):
  Valid_Size+=len(os.listdir(Valid_Path+'/'+i))
print(Valid_Size)

Steps_Per_Epoch_Train=Train_Size//Batch_Size_Train
Steps_Per_Epoch_Valid=Valid_Size//Batch_Size_Valid
print(Steps_Per_Epoch_Train)
print(Steps_Per_Epoch_Valid)

IDG_Train=ImageDataGenerator(rotation_range=40,width_shift_range=0.2,
        height_shift_range=0.2,
rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True,fill_mode='ne
arest')
IDG_Valid=ImageDataGenerator(rescale=1/255)
Train_Data=IDG_Train.flow_from_directory(Train_Path,target_size=(128,128),batch
_size=Batch_Size_Train,class_mode='categorical')
Valid_Data=IDG_Valid.flow_from_directory(Valid_Path,target_size=(128,128),batch
_size=Batch_Size_Valid,class_mode='categorical')

model.fit_generator(Train_Data,steps_per_epoch=Steps_Per_Epoch_Train,epochs=1
20,
                validation_data=Valid_Data,validation_steps=Steps_Per_Epoch_Valid)

from keras.models import load_model
model = load_model('/content/drive/MyDrive/model.h5')

from keras.preprocessing.image import ImageDataGenerator
```

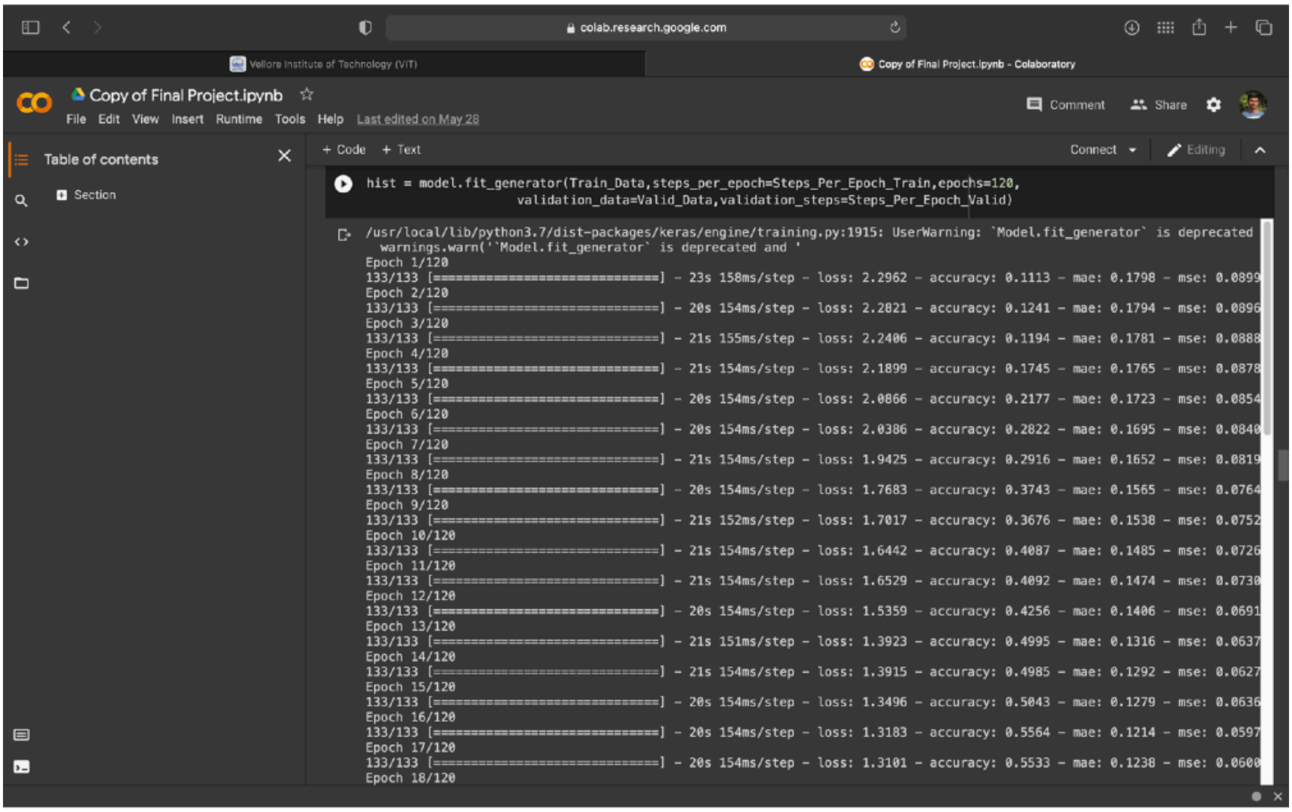
```
Test_Path='/content/drive/My Drive/NewDataset/test'
Test_Size=0
for i in os.listdir(Test_Path):
  Test_Size+=len(os.listdir(Test_Path+'/'+i))
print(Test_Size)

Batch_Size_Test=Test_Size

Steps_Per_Epoch_Test=Test_Size//Batch_Size_Test
Steps_Per_Epoch_Test

Test_Data=ImageDataGenerator(rescale=1/255).flow_from_directory(Test_Path,targe
t_size=(128,128),batch_size=Batch_Size_Test,class_mode='categorical')

import matplotlib.pyplot as plt

Categories=('Danaus_plexippus','Heliconius_charitonius','Heliconius_erato','Junonia_
coenia',
           'Lycaena_phlaeas','Nymphalis_antiopa','Papilio_cresphontes','Pieris_rapae',
           'Vanessa_atalanta','Vanessa_cardui')

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
Label_Encoding=LabelEncoder().fit_transform(Categories).reshape(-1,1)
One_Hot_Encoding=OneHotEncoder().fit_transform(Label_Encoding).toarray()
Categories_Dictionary={tuple(One_Hot_Encoding[i]):Categories[i] for i in
range(len(Categories))}
print(Categories_Dictionary)

Test_Images,Test_Labels=next(Test_Data)

Original=[]
plt.figure(figsize=(15,20))
for i in range(len(Test_Images)):
  plt.subplot(5,5,i+1)
  plt.imshow(Test_Images[i])
  Original.append(Categories_Dictionary[tuple(Test_Labels[i])])
  plt.xlabel(Original[i])
plt.show()
```

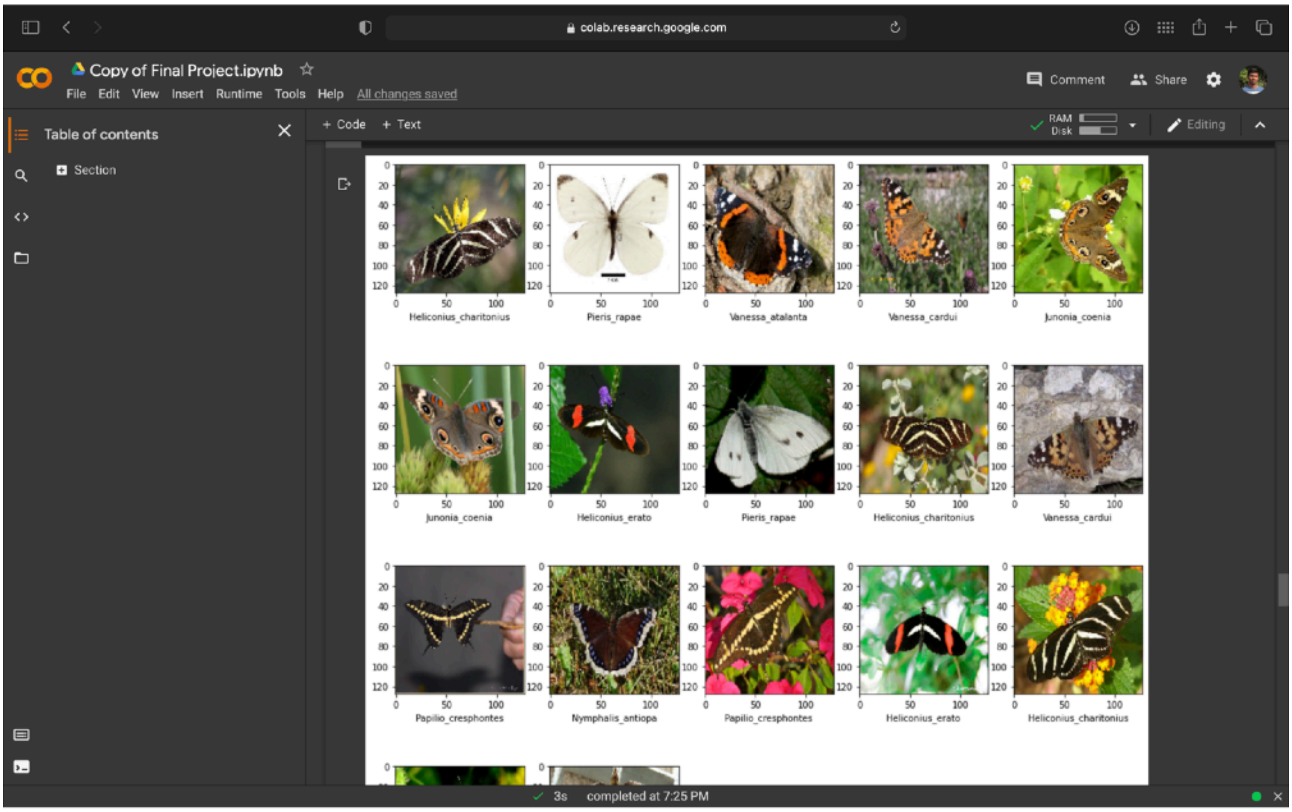
```
Predictions=model.predict_generator(Test_Data,steps=Steps_Per_Epoch_Test,verbos
e=1)

Predicted=[]
plt.figure(figsize=(15,20))
for i in range(len(Predictions)):
  Predictions[i]=(Predictions[i]>=0.5)
  if tuple(Predictions[i]) in Categories_Dictionary:
    Predicted.append(Categories_Dictionary[tuple(Predictions[i])])
  else:
    Predicted.append("Unable to predict")
  plt.subplot(5,5,i+1)
  plt.imshow(Test_Images[i])
  plt.xlabel(Predicted[i])
plt.show()

Loss,Accuracy,MSE,MAE=model.evaluate(Test_Images,Test_Labels,verbose=1)

print('Accuracy: ',Accuracy)
print('Loss: ',Loss)
print('MSE: ',MSE)
print('MAE: ',MAE)

Accuracy=0
for i in range(len(Original)):
  print(Predicted[i])
  if Original[i]==Predicted[i]:
    Accuracy+=1
    print("Correctly classified image: ",i+1)
  else:
    print("Wrongly classified "+Original[i]+' as '+Predicted[i])
    plt.figure()
    plt.imshow(Test_Images[i])
    plt.xlabel("Wrongly classified "+Original[i]+' as '+Predicted[i])
print('\nAccuracy = ',Accuracy,'/',len(Predicted),' = ',Accuracy/
len(Predicted)*100,'%')
plt.show()
```

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input,
decode_predictions


# Helper libraries
import numpy as np
import matplotlib.pyplot as plt

img = image.load_img("/content/drive/MyDrive/NewDataset/train/
Nymphalis_antiopa/0060029.png", target_size=(128,128))
img_array = image.img_to_array(img)
datagen = ImageDataGenerator(rescale=1./255)

img_batch = np.expand_dims(img_array, axis=0)
# img_preprocessed = preprocess_input(image_batch)
testing_image = datagen.flow(img_batch)
prediction = model.predict(testing_image)
# print(prediction)
prediction = (prediction>=0.5)
print(Categories_Dictionary[tuple(prediction[0])])

model.save('/drive/My Drive/trained_model.h5')
```

Result:

Image classification using CNN is successful implemented for Butterfly Dataset with accuracy of 94.13 . The data is trained with 120 epochs of each size of 133. Butterfly images are predicted correctly with trained model.





The trained model is tested with test dataset and achieved accuracy of 94.11%

```
[ ] print('Accuracy: ',Accuracy)
    print('Loss: ',Loss)
    print('MSE: ',MSE)
    print('MAE: ',MAE)

    Accuracy:  0.9411764740943909
    Loss:  0.366489976644516
    MSE:  0.013922380283474922
    MAE:  0.011857371777296066
```

Conclusion:

This paper proposed the implementation of Convolutional Neural Network (CNN) technique for identifying the species of butterflies. The proposed techniques were effectively implemented to identify images of butterflies. Overall classification accuracy successfully achieves 94.11%. Therefore, it can be concluded that the implementation of CNN technique for classification of butterflies is found to be successful. Yet, further work of adding more species of butterflies is suggested in the future.

References

[1] Almryad, A. S., & Kutucu, H. (2020). Automatic identification for field butterflies by convolutional neural networks. Engineering Science and Technology, an International Journal, 23(1), 189-195.

[2] Bakri, B. A., Ahmad, Z., & Hatim, S. M. (2019). Butterfly family detection and identification using convolutional neural network for lepidopterology. International Journal of Recent Technology and Engineering, 8(2 Spec), 635- 640.

[3] Zhao, R., Li, C., Ye, S., & Fang, X. (2019, March). Butterfly recognition based on faster R-CNN. In Journal of Physics: Conference Series (Vol. 1176, No. 3, p. 032048). IOP Publishing.

[4] Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image Classification Based on the Kohonen Network and the Data Space Modification.

[5] Shah, P., & Vayada, M. G. (2015). Review on Satellite Image Classification using Fuzzy Logic. International Journal of Science and Research (IJSR), 4(12), 1245-1248.

[6] Jaswal, D., Vishvanathan, S., & Kp, S. (2014, June). Image Classification Using Convolutional Neural Networks.

[7] Ramprasath, M., Anand, M. V., & Hariharan, S. (2018). Image classification using convolutional neural networks. International Journal of Pure and Applied Mathematics, 119(17), 1307-1319.

[8] Arzar, N. N. K., Sabri, N., Johari, N. F. M., Shari, A. A., Noordin, M. R. M., & Ibrahim, S. (2019, June). Butterfly Species Identification Using Convolutional Neural Network (CNN). In 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) (pp. 221-224). IEEE.

[9] Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network.EURASIP Journal on Image and Video Processing, 2019(1), 1-11.

[10] Samruddhi N Hegde, Manjesh R, P Sindhura, Reema Rodrigues, Sheethal A (2020, May 5). Identification Of Butterfly Speices Using VGG-16.