Animal Classification using Deep Learning Techniques

By

DEEPIKA B

(17PIT001)

A Dissertation submitted to

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

COIMBATORE - 641 043

In Partial Fulfilment of the Requirements for the Degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

APRIL 2019

Animal Classification Using Deep Learning Techniques

By

DEEPIKA B

(17PIT001)

A Dissertation submitted to

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

COIMBATORE - 641 043

In Partial Fulfilment of the Requirements for the Degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

APRIL - 2019

CERTIFIED AS A BONAFIDE RESEARCH WORK

Signature of the Head of the Department

Department of Information Technology
Avinashilingam Institute For Home Science
and Higher Education for Women
Coimbatore - 641 043

22/4/2019

Signature of the Superviso

CERTIFICATE

CELESTIAL V SOLUTIONS

Date : 10-April- 2019

PROJECT COMPLETION CERTIFICATE

This is to certify that **B.DEEPIKA (REG NO:17PIT001)** student of final year master of science(information technology).**Avinashilingam Institute For Home Science And Higher Education For Women, Coimbatore** has associated with us for project from 5th dec 2018 to 5th April 2019. she is working under the project of **"Animal classification using deep learning technique"** in our company located at Bangalore.

We wish her all the best for her future endeavours.

With Regards

For Celestial V Solutions

Authorized Signatory

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I owe my sincere thanks to **Lord Almighty** and **My lovable parents** for showering their generous blessings upon me in all endeavours.

I wish to express my gratitude to **Shri, Dr.P.R.Krishnakumar,** Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing the facilities to conduct this study.

I extend my thanks to **Hon.Col.Dr. (Tmt.) Premavathy Vijayan,** M.Sc., M.Ed., Dip.Spk.Edn., M.Phil., Ph.D., Vice Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing flamboyant help towards the completion of the study.

I record my deep sense of gratitude and indebtedness to **Dr. (Tmt.) S.Kowsalya,** M.Sc., M.Phil., Ph.D., Registrar, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing adequate help for the study.

I grateful record my sincere thanks to **Dr. (Mrs.) K. Udaya Chandrika** M.Sc., M.Phil., Ph.D., Dean and Professor, School of Physical Sciences & Computational of Sciences, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for timely help rendered throughout the course of this work.

I express my sincere thanks to **Dr. F.Paulin MCA.,M.Phil., Ph.D.,** Assistant Professor, Department of Information Technology, for her encouragement and inspiration to complete the project.

I heartily thank my esteemed project guide **Dr. D. Shanmuga Priya, M.Sc., M.Phil., Ph.D.,SET, Head, Department of Information Technology,** for imparting tremendous assistance and well-timed support for triumph of our project.

I express my honorable thanks to our project coordinator **Dr. T. Jayamalar, MCA, M.Phil Ph.D., Assistant Professor, Department of Information Technology,** for her kind advice and knowledgeable suggestions which helped us to complete our project successfully.

I sincerely thank all **the Staff members** of Department of Information Technology, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for their help and support.

I would like to express my special thanks to **my parents, my friends** and all **my well-wishers** for their constant encouragement, support and help in carrying out this work successfully.

DEEPIKA B

SYNOPSIS

SYNOPSIS

A real-world animal biometric system that detects and describes animal life in image data is an emerging subject in machine vision. These systems develop computer vision approaches for the classification of animals. A novel method for animal face classification based on one of the popular convolutional neural network (CNN) features. CNN can automatically extract features, learn and classify them. The proposed method may also be used in other areas of image classification and object recognition. The experimental results show that automatic feature extraction in CNN is better than other simple feature extraction techniques.

CONTENTS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

I.INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Animal recognition and classification is an important area which has not been discussed rapidly. Animal classification which relies on the problem of distinguishing images of different animal species is an easy task for humans, but evidence suggests that even in simple cases like cats and dogs, it is difficult to distinguish them automatically. Animals have flexible structure that could self-mask and usually they appear in complex scene. Also, as all objects, they may appear under different illumination conditions, viewpoints and scales. There are attempts to apply recognition methods on images of animals but the specific problem of animal categorization has recently attracted limited interest.

Many existing methods showing promising results for human fare recognition cannot properly represent the diversity of animal classes with complex intra-class variability and inter-class similarity. There are several kinds of approaches for solving this problem with each one having its advantages and disadvantages. The first approach constructs complex features which represents and discriminates sample images better but creating such a feature is complicated and it is problem dependent. The second approach combines the extracted features from different methods and concatenates them to build a more powerful feature vector. Increasing the size of feature space causes increased problem computation cost. Instead of using complex representation, the information is consolidated from different classifiers and a decision is made according to it. This method is known as score-level fusion.

Observing wild animals in their natural environments is a central task in ecology. The fast growth of human population and the endless pursuit of economic development are making over-exploitation of natural resources, causing rapid, novel and substantial changes to Earth's ecosystems. An increasing area of land surface has been transformed by human action, altering wildlife population, habit at and behavior. More seriously, many wild species on Earth have been driven to extinction, and many species are introduced into new areas where they can disrupt both natural and human systems. Monitoring wild animals, therefore, is essential as it provides researchers evidences to inform conservation and management decisions to maintain diverse, balanced and sustainable ecosystems in the face of those changes, Various modern technologies have been developed for wild animal monitoring, including radio tracking, wireless sensor network tracking, satellite and global positioning

system (GPS) tracking, and monitoring by motion sensitive camera traps. Motion-triggered remote cameras or “camera traps” are an increasingly popular tool for wildlife monitoring, due to their novel features equipped, wider commercial availability, and the ease of deployment and operation. For instance, a typical covert camera model is capable of not only capturing high definition images in both day and night, but also collecting information of time, temperature and moon phase integrated in image data. In addition, generous and flexible camera settings allow tracking animals secretly and continuously.

Once being fully charged, a camera can snap thousands of consecutive images, providing a large volume of data. These specifications make camera traps a powerful tool for ecologists as they can document every aspect of wildlife. Visual data, if can be captured, is a rich source of information that provide scientists evidences to answer ecology-related scientific questions such as: what are the spatial distributions of rare animals, which species are being threatened and need protection such as bandicoot, which cohort of pest species, such as red fox and rabbit, need to be controlled these are examples of key questions to understand wild animals populations, ecological relationships and population dynamics. To this end, a recently widely-used approach by ecologists is to set up several camera traps in the wild to collect image data of wild animals in their natural habitats. In this project we take CIFAR-10 datasets. From this dataset we took 10 classifications and build the CNN training model. The output of the system is we can give an image from any of 10 classified trained model and this system will gives out the image belong to which category.

The overwhelming amounts of data from camera traps highlight the need for image processing automation. From data analysis and machine learning point of views, there are some immediate techniques to make wildlife identification automated classifier with manual object bounding on hand-crafted features, convolutional neural network (CNN) model with automatic object detection, or fine-tuning CNN models inheriting model weights pretrained on a very large scale dataset such as the ImageNet. These approaches addressed the problem of wildlife monitoring automation and demonstrated promisingly empirical results. However, two primary challenges, which inhibit the feasibility of an automated wildlife monitoring application in practice, are still remaining. The first obstacle is that, to obtain applicable image classification accuracy, an enormous amount of manual preprocessing is still required to input images for detecting and bounding animal objects. The second limitation is poor performance obtained by wildlife monitoring system, in spite of complete automation, requiring much more improvements for practical application.

1.2 ABOUT PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

1.2.1 FEATURES

Python's features include – All these.

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable** − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Python preferred over other data science tools

It is 'Pythonic' when the code is written in a fluent and natural style. Apart from that, it is also known for other features that have captured the imaginations of data science community.

Easy to learn:

The most alluring factor of Python is that anyone aspiring to learn this language can learn it easily and quickly. When compared to other data science languages like R, Python promotes a shorter learning curve and scores over others by promoting an easy-to-understand syntax.

Scalability:

When compared to other languages like R, Python has established a lead by emerging as a scalable language, and it is faster than other languages like Matlab and Stata. Python's scalability lies in the flexibility that it gives to solve problems, as in the case of YouTube that migrated to Python.

Choice of data science libraries:

The significant factor giving the push for Python is the variety of data science/data analytics libraries made available for the aspirants. Pandas, StatsModels, NumPy, SciPy, and Scikit-Learn, are some of the libraries well known in the data science community. Python does not stop with that as libraries have been growing over time. What you thought was a constraint a year ago would be addressed well by Python with a robust solution addressing problems of specific nature.

Python community:

One of the reasons for the phenomenal rise of Python is attributed to its ecosystem. As Python extends its reach to the data science community, more and more volunteers are creating data science libraries. This, in turn, has led the way for creating the most modern tools and processing in Python.

The widespread and involved community promotes easy access for aspirants who want to find solutions to their coding problems. Whatever queries you need, it is a click or a Google search away. Enthusiasts can also find access to professionals on Code mentor and Stack Overflow to find the right answers for their queries.

Graphics and visualization:

Python comes with varied visualization options. Matplotlib provides the solid foundation around which other libraries like Sea born, pandas plotting, and ggplot have been built. The visualization packages help you get a good sense of data, create charts, graphical plot and create web-ready.

REVIEW OF LITERATURE

II.REVIEW OF LITERATURE

Y H SharathKumar ,C D Divya (2014). The feature selection approach in animal classification. In this paper, they propose a model for automatic classification of Animals using different classifiers Nearest Neighbour, Probabilistic Neural Network and Symbolic. Animal images Mare segmented using maximal region merging segmentation. The Gabor features are extracted from segmented animal images. Discriminative texture features are then selected using the different feature selection algorithm like Sequential Forward Selection, Sequential Floating Forward Selection, Sequential Backward Selection and Sequential Floating Backward Selection.

Shahram Taheri1, Onsen Toygar1(2018). Animal classification using facial images with score-level fusion. These systems develop computer vision approaches for the classification of animals. This method utilises a score-level fusion of two different approaches; one uses CNN which can automatically extract features, learn and classify them; and the other one uses kernel Fisher analysis (KFA) for its feature extraction phase. The proposed method may also be used in other areas of image classification and object recognition.

Tibor Trnovszky, Patrik Kamencay,al.,(2017). Animal Recognition System Based on Convolutional Neural Network. This method is compared with well known image recognition methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Binary Patterns Histograms(LBPH) and Support Vector Machine (SVM). The main goal is to compare the overall recognition accuracy of the PCA, LDA, LBPH and SVM with proposed CNN method effect on overall animal recognition performance and outperforms other examined methods.

Hung Nguyen, Sarah J. Maclagan (2012). Animal Recognition and Identification with Deep Convolutional Neural Networks. Efficient and reliable monitoring of wild animals in their natural habitats is essential to inform conservation and management decisions processing such a large volume of images captured from camera traps manually is extremely expensive, time-consuming and also monotonous. Leveraging on recent advances in deep learning techniques in computer vision, propose in this paper a framework to build automated animal recognition in the wild, aiming at an automated wildlife monitoring system.

Parinita Badre, Siddhant Bandiwadekar (2015). Automatically Identifying Animals Using Deep Learning. Having accurate, detailed, and up-to-date information about wildlife location and behavior across broad geographic areas would revolutionize our ability to study, conserve, and manage species and ecosystems. In this project, they demonstrate that such data can be automatically extracted by deep neural networks (deep learning), which is a cutting-edge type of artificial intelligence to train neural networks that automatically identifies animals.

Shenghua Zhong, Yangliu(2011). Image classification is a well-known classical problem in multimedia content analysis. A novel deep learning model called bilinear deep belief network (BDBN) for image classification. To preserve the natural tensor structure of the image data, a novel deep architecture with greedy layer-wise reconstruction and global fine-tuning is proposed. Comparative experiments on three standard datasets show that the proposed algorithm outperforms both representative classification models and existing deep learning techniques.

Jyotsna Bankar, Nitin Gavai(2015). Animal Classification Using Machine-Learning Algorithm. Identification of similar types of objects in image processing now become regular task, but when dissimilar objects come into picture it becomes quite complex. Even it become more complicated when not just we need to identify objects but also categorizes them into their specific classes. We are using machine learning technique to classify the animal and put them into specified classes. In this paper, based on Inception-v3 model in Tens or Flow platform, we use the transfer learning technology to retrain the animal category datasets, which can greatly improve the accuracy of animal classification.

Guobin Chen, RolandKays(2015). Deep convolutional neural network based species recognition for wild animal monitoring. They proposed a novel deep convolutional neural network based species recognition algorithm for wild animal classification on very challenging camera-trap imagery data. This imagery data were captured with motion triggered camera trap and were segmented automatically using the state of the art graph-cut algorithm. For the comparison purpose, we use the traditional bag of visual words model as the base line species recognition algorithm. It is clear that the proposed deep convolutional neural network based species recognition achieves superior performance.

SimHiewMoi, Hishammuddin Asmuni.(2014). Multimodal biometrics: Weighted score level fusion based on non-ideal iris and face images. The iris and face are among the most promising biometric traits that can accurately identify a person because their unique textures can be swiftly extracted during the recognition process. Iris and face biometric authentication often deals with non-ideal scenarios such as off-angles, reflections, expression changes, variations in posing, or blurred images. These limitations imposed by unimodal biometrics can be overcome by incorporating multimodal biometrics.

Heydar Maboudi Afkham, Alireza Tavakoli Targhi (2008). Joint Visual Vocabulary For Animal Classification. This paper presents a method for visual object categorization based on encoding the joint textural information in objects and the surrounding background, and requiring no segmentation during recognition. The database in a set of experiments in which we compare the performance of our approach with a recently proposed method.

AdityaKhosla, Michael Bernstein (2014). ImageNet Large Scale Visual Recognition Challenge. The ImageNet Large Scale Visual Recognition Challenge is a benchmark in object category classification and detection on hundreds of object categories and millions of images. We discuss the challenges of collecting large-scale ground truth annotation, highlight key break throughs in categorical object recognition, provide a detailed analysis of the current state of the field of large-scale image classification and object detection, and compare the state of the art computer vision accuracy with human accuracy.

P. N. Druzhkov, V. D. Kustikova (2015). Survey of Deep Learning Methods and Software Tools for Image Classification and Object Detection. Deep learning methods for image classification and object detection are overviewed. In particular we consider such deep models as auto encoders, restricted Boltzmann machines and convolutional neural networks. Existing software packages for deep learning problems are compared. Image classification problem requires determining the category (class) that it belongs to. The problem is considerably complicated with the growth of categories count, if several object of different classes are present at the image and if the semantic classes'.

IlyaSutskever, Geoffrey E.Hinton (2017). ImageNet Classification with Deep Convolutional Neural Networks. They trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the

1000 different classes. To make training faster, we used non saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce over fitting in the fully connected layers we employed a recently developed regularization method called “dropout” that proved to be very effective.

Simonyan, K., Zisserman, A.(2015). Very deep convolutional networks for large scale image recognition. In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. They have made our two best-performing Net models publicly available to facilitate further research on the use of deep visual representations in computer vision.

He, K., Zhang, X., Ren, S., (2014). Pooling in deep convolutional networks for visual recognition. Existing deep convolutional neural networks (CNNs) require a fixed-size (e.g., 224×224) input image. This requirement is “artificial” and may reduce the recognition accuracy for the images or sub-images of an arbitrary size/scale. In this work, we equip the networks with another pooling strategy, “spatial pyramid pooling”, to eliminate the above requirement. The new network structure, called SPP-net, can generate a fixed-length representation regardless of image size/scale. Pyramid pooling is also robust to object deformations. With these advantages, SPP-net should in general improve all CNN-based image classification methods.

AUTHORS	DATA USED	TECHNIQUES USED	OBSERVATIONS
Y H SharathKumar ,C D Divya (2014)	Kaggle	Sequential Forward Selection, Sequential Floating Forward Selection.	87%
Shahram Taheri1, OnsenToygar1(2018)	UCI and ICMR-INDIAB	CNN	75%
TiborTrnovszky, PatrikKamencay,al., (2017)	UCI, ICMR-INDIAB	Component Analysis (PCA), Linear Discriminant Analysis (LDA),	78%
Hung Nguyen, Sarah J. Maclagan (2012)	Kaggle, UCI, Data set	Local Binary Patterns Histograms(LBPH) , Support Vector Machine (SVM)	76%
Parinita Badre, Siddhant Bandiwadekar (2015)	Kaggle	AK-mode algorithm, K-mode algorithm, ROCK algorithm, MULIC	81%
Shenghua Zhong, Yangliu(2011)	UCI, ICMR-IN	J48, Naïve bayes	75%
Jyotsna Bankar, Nitin Gavai(2015)	Titanic Dataset, K-Nearest Neighbor	K-Nearest Neighbor,Naïve Bayesian classifier	83%

Guobin Chen, RolandKays(2015)	UCI data set	Gaussian Naive Bayes, KNN, SVM and Decision Tree	65%
SimHiewMoi, Hishammuddin Asmuni.(2014).	Benchmark	Binary Logistic Regression, Multilayer Perceptron.	79%
Heydar Maboudi Afkham, Alireza Tavakoli Targhi (2008)	PID, UCI	MLP, NB, RF, FCV, PS	66%
AdityaKhosla, Michael Bernstein (2014)	UCI	ANN, SVM	69%
P. N. Druzhkov, V. D. Kustikova (2015)	Bench mark and Titanic Dataset	Genetic algorithm, ANN	87%
IlyaSutskever, Geoffrey E.Hinton (2017)	PID	Naïve Bayes, clustering and decision tree	71%
Simonyan, K., Zisserman, A.(2015)	Kaggle, UCI, Data set	CNN,ANN	92%
He, K., Zhang, X., Ren, S., (2014).	UCI, ICMR-INDIAB	FCV, PS	82%

Figure 2.1 : Overview Of Literature Review

METHODOLOGY

III.METHODOLOGY

It will cover the details explanation of methodology that is being used to make this project complete and working well. Many methodology or findings from this field mainly generated into journal for others to take advantages and improve as upcoming studies. The method is use to achieve the objective of the project that will accomplish a perfect result. In order to evaluate this project, the methodology based on System Development Life Cycle (SDLC), generally three major step, which is planning, implementing and analysis.

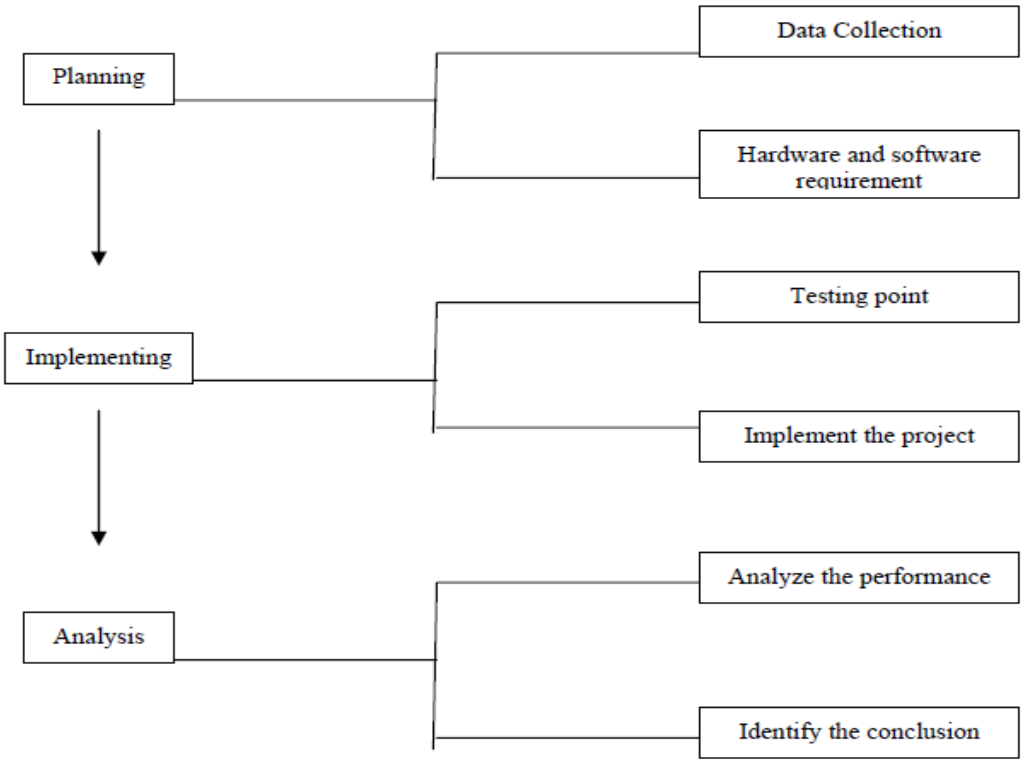


Figure 3.1: Steps Of Methodology

3.1 PLANNING

To identify all the information and requirement such as hardware and software, planning must be done in the proper manner. The planning phase has two main elements namely data collection and the requirements of hardware and software.

3.1.1 DATA COLLECTION

Machine learning needs two things to work, data (lots of it) and models. When acquiring the data, be sure to have enough features (aspect of data that can help for a prediction, like the surface of the house to predict its price) populated to train correctly your learning model. In general, the more data you have the better so make to come with enough rows.

The primary data collected from the online sources remains in the raw form of statements, digits and qualitative terms. The raw data contains error, omissions and inconsistencies. It requires corrections after careful scrutinizing the completed questionnaires. The following steps are involved in the processing of primary data. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses.

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. This technique is performed before the execution of Iterative Analysis. It includes -

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Data Preprocessing is necessary because of the presence of unformatted real-world data. Mostly real-world data is composed of -

- **Inaccurate data (missing data) -** There are many reasons for missing data such as data is not continuously collected, a mistake in data entry, technical problems with biometrics and much more.
- **The presence of noisy data (erroneous data and outliers) -** The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more.

- **Inconsistent data -** The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e., violation of data constraints and much more.

3.2 IMPLEMENTATION

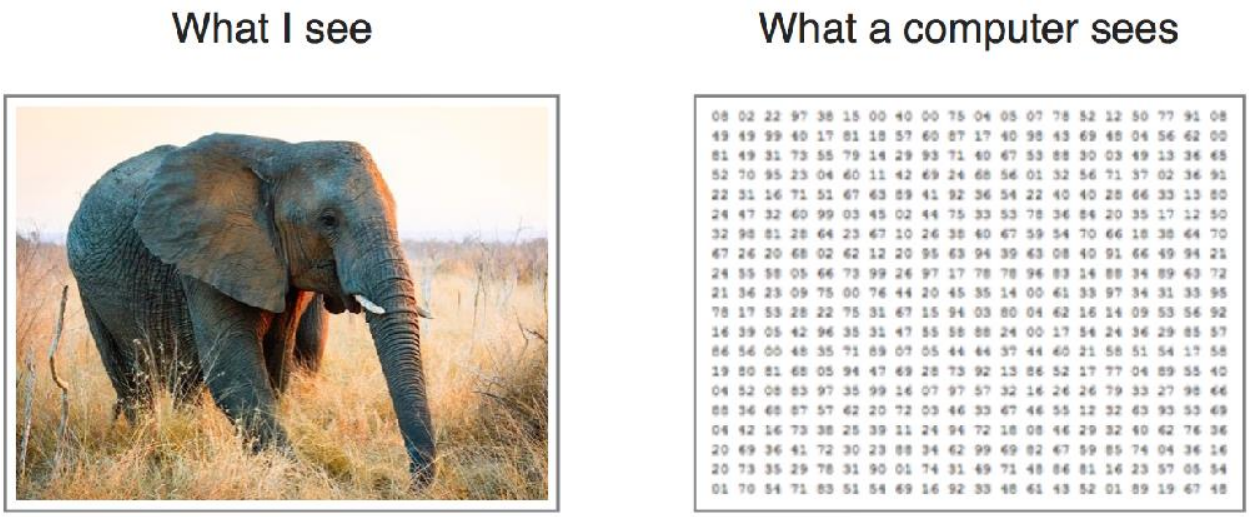
In this work, a business intelligent model has been developed, to classify different animals, based on a specific business structure deal with Animal classification using a suitable machine learning technique. The model was evaluated by a scientific approach to measure accuracy. We are using Convolutional Neural Network (CNN) to build our model.

3.2.1 ALGORITHM DESCRIPTION

Convolutional Neural Network

A convolutional neural network (CNN) is a special architecture of artificial neural networks, proposed by YannLeCun in 1988. CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification. For example Facebook uses CNN for automatic tagging algorithms, Amazon—for generating product recommendations and Google for search through among users' photos.

Let us consider the use of CNN for image classification in more detail. The main task of image classification is acceptance of the input image and the following definition of its class. This is a skill that people learn from their birth and are able to easily determine that the image in the picture is an elephant. But the computer sees the pictures quite differently:



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.

To solve this problem the computer looks for the characteristics of the base level. In human understanding such characteristics are for example the trunk or large ears. For the computer, these characteristics are boundaries or curvatures. And then through the groups of convolutional layers the computer constructs more abstract concepts.

In more detail: the image is passed through a series of convolutional, nonlinear, pooling layers and fully connected layers, and then generates the output.

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 an input matrix.
This operation, from a human perspective, is analogous to identifying boundaries and simple colors on the image. But in order to recognize the properties of a higher level such as the trunk or large ears the whole network is needed.

The network will consist of several convolutional networks mixed with nonlinear and pooling layers. When the image passes through one convolution layer, the output of the first layer becomes the input for the second layer. And this happens with every further convolutional layer.

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 (as a class label).

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

This means that if some features (as for example boundaries) have already been identified in the previous convolution operation, than a detailed image is no longer needed for further processing, and it is compressed to less detailed pictures.

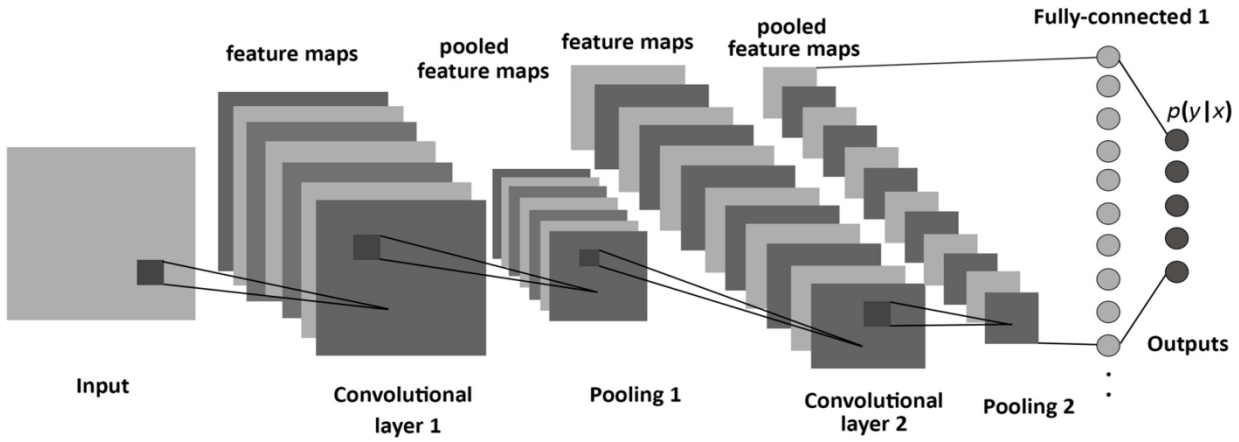


Figure 3. 2: Compressed To Less Detailed Pictures

After completion of series of convolutional, nonlinear and pooling layers, it is necessary to attach a fully connected layer. This layer takes the output information from convolutional networks. Attaching a fully connected layer to the end of the network results in an N dimensional vector, where N is the amount of classes from which the model selects the desired class.

3.3 ANALYSIS

In this final phase, i test our classification model on our prepared image dataset and also measure the performance on our dataset. To evaluate the performance of our created classification and make it comparable to current approaches, we use accuracy to measure the effectiveness of classifiers.

After model building, knowing the power of model prediction on a new instance, is very important issue. Once a predictive model is developed using the historical data, one would be curious as to how the model will perform on the data that it has not seen during the model building process. One might even try multiple model types for the same prediction problem, and then, would like to know which model is the one to use for the real-world decision making situation, simply by comparing them on their prediction performance (e.g., accuracy). To measure the performance of a predictor, there are commonly used performance metrics, such as accuracy, recall etc. First, the most commonly used performance metrics will be

described, and then some famous estimation methodologies are explained and compared to each other. "Performance Metrics for Predictive Modeling In classification problems, the primary source of performance measurements is a coincidence matrix (**classification matrix or a contingency table**)". Above figure shows a coincidence matrix for a two-class classification problem. The equations of the most commonly used metrics that can be calculated from the coincidence matrix.

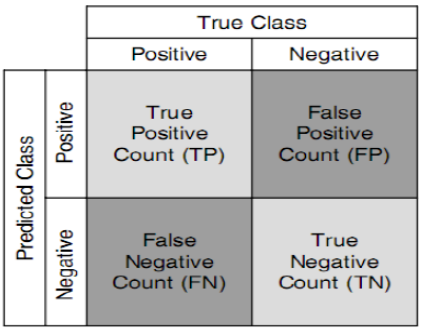


Figure 3.3: Confusion Matrix And Formulae

As being seen in above figure, the numbers along the diagonal from upper-left to lower-right represent the correct decisions made, and the numbers outside this diagonal represent the errors. "The true positive rate (also called hit rate or recall) of a classifier is estimated by dividing the correctly classified positives (the true positive count) by the total positive count. The false positive rate (also called a false alarm rate) of the classifier is estimated by dividing the incorrectly classified negatives (the false negative count) by the total negatives. The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

CNNs have wide applications in image and video recognition, recommender systems and natural language processing. In this article, the example that I will take is related to Computer Vision. However, the basic concept remains the same and can be applied to any other use-case!

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that we developed for neural networks still apply on CNNs. Pretty straightforward, right?

A CNN sequence to classify handwritten digit:

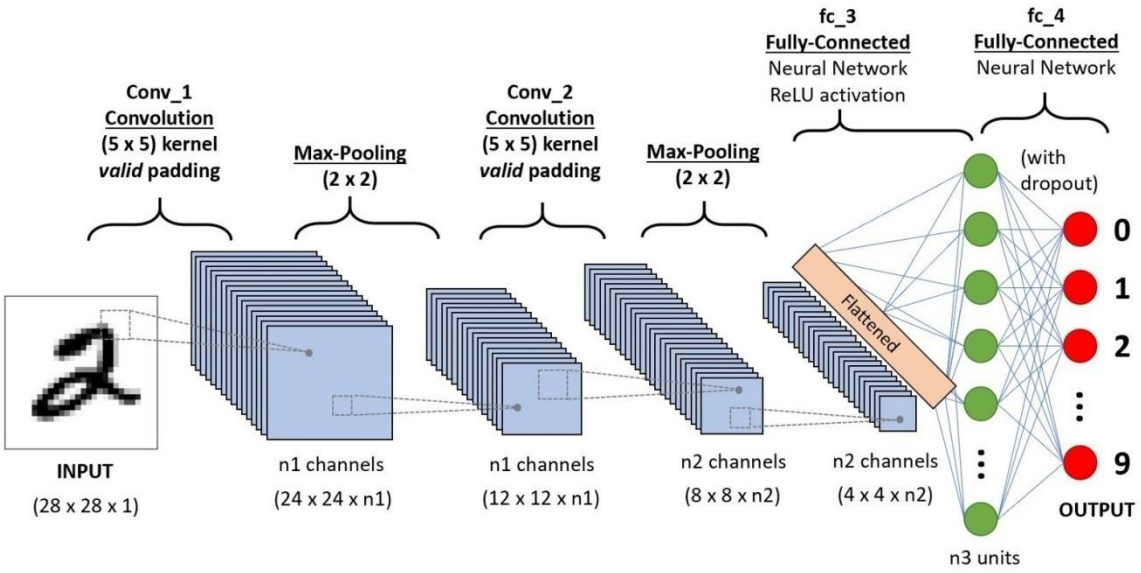


Figure 3.4: CNN Sequence

CNNs operate over Volumes:

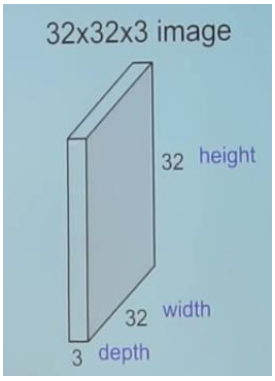


Figure 3. 5 : CNNs Operate Over Volumes

Unlike neural networks, where the input is a vector, here the input is a multi-channeled image (3 channeled in this case).

Example of a trained network:

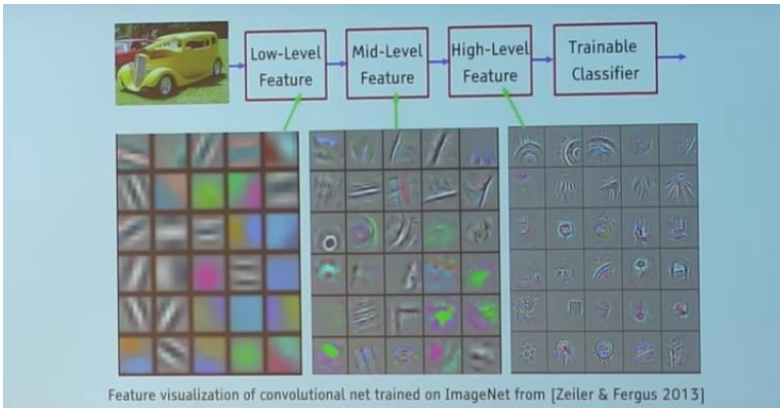


Figure 3. 6 : Trained Network

Take a look at the filters in the very first layer (these are our 5*5*3 filters). Through back propagation, they have tuned themselves to become blobs of colored pieces and edges. As we go deeper to other convolution layers, the filters are doing dot products to the input of the previous convolution layers. So, they are taking the smaller colored pieces or edges and making larger pieces out of them.

Take a look at image 4 and imagine the 28*28*1 grid as a grid of 28*28 neurons. For a particular feature map (the output received on convolving the image with a particular filter is called a feature map), each neuron is connected only to a small chunk of the input image and all the neurons have the same connection weights. So again coming back to the differences between CNN and a neural network.

There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future. Some of them have been listed below:

LeNet

AlexNet

VGGNet

GoogLeNet

ResNet

ZFNet

Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolution networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.
- Keras is compatible with: **Python 2.7-3.6**.

Summarize the construction of deep learning models in Keras as follows:

1. **Define your model**. Create a sequence and add layers.
2. **Compile your model**. Specify loss functions and optimizers.
3. **Fit your model**. Execute the model using data.
4. **Make predictions**. Use the model to generate predictions on new data.

3.4 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Installation:

Windows, Linux and macOS distributions have matplotlib and most of its dependencies as wheel packages. Run the following command to install matplotlib package :

```
python -mpip install -U matplotlib
```

Importing matplotlib:

```
from matplotlib import pyplot as plt
or
import matplotlib.pyplot as plt
```

Basic plots in Matplotlib :

Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information.

Uses of matplotlib

- Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-**oriented** API for **embedding**plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.
- MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.
- Matplotlib is a Python package for 2D plotting that generates production-quality graphs. It supports interactive and non-interactive plotting, and can save images in several output formats (PNG, PS, and others). It can use multiple window toolkits (GTK+, wxWidgets, Qt, and so on) and it provides a wide variety of plot types (lines, bars, pie charts, histograms, and many more).

Merits of Matplotlib

- The idea behind Matplotlib can be summed up in the following motto as quoted by John Hunter, the creator and project leader of Matplotlib:
- “Matplotlib tries to make easy things easy and hard things possible”.
- Matplotlib was born in the scientific area of computing, where gnuplot and MATLAB were (and still are) used a lot.
- Matplotlib was modeled on MATLAB, because graphing was something that MATLAB did very well. The high degree of compatibility between them made many people move from MATLAB to Matplotlib, as they felt like home while working with Matplotlib.
- But what are the points that built the success of Matplotlib? Let's look at some of them:
- **It uses Python:** Python is a very interesting language for scientific purposes (it's interpreted, high-level, easy to learn, easily extensible, and has a powerful standard library) and is now used by major institutions such as NASA, JPL, Google, DreamWorks, Disney, and many more.
- **It's open source, so no license to pay:** This makes it very appealing for professors and students, who often have a low budget.
- **It's a real programming language:** The MATLAB language (while being Turing-complete) lacks many of the features of a general-purpose language like Python.
- **It's much more complete:** Python has a lot of external modules that will help us perform all the functions we need to. So it's the perfect tool to acquire data, elaborate the data, and then plot the data.
- **It's very customizable and extensible:** Matplotlib can fit every use case because it has a lot of graph types, features, and configuration options.
- **It's integrated with LaTeX markup:** This is really useful when writing scientific papers.
- **It's cross-platform and portable:** Matplotlib can run on Linux, Windows, Mac OS X, and Sun Solaris (and Python can run on almost every architecture available).
- The aim of Matplotlib is to generate graphs. So, we need a way to actually view these images or even to save them to files. We're going to look at the various output formats available in Matplotlib and the graphical user interfaces (GUIs) supported by the library.

- Matplotlib supports both the categories, particularly with the following output format.
- Matplotlib supports both the categories, particularly with the following output formats:

Format	Type Description	Description
EPS	Vector	Encapsulated PostScript
JPG	Raster	Graphic format with lossy compression method for photographic output.
PDF	Vector	Portable Document Format (PDF).
PNG	Raster	Portable Network Graphics (PNG), a raster graphics format with a lossless compression method (more adaptable to line art than JPG).
PS	Vector	Language widely used in publishing and as printers jobs format.
SVG	Vector	Scalable Vector Graphics (SVG), XML based.

Figure 3.4.1: Matplotlib Table

3.4.1 BACKEND

- A backend that displays the image on screen is called a user interface backend.
- The backend is that part of Matplotlib that works behind the scenes and allows the software to target several different output formats and GUI libraries (for screen visualization).

- In order to be even more flexible, Matplotlib introduces the following two layers structured (only for GUI output):

  The renderer: This actually does the drawing

  The canvas: This is the destination of the figure.

- The standard renderer is the Anti-Grain Geometry (AGG) library, a high performance rendering engine which is able to create images of publication level quality, with anti-aliasing, and sub pixel accuracy. AGG is responsible for the beautiful appearance of Matplotlib graphs.

- The canvas is provided with the GUI libraries, and any of them can use the AGG rendering, along with the support for other rendering engines (for example, GTK+).

- Let's have a look at the user interface toolkits and their available renderers:

Backend	**Description**
GTKAgg	GTK+ (The GIMP ToolKit GUI library) canvas with AGG rendering.
GTK	GTK+ canvas with GDK rendering. GDK rendering is rather primitive, and doesn't include anti-aliasing for the smoothing of lines.
GTKCairo	GTK+ canvas with Cairo rendering.
WxAgg	wxWidgets (cross-platform GUI and tools library for GTK+, Windows, and Mac OS X. It uses native widgets for each operating system, so applications will have the look and feel that users expect on that operating system) canvas with AGG rendering.
WX	wxWidgets canvas with native wxWidgets rendering.
TkAgg	TkAgg Tk (graphical user interface for Tcl and many other dynamic languages) canvas with AGG rendering.

Figure 3.4.2 Backend Table

RESULTS AND DISCUSSION

IV.RESULTS AND DISCUSSION

4.1 SCREEN SHOT

Command prompt

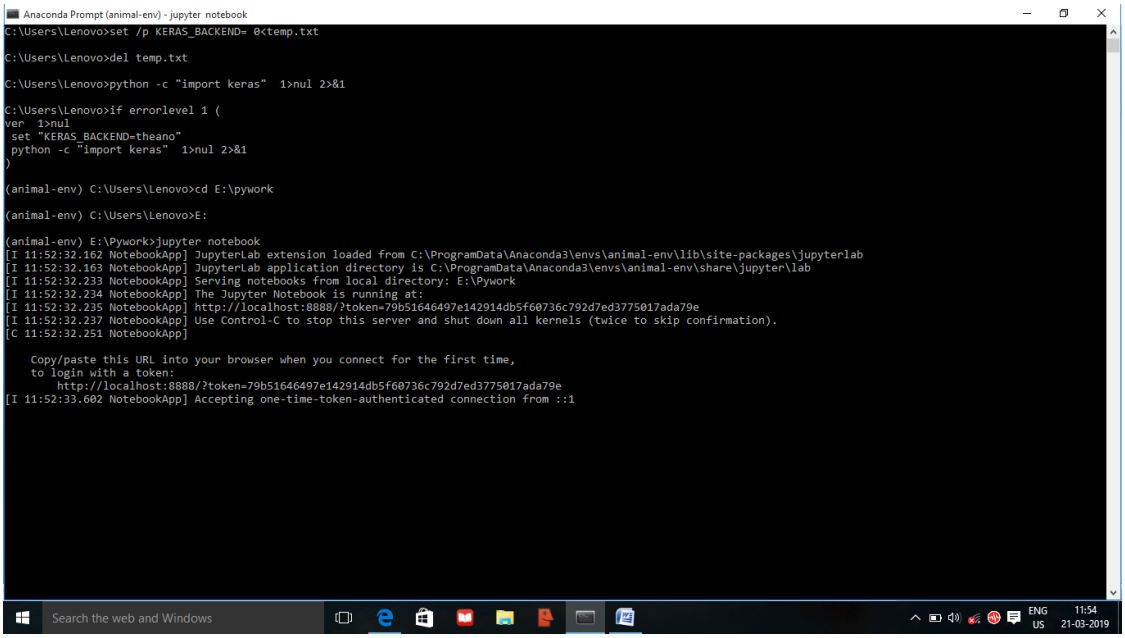


Figure 4.1.1 Run Page

Home page

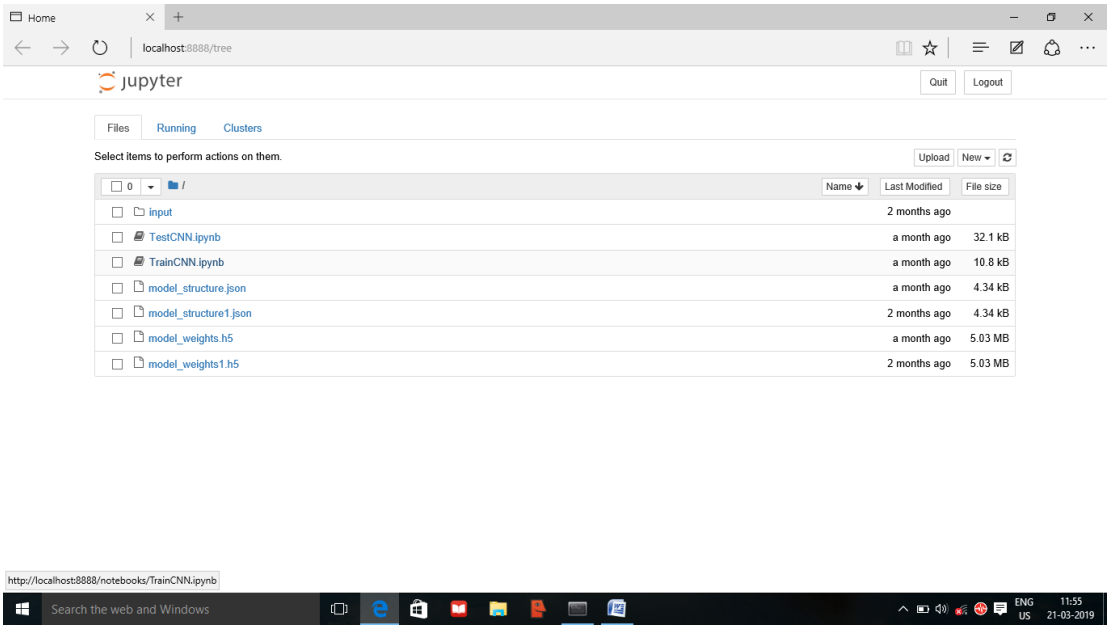


Figure 4.1.2 Home Page

Test page

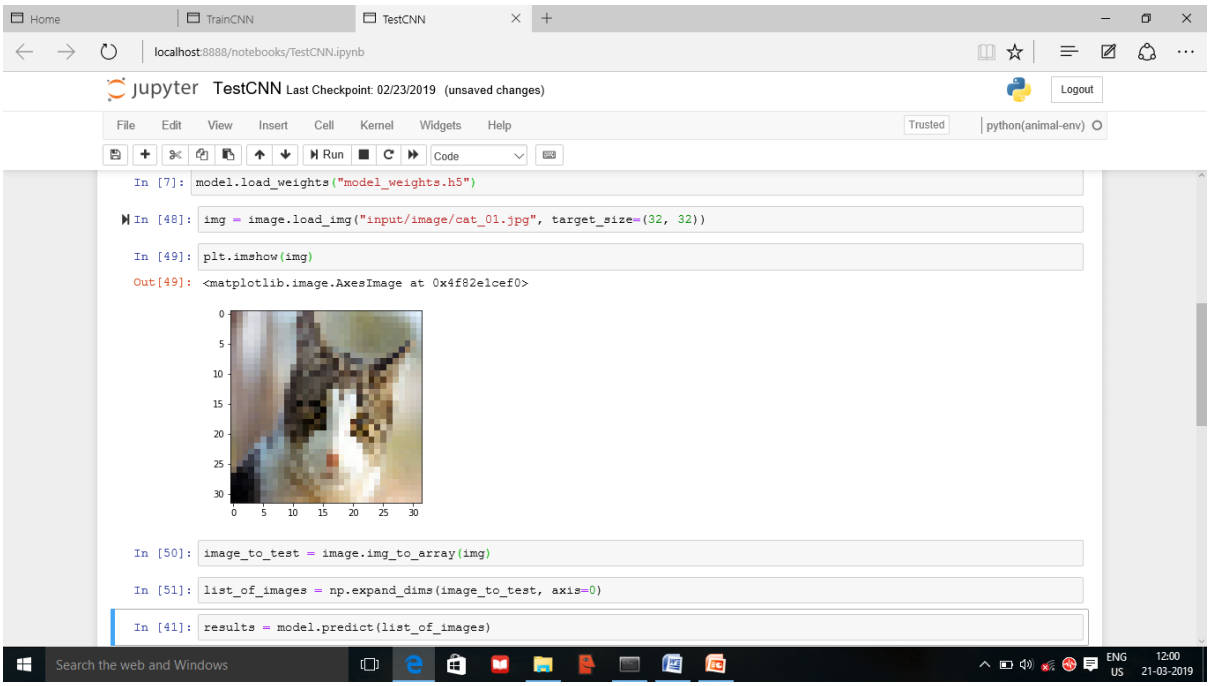


Figure 4.1.3 Testing Page

Output

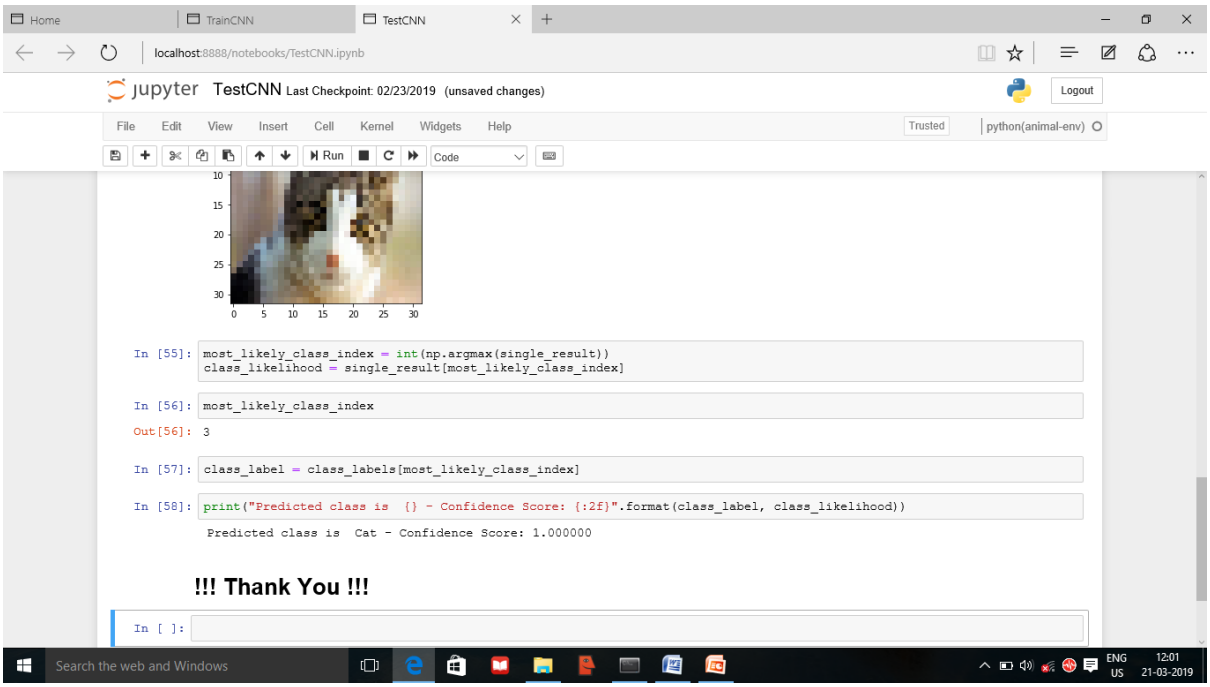


Figure 4.1.4 Output Page

Test page

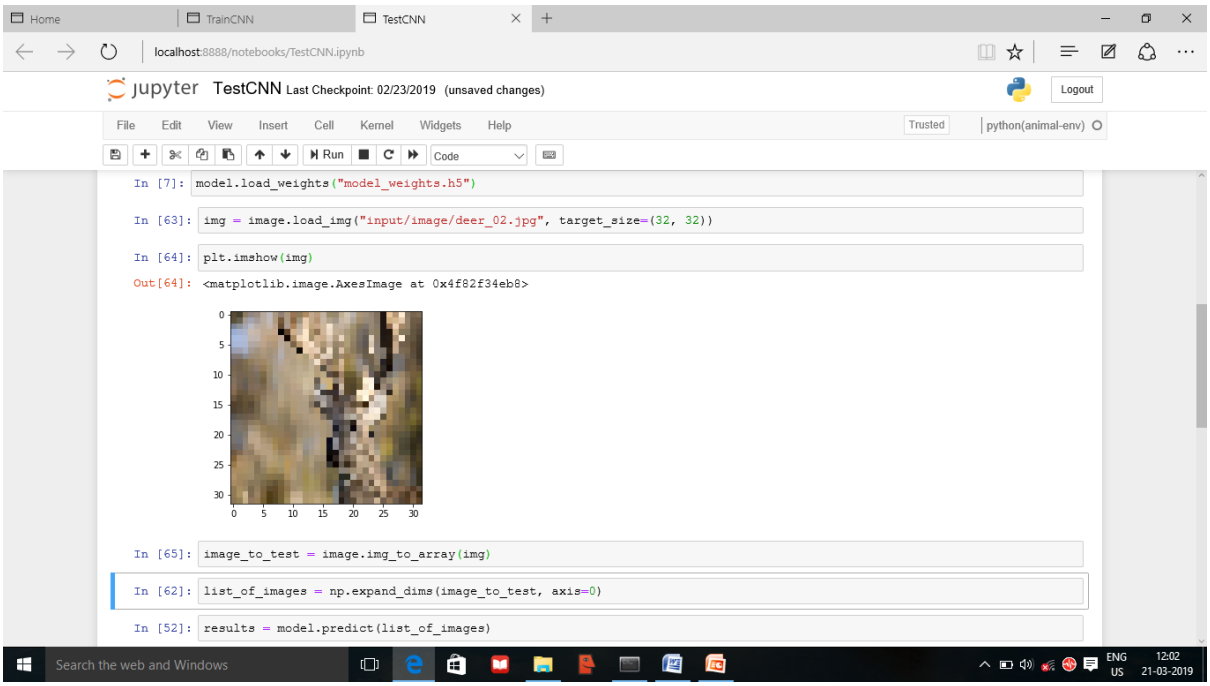


Figure 4.1.5 Testing Page

Output

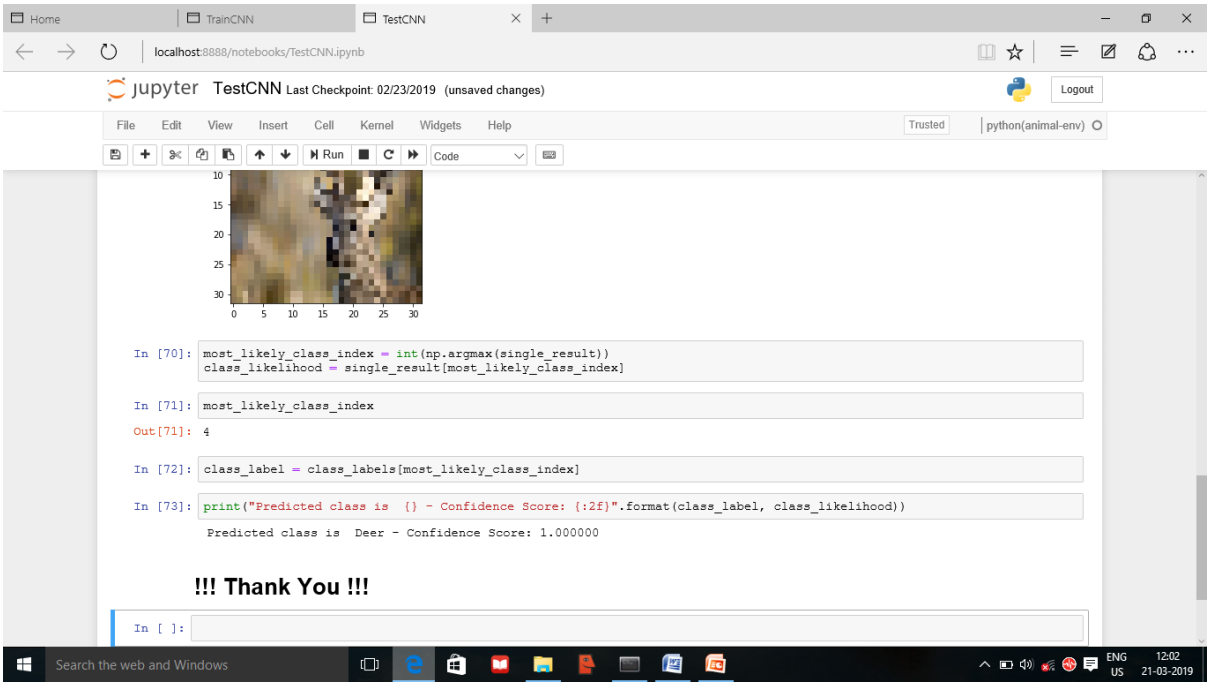


Figure 4.1.6 Output Page

Test page

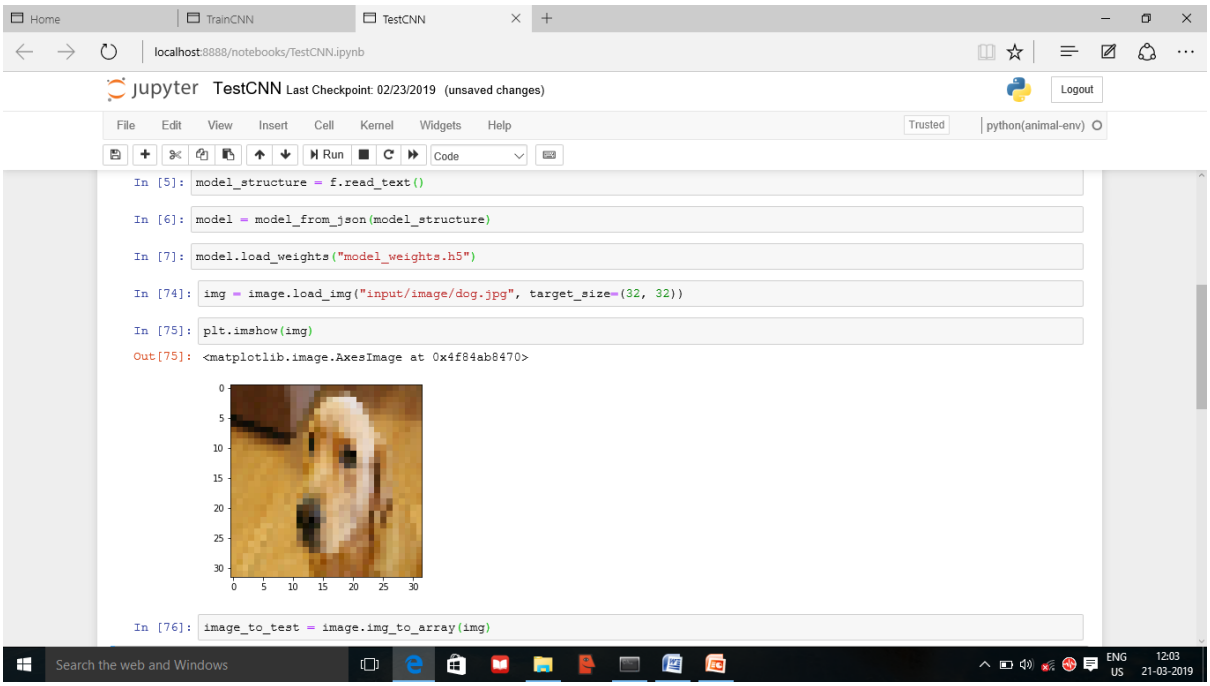


Figure 4.1.7 Testing Page

Output

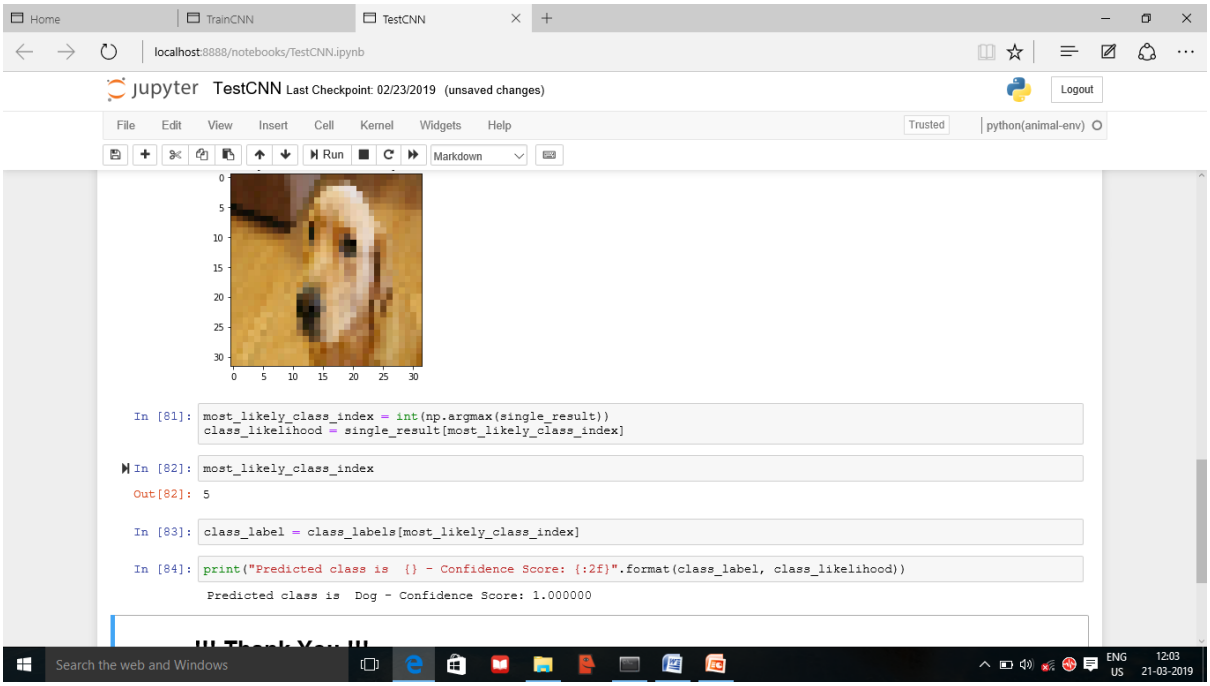


Figure 4.1.8 Output Page

Test page

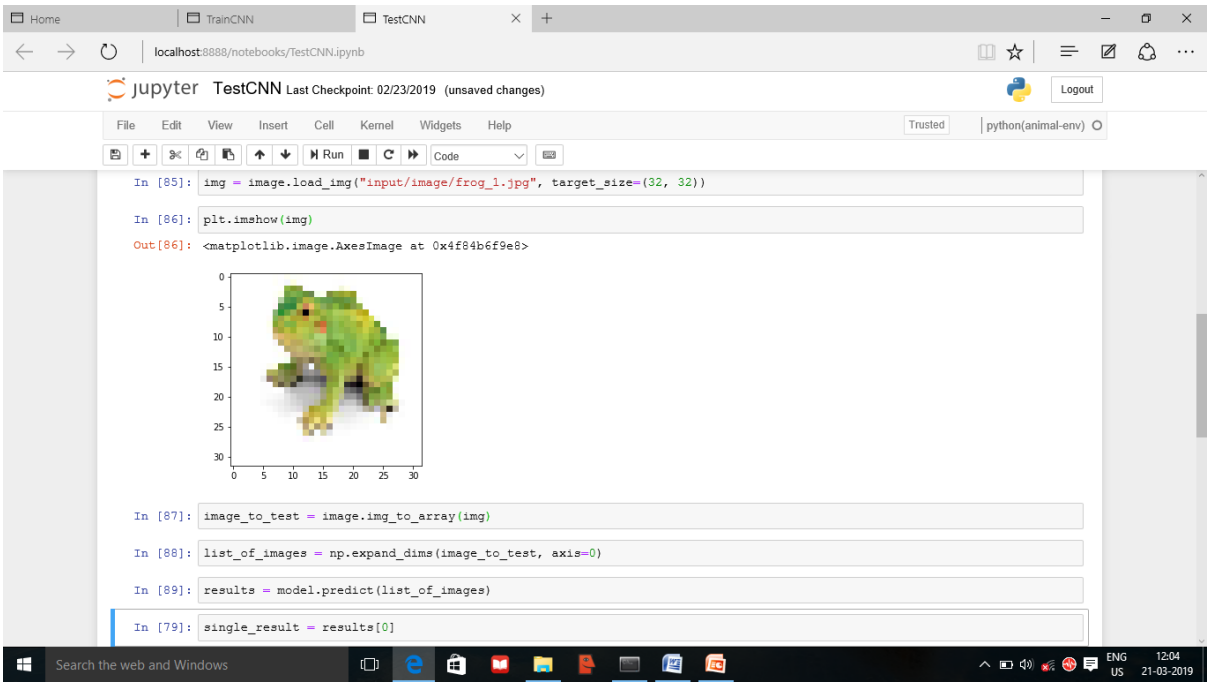


Figure 4.1.9 Testing Page

Output

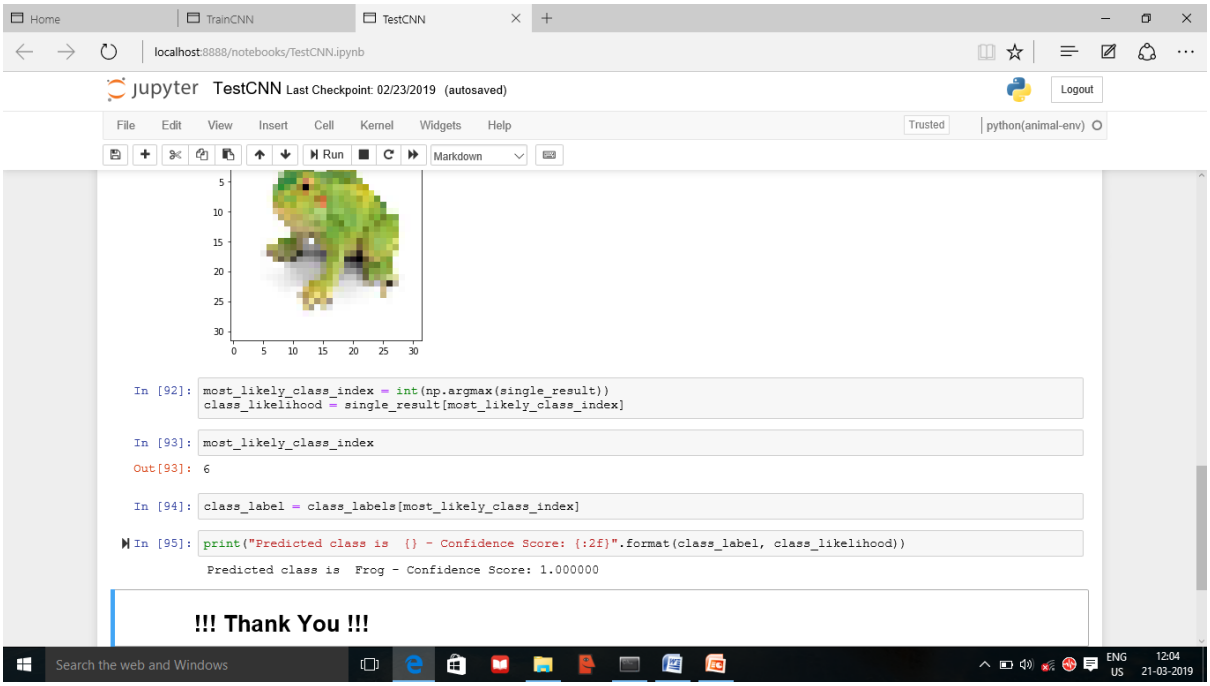


Figure 4.1.10 Output Page

Test page

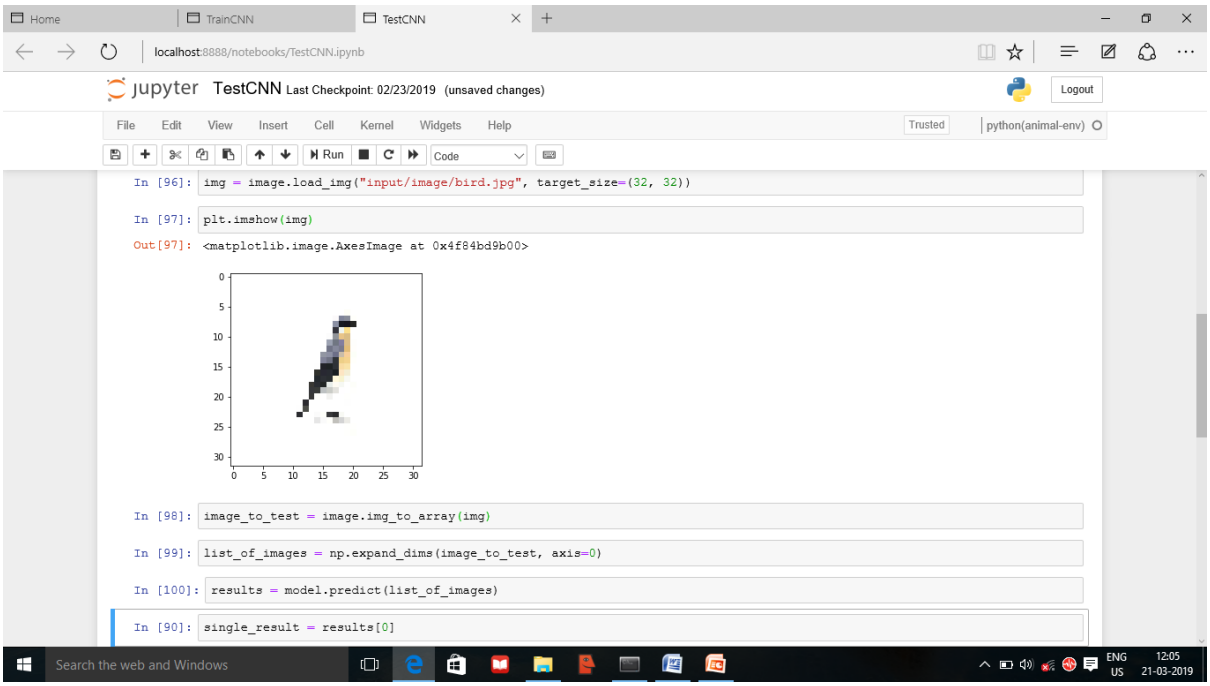


Figure 4.1.11 Testing Page

Output

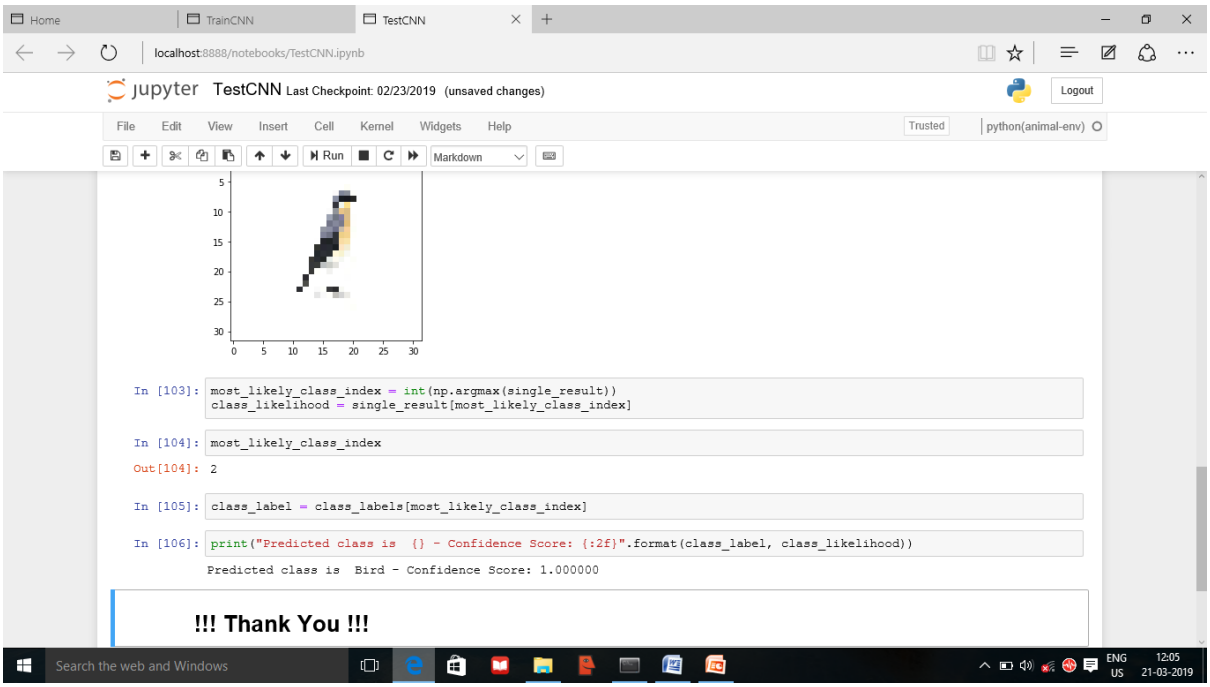


Figure 4.1.12 Output Page

Test page

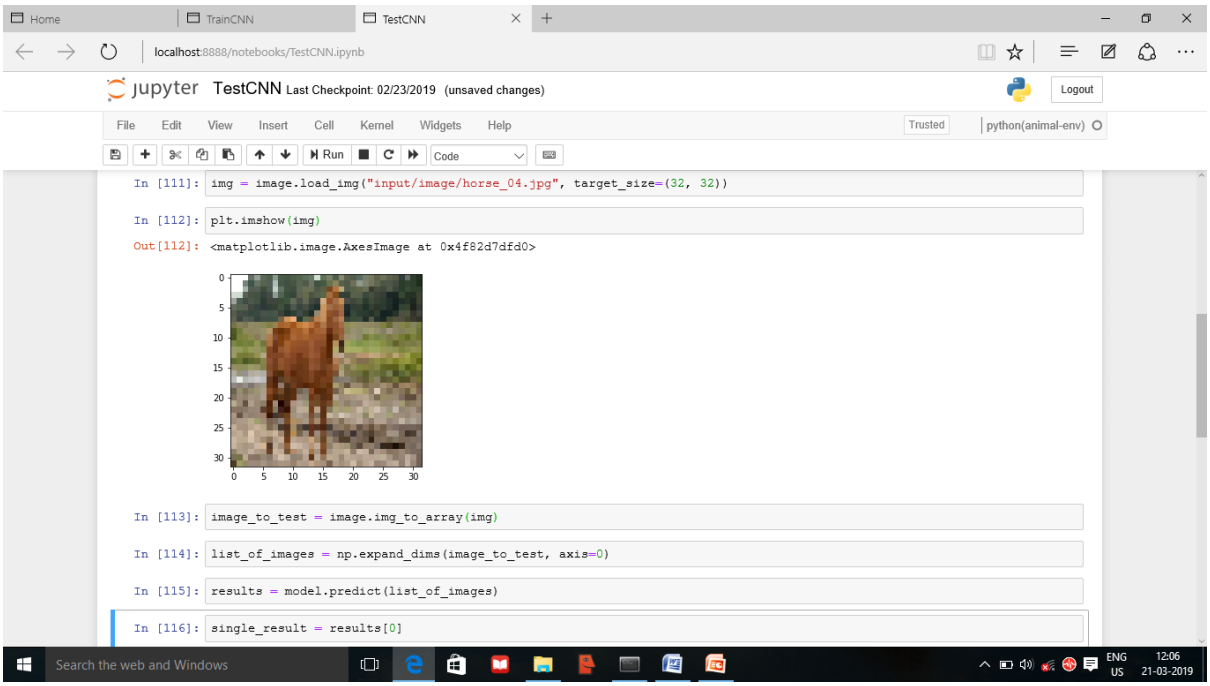


Figure 4.1.13 Testing Page

Output

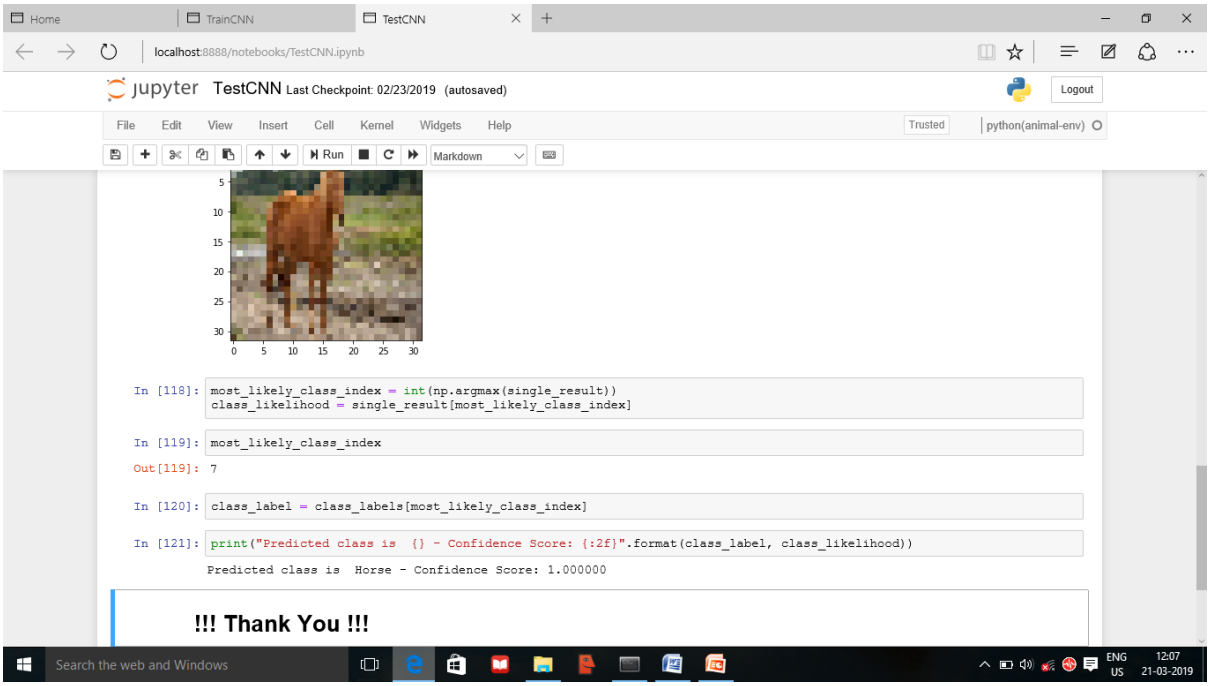


Figure 4.1.14 Output Page

CONCLUSION

V.CONCLUSION

This system comes under deep learning which is advanced technique at present. Animal recognition and classification is an important area which has not been discussed rapidly outperform the state-of-the-art systems for animal face classification. Animal classification which relies on the problem of distinguishing images of different animal species is an easy task for humans, but evidence suggests that even in simple cases like cats and dogs, it is difficult to distinguish them automatically. Conclude the experimental result are getting from developed system 70% Accurate.

SCOPE FOR FUTURENHANCEMENT

VI.SCOPE FOR FUTURE ENHANCEMENT

The present system is developed with ten classifications using convolutional neural network (CNN) a deep learning concept in python. In future we can able to train more classification not only for animal it can able to classify other objects like leaf recognition, face recognition, fire recognition, moving image. This independence from prior knowledge and human effort in feature design is a major advantage in image classification. Technical advances reduced the error rate of deep convolutional neural nets by a further factor of three so that they are now quite close to human performance for static images.

REFERNCES

VII.REFERNCES

Y.H.SharathKumar,C.D.Divya (2014). Feature selection approach in animal classification. *An International Journal* (SIPIJ) Vol.5, No.4, August 2014.

Shahram Taheri1, OnsenToygar1(2018).Animal classification using facial images with score-level fusion. *Published as a Journal paper at* April 2018.

TiborTrnovszky, PatrikKamencay(2017).Animal Recognition System Basedon Convolutional Neural Network.*Journal of Sensors*Volume 2018.

Hung Nguyen, Sarah J. Maclagan(2012).Animal Recognition and Identificationwith Deep Convolutional Neural Networks.*Research paper at* Jun 10.15598/aeee.v15i3.2202.

ParinitaBadre, SiddhantBandiwadekar (2015).Automatically Identifying Animals Using Deep Learning.International *Journal of Computer Vision* November 17, 2015.

Sheng-huaZhong ,Yangliu, (2011).Image classification is a well-known classical problem in multimedia content analysis.*Published as a Journal paper at* 30 Jan 2011 v2.

JyotsnaBankar, NitinGavai(2015). Animal Classification Using Machine-Learning Algorithm.*Published as a Journal paper at* Jul 15, 2015 Volume 26.

GuobinChen,RolandKays(2014).Deep convolutional neural network based species recognition for wild animal monitoring.2014 IEEE International Conference on Image Processing (ICIP).

SimHiewMoi, Hishammuddin Asmuni.(2014)Multimodal biometrics: Weighted score level fusion based on non-idealiris and face images.*Published as a Journal paper at* 14 March 2014.

HeydarMaboudiAfkham, AlirezaTavakoliTarghi (2008).Joint Visual Vocabulary For Animal Classification. *International Conference* on Pattern Recognition (ICPR), 2008.

AdityaKhosla, Michael Bernstein (2015).ImageNet Large Scale Visual Recognition Challenge. *International Journal* of Computer Vision September 2014(Submitted on 1 Sep 2014 (v1), last revised 30 Jan 2015 v3).

P. N. Druzhkov, V. D. Kustikova (2015).A Survey of Deep Learning Methods and Software Tools for Image Classification and Object Detection.Published as *a Journal paper* at January 2016, Volume 26.

IlyaSutskever, Geoffrey E.Hinton (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Published as a Journal paper* at May 24, 2017 - Volume 60 Issue 6, June 2017.

Simonyan, K., Zisserman, A.(2015).Very deep convolutional networks for large scale image recognition. *Published as a conference paper* at ICLR 2015.

He,K.,Zhang,X., Ren,S.(2014) Pooling in deep convolutional networks for visual recognition. *Published as a Journal paper at* CoRR, abs/1406.4729v2, 2014.