**IMAGE CLASSIFICATION USING DEEP STRUCTURED NEURAL NETWORK**

**A PROJECT REPORT**

*Submitted by*

**FATHIMA PARVEEN.S (810015104025)**

**PANDIMEENA.S (810015104056)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**UNIVERSITY COLLEGE OF ENGINEERING – BIT CAMPUS,**

**TIRUCHIRAPPALLI**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2019**

**UNIVERSITY COLLEGE OF ENGINEERING,**

**BIT CAMPUS,**

**TIRUCHIRAPPALLI-620 024**

**BONAFIDE CERTIFICATE**

Certified that this project report “**Image Classification Using Deep Structured Neural Network**” is the bonafide work of **“Ms. S.FATHIMA PARVEEN (810015104025)** and **Ms**. **S.PANDIMEENA (810015104056)”** who carried out the project work under my supervision.

**Dr.D.VENKATESAN Mrs.R.S.RAMPRIYA**

HEAD OF THE DEPARTMENTSUPERVISOR

Assistant Professor Teaching Fellow

Department of CSE Department of CSE

Anna University, BIT Campus Anna University, BIT Campus

University College of Engineering University College of Engineering

Tiruchirappalli - 620 024 Tiruchirappalli - 620 024

Submitted for the project viva voce examination held on ………………

**Internal Examiner External Examiner**

**DECLARATION**

We hereby declare that the work entitled **“IMAGE CLASSIFICATION USING DEEP STRUCTURED NEURAL NETWORK”** is submitted in partial fulfillment of the requirements for the award of the degree in B.E.-Computer Science and Engineering, University College of Engineering, BIT Campus, Tiruchirappalli, is a record of our own work carried out by us during the academic year 2018-2019 under the guidance of **Mrs. R.S.RAMPRIYA,** Teaching Fellow, Department of Computer Science and Engineering, University College of Engineering, BIT Campus, Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other Degree, either in this or any other University.

**SIGNATURE OF THE CANDIDATES**

**FATHIMA PARVEEN S**

**PANDIMEENA S**

I certify that the declaration made above by the candidate is true.

Signature of the Project Guide

**Mrs. R. S. RAMPRIYA**

Teaching Fellow

Department of Computer Science and Engineering

University College of Engineering (BIT Campus)

Tiruchirappalli – 620 024

**ACKNOWLEDGEMENT**

First and foremost of all, we would like to thank our beloved Parents and God Almighty for giving us the strength, knowledge, ability and opportunity to undertake this project study and to persevere and complete it satisfactorily.

We would like to convey my heartfelt thanks to our honorable Dean **Dr. T. SENTHILKUMAR,** Associate Professor for having provided me with all required facilities to complete my project without hurdles.

We would like to express my sincere thanks and deep sense of gratitude to guide **Dr. D. VENKATESAN,** Assistant Professor and Head, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

We would like to thank my project guide **Mrs. R. S. RAMPRIYA,** Teaching Fellow, Department of Computer Science and Engineering, for his valuable guidance throughout the phase of the project. It is our responsibility to thank our project coordinator **Mr. C. SANKAR RAM,** Assistant Professor, Department of Computer science and Engineering for his constant inspiration that he has all through the project period.

We would like to thank **Mr. C. SURESH KUMAR,** Teaching Fellow, Department of Computer Science and Engineering, for his encouragement for this work.

We extend my thanks to all other teaching and non-teaching staffs for their encouragement and support.

And we would like to thank all our well-wishers who have encouraged us to do this project.

**ABSTRACT**

This project has an approach for image classification model using a deep structured neural network. A binary image dataset has been considered for the classification. Conventional back propagation neural network has an input layer, hidden layer, and an output layer but convolutional neural network, has a convolutional layer, and a max pooling layer. We train this proposed classifier to calculate the decision boundary of the image dataset. The data in the real world is mostly in the form of unlabeled and unstructured format. Useful information cannot be easily derived from neural networks which are shallow i.e. the ones which have less number of hidden layers. We propose deep neural network based CNN classifier which has a large number of hidden layers and can derive meaningful information from images. Image classification is implemented using Anaconda Prompt with Tensor flow. Tensor Flow is a popular open source library for machine learning and deep neural networks. It has its application in Medial image analysis, drug design and computer vision.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| CHAPTER. NO | TITLE | PAGE NO |
|  | **ABSTRACT** | i |
|  | **LIST OF FIGURES** | iv |
|  | **LIST OF ABBREVIATIONS** | v |
| 1 | **INTRODUCTION** |  |
|  | 1.1 Image classification | 1 |
|  | 1.2 Deep learning | 2 |
|  | 1.3 Neural networks | 3 |
|  | 1.3.1 Artificial neural network | 4 |
|  | 1.3.2 Deep structured neural network | 5 |
| 2 | **LITERATURE SURVEY** | 7 |
| 3 | **SYSTEM ANALYSIS** |  |
|  | 3.1 Existing system | 12 |
|  | 3.1.1 Naïve Bayes theorem | 12 |
|  | 3.1.2 K-means Algorithm | 14 |
|  | 3.1.3 Support Vector Machine | 15 |
|  | 3.1.4 Artificial neural network | 16 |
|  | 3.2 Proposed system | 18 |
| 4 | **SYSTEM SPECIFICATION** |  |
|  | 4.1 Hardware requirements | 19 |
|  | 4.2 Software requirements | 19 |
|  | 4.3 About the software | 19 |
| 5 | **SYSTEM DESIGN** |  |
|  | 5.1 System Architecture | 23 |
|  | 5.2 System modules | 24 |
| 6 | **DATASETS USED** | 31 |
| 7 | **SYSTEM IMPLEMENTATION** |  |
|  | 7.1 Input Extraction | 34 |
|  | 7.2 Convolution+ ReLU Algorithm | 35 |
|  | 7.3 Max pooling Algorithm | 36 |
|  | 7.4 Fully Connected Algorithm | 37 |
| 8 | **CONCLUSION AND FUTURE WORK** | 38 |
|  | **APPENDIX** | 39 |
|  | **REFERENCES** | 49 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1 | Support vector machine | 15 |
| 2 | Artificial neural network | 17 |
| 3 | Architecture of Deep structured neural network | 23 |
| 4 | Convolution layer | 25 |
| 5 | ReLU | 26 |
| 6 | Pooling | 28 |
| 7 | Flatten | 29 |
| 8 | Sample Dog images | 33 |
| 9 | Sample Cat images | 33 |
| 10 | Input | 34 |
| 11 | Convolution + ReLU operation | 35 |
| 12 | Pooling process | 36 |
| 13 | Extraction of dataset | 43 |
| 14 | Dataset | 44 |
| 15 | Image visualisation | 45 |
| 16 | Test image visualisation | 46 |
| 17 | Convolution Image | 46 |
| 18 | Conv2D Image | 47 |
| 19 | Image after ReLU | 47 |
| 20 | Convoluted Image | 48 |
| 21 | Pooled Image | 48 |
| 22 | Prediction | 49 |

**LIST OF ABBREVIATIONS**

**DL** Deep Learning

**ANN** Artificial Neural Network

**CNN** Convolution Neural Network

**DNN** Deep Neural Network

**GPU** Graphics Processing Units

**CPU** Central Processing Units

**SVM** Support Vector Machine

**VHR** Very High Resolution

**SAR** Synthetic aperture radar

**CT** Computed tomography

**FLIR** Forward-looking infrared

**JSON** JavaScript Object Notation

**RELU** Rectified Linear Unit

**ASR**  Automatic Speech Recognition

**LBP**  Local Binary Pattern

**MCG** Multiscale Combinational Grouping

**VGG** Visual Geometry Group

**CHAPTER 1**

**INTRODUCTION**

**1.1 IMAGE CLASSIFICATION**

Image classification is assigning pixels in the image to categories or classes of interest. Image classification is the process of taking an **input** (like a picture) and outputting a **class** (like “cat”) or a **probability** that the input is a particular class (“there’s a 90% probability that this input is a cat”).

In order to classify a set of data into different classes or categories, the relationship between the data and the classes into which they are classified must be well understood. To achieve this by computer, the computer must be trained. Image classification is a process including image preprocessing, image segmentation, key feature extraction and matching identification.

**Types of Image Classification**

* Unsupervised classification
* Supervised classification

**Unsupervised classification**

Unsupervised classification is where the outcomes (groupings of pixels with common characteristics) are based on the software analysis of an **image** without the user providing sample classes. Training sites (also known as testing sets or input classes) are selected based on the knowledge of the user.

**Supervised classification**

**Supervised classification** is based on the idea that a user can select sample pixels in an image that are representative of specific classes and then direct the image processing software to use these training sites as references for the classification of all other pixels in the image.. Training sites (also known as testing sets or input classes) are selected based on the knowledge of the user. Image classification is a supervised learning problem, it define a set of target classes and train a model to recognize them using labeled data.

Image classification plays an important role in computer vision. With the latest figures image classification techniques, we not only get the picture information faster than before, we apply it to scientific experiments, traffic identification, security, medical equipment, face recognition and other fields.

* 1. **DEEP LEARNING**

Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised. Deep learning is also known as deep structured learning or hierarchical learning.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network, image dataset filtering machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts. Deep learning models are vaguely inspired by information processing and communication patterns in biological nervous systems yet have various differences from the structural and functional properties of biological brains (especially human brains), which make them incompatible with neuroscience evidences.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own example, varying numbers of layers and layer sizes can provide different degrees of abstraction. Most modern deep learning models are based on an artificial neural network, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines.

* 1. **NEURAL NETWORKS**

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes.[[1]](https://en.wikipedia.org/wiki/Neural_network#cite_note-1) Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1. Neural networks have different types of categories like recurrent neural network, Artificial neural network and Deep structured neural network.

**1.3.1 ARTIFICIAL NEURAL NETWORKS**

Artificial neural networks (**ANNs**) are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming.

An ANN is based on a collection of connected units called artificial neurons, (analogous to biological neurons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream.

Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.  Different layers may perform different kinds of transformations on their inputs. Neural networks typically have a few thousand to a few million units and millions of connections.

**1.3.2 DEEP STRUCTURED NEURAL NETWORK**

A deep structured neural network is an artificial neural network (ANN) with multiple layers between the input and output layers. The Deep structured neural network finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a Deep structured neural network that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex Deep structured neural network have many layers, hence the name "deep" networks. The goal is that eventually, the network will be trained to decompose an image into features, identify trends that exist across all samples and classify new images by their similarities without requiring human input.

Deep structured neural network can model complex non-linear relationships. Deep structured neural network architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets. Convolutional deep neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).DNNs are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data.

**CHAPTER 2**

**LITERATURE SURVEY**

In this section the literature survey has been carried out. The main focus is on image classification on neural networks. The literature survey gives a clear idea for proposed system.

**2.1 Multi-class segmentation with relative location prior by s. gould, j. rodgers, d. cohen, g. elidan, and d. koller.**

An artificial neural network is applied for classifying texture data of various natural objects found in FLIR images. Hermite functions are used for texture feature extraction from segmented regions of interest in natural scenes taken as a video sequence [30]. A total of 2680 samples for a total of twelve different classes are used for object recognition. Neural networks are found to be extremely effective in robust classification of data giving an average recognition rate of 91.8%. The results show that Hermite functions provide an effective methodology for texture measurement in FLIR imagery.

**2.2 Face description with local binary patterns: Application to face recognition. ‘Pattern Analysis and Machine Intelligence by Ahonen, T., Hadid, A., & Pietikinen, M..**

In facial image representation based on local binary pattern (LBP) texture features, the face image is divided into several regions from which the LBP feature distributions are extracted and concatenated into an enhanced feature vector to be used as a face descriptor [3]. The performance of this method is assessed in the face recognition problem under different challenges. The texture description of a single region describes the appearance of the region and the combination of all region descriptions encodes the global geometry of the face. The results clearly show that facial images can be seen as a composition of micropatterns such as flat areas, spots, lines, and edges which can be well described by LBP.

**2.3 Decaf: A deep convolutional activation feature for generic visual recognition by Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T.**

The features extracted from the activation of a deep convolutional network trained in a fully supervised fashion on a large, ﬁxed set of object recognition tasks can be repurposed to novel generic tasks [11]. Investigated and visualized the semantic clustering of deep convolutional features with respect to a variety of such tasks, including scene recognition, and ﬁne-grained recognition challenges. Releasing of DeCAF, an open-source implementation of these deep convolutional activation features, along with all associated network parameters. The numerical results consistently and robustly demonstrate that multi-task feature learning framework can substantially improve the performance of a wide variety of existing methods across a spectrum of visual recognition tasks.

**2.4 Multiscale combinatorial grouping*.* In Computer vision and pattern recognition by Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., & Malik, J.**

An approach for bottom-up hierarchical image segmentation and object candidate generation for recognition is called Multiscale Combinatorial Grouping (MCG). For this purpose, ﬁrst a fast normalized cuts algorithm was developed and then a high-performance hierarchical segmenter that makes effective use of multiscale information was proposed [4]. Finally, a grouping strategy was proposed that combines multiscale regions into highly-accurate object candidates by exploring efﬁciently their combinatorial space.

**2.5 Contextualizing object detection and classification by Chen, Q., Song, Z., Huang, Z., Hua, Y., & Yan, S.**

The object classiﬁcation and detection are iteratively and mutually boosted by taking the outputs from one task as the context of the other one. In this method, Contextualized Support Vector Machine (Context-SVM) is proposed, where the context takes the responsibility of dynamically adjusting the classiﬁcation hyperplane, and thus the context-adaptive classiﬁer is achieved. Then, an iterative training procedure is presented [9]. In each step, Context-SVM, associated with the output context from one task, is instantiated to boost the performance for the other task, whose augmented outputs are then further used to improve the former task by Context-SVM and achieved the state-of-the-art performance for both tasks.

**2.6 Going Deeper with Convolutions by Christian Szegedy1, Wei Liu2, Yangqing Jia1, Pierre Sermanet1, Scott Reed3, Dragomir Anguelov1, Dumitru Erhan1, Vincent Vanhoucke1, and Andrew Rabinovich4 .**

Deep convolutional neural network architecture achieves the new state of the art for classiﬁcation and detection in the ImageNet Large-Scale Visual Recognition Challenge. By a carefully crafted design, increased the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing [10]. Results yield a solid evidence that approximating the expected optimal sparse structure by readily available dense building blocks is a viable method for improving neural networks for computer vision.

**2.7 Deep Residual Learning for Image Recognition by Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun.**

A residual learning framework is used to ease the training of networks that are substantially deeper than those used previously. The residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the Image Net dataset, evaluated residual nets with a depth of up to 152 layers—8× deeper than VGG nets but still having lower complexity [23]. An ensemble of these residual nets achieves 3.57% error on the Image Net test set. Due to extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset.

**2.8 Recognizing Images of Handwritten Digits using Learning Vector Quantization Artificial Neural Network by Sunil Kumar Khatri, Shivali Dutta, Prashant Johri .**

Recognition of letter and especially font from images which are containing texts. To obtain font and letter of the text, recognition and separation process is performed respectively. After recognizing, these individual parts are sended to the pre-trained deep convolutional neural network [33]. %79.08 of letter and %75 of font success has been attained in the tests. According to the results, letter recognition with this network has nearly 100 %, but the accuracy of font recognition is low. But using the probability, font recognition percentage has been increased.

**2.9 Simple Convolutional Neural Network on Image Classification, 2017 IEEE 2nd International Conference on Big Data Analysis by Tianmei Guo, Jiwen Dong ,Henjian LiˈYunxing Gao..**

In recent years, deep learning has been used in image classification, object tracking, text detection and recognition and action recognition. Auto Encoder, Sparse coding, Restricted Boltzmann Machine, Deep Belief Networks and Convolutional neural networks is commonly used models in deep learning [34]. Among different type of models, CNN has been demonstrated high performance on image classification. On the basis of the CNN, different methods of learning rate set and optimization algorithm of solving the optimal parameters of the influence on image classification is analyzed.

**2.10 Research on face recognition algorithm based on improved convolution neural network by Liu Hui, Song Yu-jie.**

Automated classification of human anatomy is an important prerequisite for many computer-aided diagnosis systems. “Deep learning” methods such as convolutional networks (ConvNets) is used in image classification tasks [26]. In this work, a method for organ- or body- part-specific anatomical classification of medical images acquired using computed tomography (CT) with ConvNets.

**2.11 Anatomy-specific classification of medical images using deep convolutional nets by Holger R. Roth, Christopher T. Lee, Hoo-Chang Shin, Ari Seff, Lauren Kim, Jianhua Yao, Le Lu, Ronald M. Summers,.**

Using ConvNets and data augmentation, we achieve anatomy-specific classification error of 5.9 % and area-under-the-curve (AUC) values of an average of 0.998 in testing [15].

**2.12 Classification of Very High Resolution SAR Image Based on Convolutional Neural Network by Jinxin Li, Chao Wang, Shigang Wang, Hong Zhang , Bo Zhang.**

The new advanced very high resolution (VHR) synthetic aperture radar (SAR) sensor is a kind of high-tech imaging radar developed rapidly in recent years. To achieve high precision classification performance of the VHR SAR image, convolutional neural network (CNN), a kind of representative deep learning method, is applied [20]. The CNN method obtained better classification result with 97.0% average accuracy. Experimental results demonstrated that the CNN method outperformed other traditional pixel-based methods (such as the minimum distance method and the maximum likelihood method) in VHR SAR image classification.

**2.13 Font and Turkish Letter Recognition in Images with Deep Learning by Aylin Sevik, Pakize Erdogmus, Erdi Yalcin.**

Face recognition algorithm based on improved convolution neural network and Fisher criterion is brought up to resolve the difficulty of poor property of CNN under small samples. CNN is a depth learning algorithm that can automatically extract features [5]. This algorithm makes full use of the advantages of CNN in feature extraction and the advantages of SVM in dealing with small samples and nonlinear problems. The experimental results show that the face recognition algorithm which based on the Fisher neural network combined with SVM can achieve good results in the case of fewer samples.

**2.14 Fully Convolutional Neural Networks for Tissue Histopathology Image Classification and Segmentation by Binbin Peng, Lin Chen, Mingsheng Shang, Jianjun Xu.**

Tissue histopathology image analysis is a new research area of the computer vision. A novel classification and segmentation framework based on fully convolutional neural networks is proposed, which can largely speed up large histopathology image analysis without damaging the accuracy too much [7]. This framework can perform well for stomach histopathology images. The experimental results show that the proposed method achieves comparable accuracy in both classification and segmentation tasks, along with a 16 times faster speed than the tested state-of-the-art methods.

**2.15 Class Agnostic Image Common Object Detection by Shuqiang Jiang , Senior Member, IEEE, Sisi Liang, Chengpeng Chen, Yaohui Zhu, and Xiangyang Li.**

The Common Object Detection Network (CODN) to detect class agnostic common objects from two images. It consists of two main modules: locating module and matching module. The locating module generates candidate proposals of each two images. The matching module learns the similarities of the candidate proposal pairs from two images, and reﬁnes the bounding boxes of the candidate proposals [32]. The learning procedure of CODN is implemented in an integrated way and a multi-task loss is designed to guarantee both region localization and common object matching.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.

Machine learning algorithms give less accuracy in Image classification when compared to Deep learning and also it takes huge time to process the data. Machine learning algorithms can process less number of dataset than in Deep learning algorithms. Some of the machine learning algorithms are Naïve Bayes algorithm, Support vector Machine(SVM), K-means algorithm, Artificial neural network.

**3.1.1 Naïve Bayes theorem**

 Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying  Bayes' theorem with strong (naive) independence assumptions between the features. It is a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features.

With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis. Naive (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables other types of classifiers.

In the statistics and computer science literature, naive Bayes models are known under a variety of names, including simple Bayes and independenceBayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

In plain English, using Bayesian probability terminology, the above equation can be written as

{\displaystyle {\text{posterior}}={\frac {{\text{prior}}\times {\text{likelihood}}}{\text{evidence}}}\,}

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

**3.1.2 K-means algorithm**

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-mean clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

The algorithm has a loose relationship to the *k*-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with *k*-means due to the name. Applying the 1-nearest neighbor classifier to the cluster centers obtained by *k*-means classifies new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm. The algorithm is often presented as assigning objects to the nearest cluster by distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging. Various modifications of *k*-means such as spherical *k*-means and *k*-medoids have been proposed to allow using other distance measures.

Three key features of *k*-means that make it efficient are often regarded as its biggest drawbacks. They are Euclidean distance is used as a metric and variance is used as a measure of cluster scatter, The number of clusters *k* is an input parameter: an inappropriate choice of *k* may yield poor results. That is why, when performing *k*-means, it is important to run diagnostic checks for determining the number of clusters in the data set, Convergence to a local minimum may produce counterintuitive ("wrong") results. The following figure shows the sample K-means algorithm.

**3.1.3 SUPPORT VECTOR MACHINE**

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features we have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.

y

Support Vectors

x

Fig 1: Support vector machine

**Pros and Cons associated with SVM**

**Pros:**

* + - It works really well with clear margin of separation
    - It is effective in high dimensional spaces.
    - It is effective in cases where number of dimensions is greater than the number of samples.

**Cons:**

* It doesn’t perform well, when we have large data set because the required training time is higher.
* It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping.

**Applications**

SVMs can be used to solve various real-world problems:

* Classification of images can also be performed using SVMs.
* Hand-written characters can be recognized using SVM.

**3.1.4 ARTIFICIAL NEURAL NETWORK**

An artificial neural network is an interconnected group of nodes, similar to the vast network of neurons in a brain. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. The goal of the ANN approach was to solve problems in the same way that a human brain would.

For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

**Input Output**

**Hidden**

**Fig 2: Artificial neural network**

**Applications**

Application areas include system identification and control quantum chemistry, general game playing, pattern recognition, medical diagnosis, finance, data mining, visualization, machine translation, social network filtering and e-mail spam filtering.

Artificial neural networks have been used to diagnose cancers, including lung cancer, prostate cancer, colorectal cancer and to distinguish highly invasive cancer cell lines from less invasive lines using only cell shape information.

**3.2 PROPOSED SYSTEM**

In deep learning, a **convolutional neural network** (**CNN** or **ConvNet**) is a class of deep neural networks, most commonly applied to analyzing visual imagery. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers. ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing. Convolutional deep neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).

**CHAPTER 4**

**SYSTEM SPECIFICATION**

**4.1 HARDWARE REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| * CPU SPEED * CPU TYPE | : 3.1 GHz  : 64 bit | |
| * STORAGE | | : 200 GB |
| * OS | | : Microsoft Windows 10 64 bit |
| * RAM | | : 8 GB |

**4.2 SOFTWARE REQUIREMENTS**

* Anaconda navigator
* Colaboratory
* Tensorflow
* Jupyter notebook

**4.3 ABOUT THE SOFTWARE**

**ANACONDA NAVIGATOR**

Anaconda is a free and open-sourcedistribution of the Python and R programming languages for scientific computing (data science,  machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager, called Anaconda Navigator, so it eliminates the need to learn to install each library independently. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands.

The following applications are available by default in Anaconda Navigator

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Lab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* [QtConsole](https://qtconsole.readthedocs.io/en/latest/)
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glueviz](http://glueviz.org/)
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [Rstudio](https://en.wikipedia.org/wiki/Rstudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)

**Conda**

Conda is an open source, cross-platform language-agnostic [package manager](https://en.wikipedia.org/wiki/Package_manager) and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

**Anaconda cloud**

Anaconda Cloud is a package management service by Anaconda where you can find, access, store and share public and private notebooks, environments, and conda and PyPI packages. Cloud hosts useful Python packages, notebooks and environments for a wide variety of applications. You do not need to log in or to have a Cloud account, to search for public packages, download and install them.

**Jupyter Notebook**

Jupyter [Notebook](https://en.wikipedia.org/wiki/Notebook_interface) (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebooks documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context.

  A Jupyter Notebook document is a [JSON](https://en.wikipedia.org/wiki/JSON) document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.wikipedia.org/wiki/Markdown)), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

**Colaboratory**

**Colaboratory**(short for **Colab)**, a tool for machine learning education and research, created as a Google research project. Creating and running Jupyter Notebook on Colab is super easy and it’s free. We can save your notebook to Google Drive or GitHub and even train your deep learning on  Graphics Processing Units (GPU).

By default it’s created in your Google Drive under Colab Notebooks. Since it’s in Google Drive, you can move it around and share it just like any other Drive documents.

**TensorFlow**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. It is a standard expectation in the industry to have experience in TensorFlow to work in machine learning.

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE**

A Deep Structured neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.

**Fig 3:** Architecture of Deep structured neural network

Neural Networks receive an input (a single vector), and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully-connected layer is called the “output layer” and in classification settings it represents the class scores.

**5.2 SYSTEM MODULES**

* + Convolution
  + ReLU
  + Pooling
  + Flatten
  + Fully connected
  + Softmax

**Convolution**

The Conv layer is the core building block of a Convolutional Network that does most of the computational work.Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The name ConvNets had been derived from an operator called ‘Convolution’. The primary objective of this operator is the extraction of input image features. Convolution learns image features and works in coordination with pixels by using small squares of input data.

Consider a 5 x 5 image with pixel values as 0 and 1 only. Pixel values vary from 0 to 255.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Consider one more 3 x 3 matrix as shown below:

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Convolution of the 5 x 5 image and the 3 x 3 matrix can be calculated below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 4 | 3 | 4 |
| 2 | 4 | 3 |
| 2 | 3 | 4 |

Convolved Feature

Image

**Fig 4:** Convolution layer

For each position, we perform element-wise multiplication among the two matrices and summate the outputs of multiplication to obtain the final single element (an integer) of the output matrix. The computation of the dot product is referred as the ‘Convolved Feature’ or ‘Activation Map’ or the ‘Feature Map’.

**ReLU**

In a typical ConvNet, a supplementary operation known as ReLU is performed after every convolution operation. ReLU is the abbreviation for Rectified Linear Unit and to our surprise, is itself a non-linear operation.

ReLU is an operation working element-wise i.e. is applied per pixel and substitutes every negative pixel value by 0 in the feature map. It serves the purpose of introducing non-linearity in ConvNet, because the maximum real-life data we will want to feed into our ConvNet is non-linear. Convolution is a linear operation i.e. possesses element-wise matrix type multiplication and addition and so, there’s a need for the introduction of a nonlinear function such as ReLU to account for non-linearity.

Non-linear functions such as ‘tanh’ or ‘sigmoid’ can be put to use for operation in the place of ReLU. However, ReLU is most widely used due to its consistently better performance in most situations.

Filter 1 Feature map

|  |  |  |
| --- | --- | --- |
| 4 | 3 | 0 |
| 0 | 4 | 3 |
| 2 | 0 | 4 |

|  |  |  |
| --- | --- | --- |
| 4 | 3 | -4 |
| -2 | 4 | 3 |
| 2 | -3 | 4 |

**Fig 5:** ReLU

**Pooling**

To reduce the dimensionality of individual feature maps and yet sustain the crucial information, we use Spatial Pooling commonly known as downsampling or subsampling.

Spatial pooling can be of different types:

* Max Pooling
* Average Pooling
* Sum Pooling

For Max Pooling, we first specify a spatial neighborhood (such as a 2×2 window) and then pick out the largest element of that feature rectified map within that window. Instead of largest element, if we pick the Average one it’s called Average Pooling and when the summation of elements in the window is taken, we call it Sum Pooling. Among these, Max Pooling has proven itself as the best.

Convolution + ReLU operation is executed on a rectified feature map and then Max Pooling operation is performed by using a 2×2 window as shown below.

It Combine the output of the neuron clusters at one layer into a single neuron in the next layer. It Uses the maximum value from each of a cluster of neurons at the prior layer and reduce the spatial size of the representation to reduce the amount of parameters computations in the network.

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **1** | **2** | **4** |
| **5** | **6** | **7** | **8** |
| **3** | **2** | **1** | **0** |
| **1** | **2** | **3** | **4** |

x

|  |  |
| --- | --- |
| **6** | **8** |
| **3** | **4** |

**x**

Max pool with 2×2 filters and stride 2

Rectified feature map

**y**

**Fig 6:** Pooling

The basic purpose of pooling is to tremendously reduce the structural size of the given input, lessens the parameters used, makes the network resistant to distortions, translations in the input image and to small transformations and the computations in the network to efficiently control over fitting.

**Flatten**

Once the pooled featured map is obtained, the next step is to flatten it. Flattening involves transforming the entire pooled feature map matrix into a single column which is then fed to the neural network for processing.

|  |
| --- |
| **4** |
| **3** |
| **0** |
| **0** |
| **4** |
| **3** |
| **2** |
| **0** |
| **1** |

|  |  |  |
| --- | --- | --- |
| 4 | 3 | 0 |
| 0 | 4 | 3 |
| 2 | 0 | 4 |

Flattening

Pooled feature map

**Fig 7:** Flatten

**Fully Connected**

The output layer in a **CNN** as mentioned previously is a fully connected layer, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network.

The word “Fully Connected” in the fully connected layer indicate that every neuron in the next layer is connected to every individual neuron in the previous layer. This step is made up of the input layer, the fully connected layer, and the output layer. This layer is described as a Multilayer Perceptron which utilizes a softmax activation function present in the output layer.

In the fully-connected operation of a neural network, the input representation is flattened into a feature vector and passed through a network of neurons to predict the output probabilities.

Pooling and convolutional layer outputs depict high-resolution features of the input image. The fully connected layer on the basis of the training dataset utilizes these features for categorizing input images into different classes.

**Softmax**

The soft-max layer outputs a probability distribution, i.e. the values of the output sum to 1. Softmax is implemented through a neural network layer just before the output layer. The Softmax layer must have the same number of nodes as the output layer.

The softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function. But it also divides each output such that the total sum of the outputs is equal to 1.

**CHAPTER 6**

**DATASETS USED**

Deep Structured Neural Network aims to classify dog and cat in a largescale image dataset. This classification is an essential step in a variety of applications, such as image recognition, Medical image analysis, Face recognition and object detection, etc. Because of its importance, a significant number of benchmark datasets are available for image classification. One of those datasets we used Standard benchmark Kaggle image dataset.

**KAGGLE IMAGE DATASET**

This data set was collected over the Kaggle. Kaggle is an online community of data scientists and machine learners, owned by Google LLC. Kaggle allows users to find and publish data sets.

Kaggle is an online community of data scientists and machine learners, owned by Google LLC. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

Our dataset consist of nearly 22,000 images for training and testing the images. After extracting the dataset, the images migrate to a directory Called Data.

In the data folder, the dataset is categorized into five different packages. They are models, sample, test, train and valid. The total size of data folder is 817 MB.

**Fig 8:** Sample Dog images

The Dog and Cat images are downloaded from Standard benchmark Kaggle.



**Fig 9:** Sample Cat images

**CHAPTER 7**

**SYSTEM IMPLEMENTATION**

**7.1 Input Extraction**

Our dataset consist of nearly 22,000 images for training and testing the images. After extracting the data set, the images migrate to a directory Called Data.

Original dataset

Extracting the data set

Migrate to data directory

**Fig 10:**Input

* 1. **Convolution+ ReLU**

The Conv layer is the core building block of a Convolutional Network that does most of the computational work.Convolutional layers apply a convolution operation to the input, passing the result to the next layer. ReLU is an operation working element-wise i.e. is applied per pixel and substitutes every negative pixel value by 0 in the feature map.

**Fig 11:** convolution + ReLU operation

**Convolution+ ReLU Algorithm**

**Step 1:** Download the dogs and cats dataset and save it to a local directory called data.

**Step 2:** Calculate the match to a feature across the whole image.

**Step 3:** Multiply each pixel in the feature by the value of the corresponding pixel in the image.

**Step 4:** Sum the multiplied values to have a single number.

*Repeat this process for every possible image patch*

**Step 5:** Improve the non-linearity in the image by using Activation function ReLU (Rectified linear Unit).

**7.3** **Max pooling**

For Max Pooling, we first specify a spatial neighborhood (such as a 2×2 window) and then pick out the largest element of that feature rectified map within that window.

Placed in between convolution layers

Reduce the spatial size of data

Identify the important features

**Fig 12:** Pooling process

**Max pooling Algorithm**

**Step 1:** Combine the outputs of neuron clusters at one layer into a single neuron in the next layer.

**Step 2:** Uses the maximum value from each of a cluster of neurons at the prior layer.

**Step 3:** Reduce the spatial size of the representation to reduce the amount of parameters and computation in the network.

**7.4 Fully Connected**

The word “Fully Connected” in the fully connected layer indicate that every neuron in the next layer is connected to every individual neuron in the previous layer. This step is made up of the input layer, the fully connected layer, and the output layer.

* + They usually sit at the top of the network hierarchy.
  + To identify the specific features detected by the lower layers.
  + The input has been reduced to a compact representation of features.

**Fully Connected Algorithm**

**Step 1:** Connect every neuron in one layer to every neuron in another layer.

**Step 2:** Flatten the input to a compact representation of features.

**Step 3:** The flattened matrix goes through a fully connected layer to classify the images.

**CHAPTER 8**

**CONCLUSION AND FUTURE WORK**

This paper proposes the use of Deep Structured Neural Network model for the classification of images. We have implemented this network model using Tensor Flow. Neural Networks receive an input, and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully-connected layer is called the “output layer” and in classification settings it represents the class scores.

This proposed method has achieved a high accuracy of 100% in classifying the images. However, we used a GPU based system instead of CPU based system, the training time would have been less.

To achieve better computational power, we will use Recurrent Neural Networks (RNN) in future. It is a powerful and robust type of neural networks and belong to the most promising algorithms out there at the moment because they are the only ones with an internal memory. Because of their internal memory, RNN’s are able to remember important things about the input they received, which enables them to be very precise in prediction.

**APPENDIX**

1. **SAMPLE CODE**

***##### download the dogs and cats dataset and save it to a local directory called data***

!mkdir data && wget http://files.fast.ai/data/dogscats.zip && unzip dogscats.zip -d data/

***#### Extract a list of image file names from the directory to visualise***

PATH = 'data/dogscats/'

***#### run the list files command and get the top results and save it into a list object***

files = !ls {PATH}valid/cats | head

print(files)

***#### Extract the label of the image from the image path***

label = imagePath.split(os.path.sep)[-1].split(".")[0]

***#### Resize the Image and then use the raw pixel values as features***

features = cv2.resize(image, (32, 32))

***#### Append the features and labels to our list variables***

data.append(features)

labels.append(label)

***#### Create a train test split***

(trainData, testData, trainLabels, testLabels) =

train\_test\_split(data, labels, test\_size=0.25, random\_state=42)

***#### input image dimensions***

input\_shape = data[0].shape

model = Sequential()

***####Create a Convolution and Relu***

model.add(Conv2D(32, kernel\_size=(3, 3), strides=(1, 1),

activation='relu', input\_shape = input\_shape))

***####Create a Max pooling layer***

model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(64, (5, 5), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

***#### Create a Flatten***

model.add(Flatten())

model.add(Dense(1000, activation='relu'))

***#### Create a Softmax***

model.add(Dense(2, activation='softmax'))

**##### Intialise a Stochastic Gradient Descent Optimiser Object**

sgd = SGD(lr=0.01)

model.compile(loss="binary\_crossentropy",optimizer=sgd,metrics=["accuracy"])

***#### Train the Model***

model.fit(trainData, trainLabels, epochs=2, batch\_size=256)

***##### Flatten the test image***

test\_feature = cv2.resize(image, (32, 32))

print(test\_feature.shape)

***##### Reshape the test image shape to match the network thats expecting a set of images***

test\_feature=test\_feature.reshape(1,test\_feature.shape[0],test\_feature.shape[1],test\_feature.shape[2])

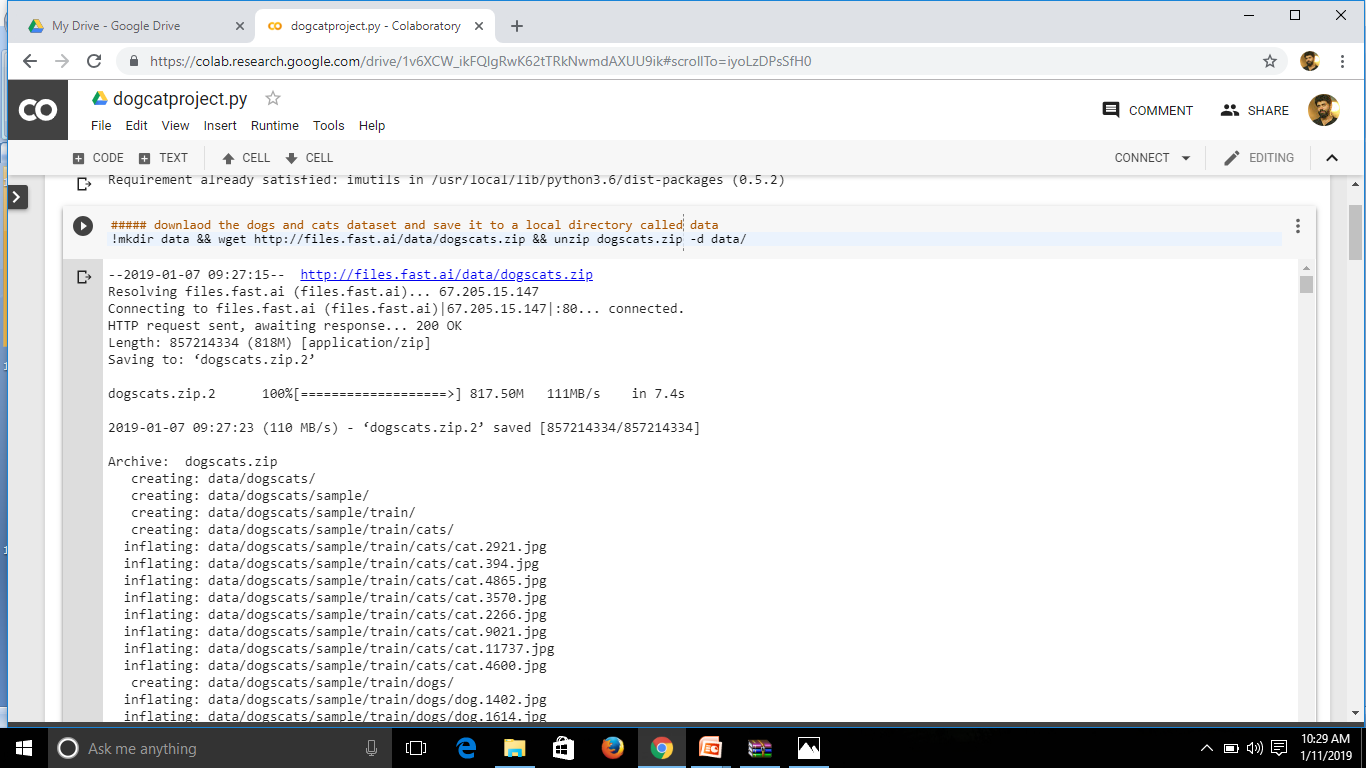
print(test\_feature.shape)

***##### Print Predictions***

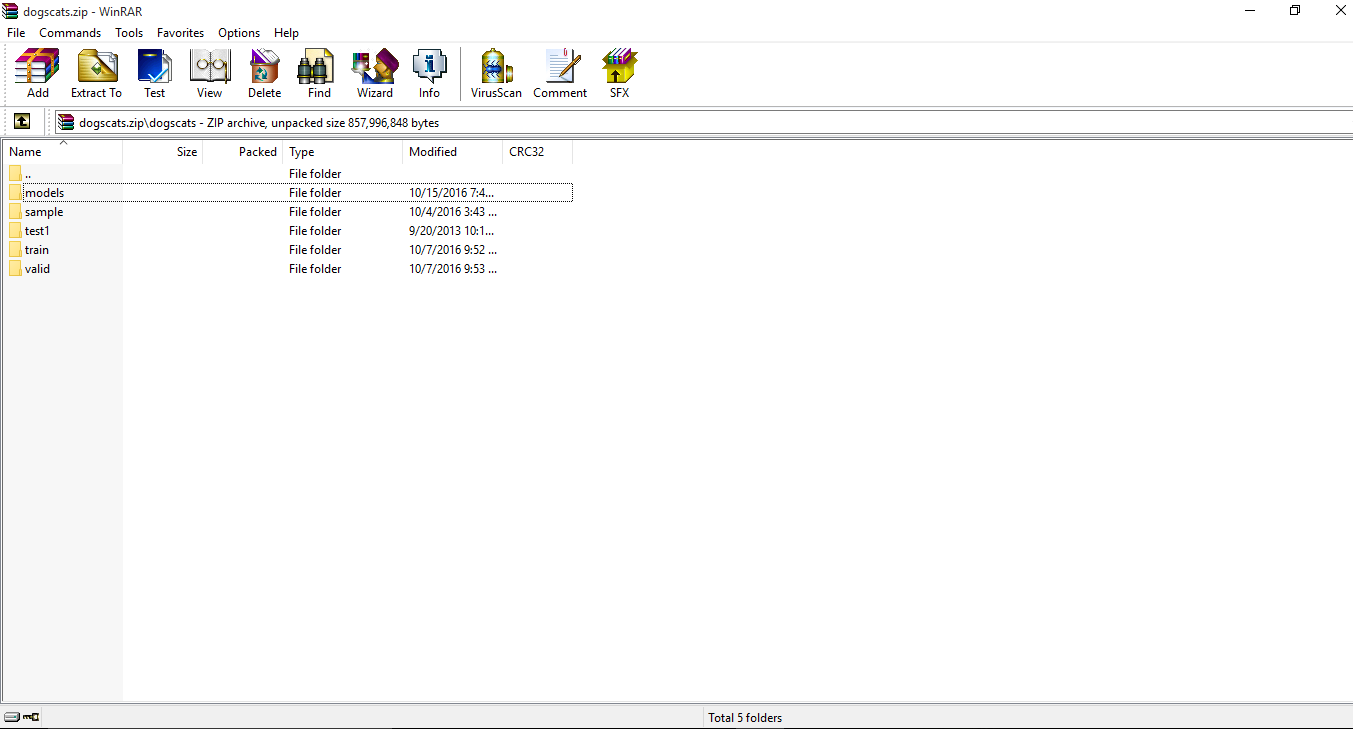
test\_prediction = model. predict(test\_feature)

print("[INFO]CAT={:.10f}%,DOG:{:.10f}%".format(test\_prediction[0][0]\*100, test\_prediction[0][1]\*100))

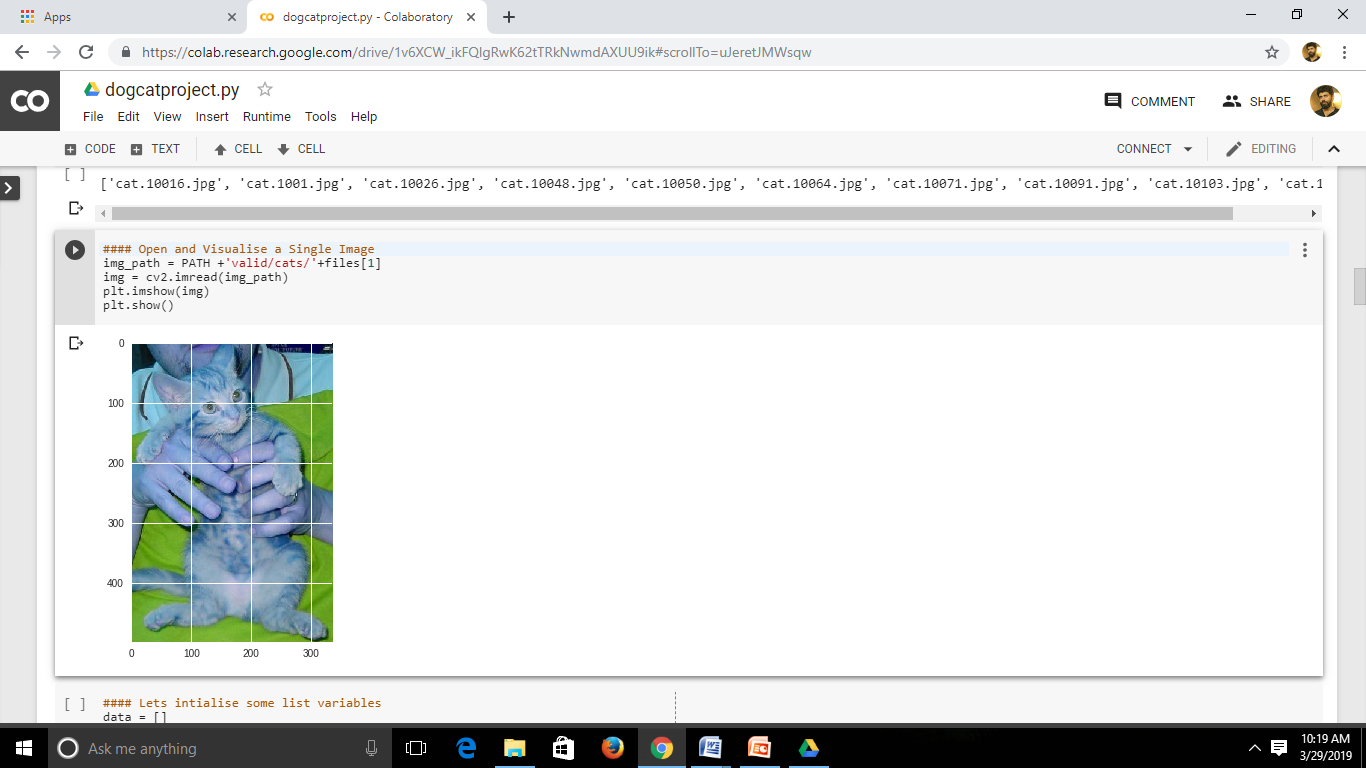
1. **SCREENSHOTS**

****

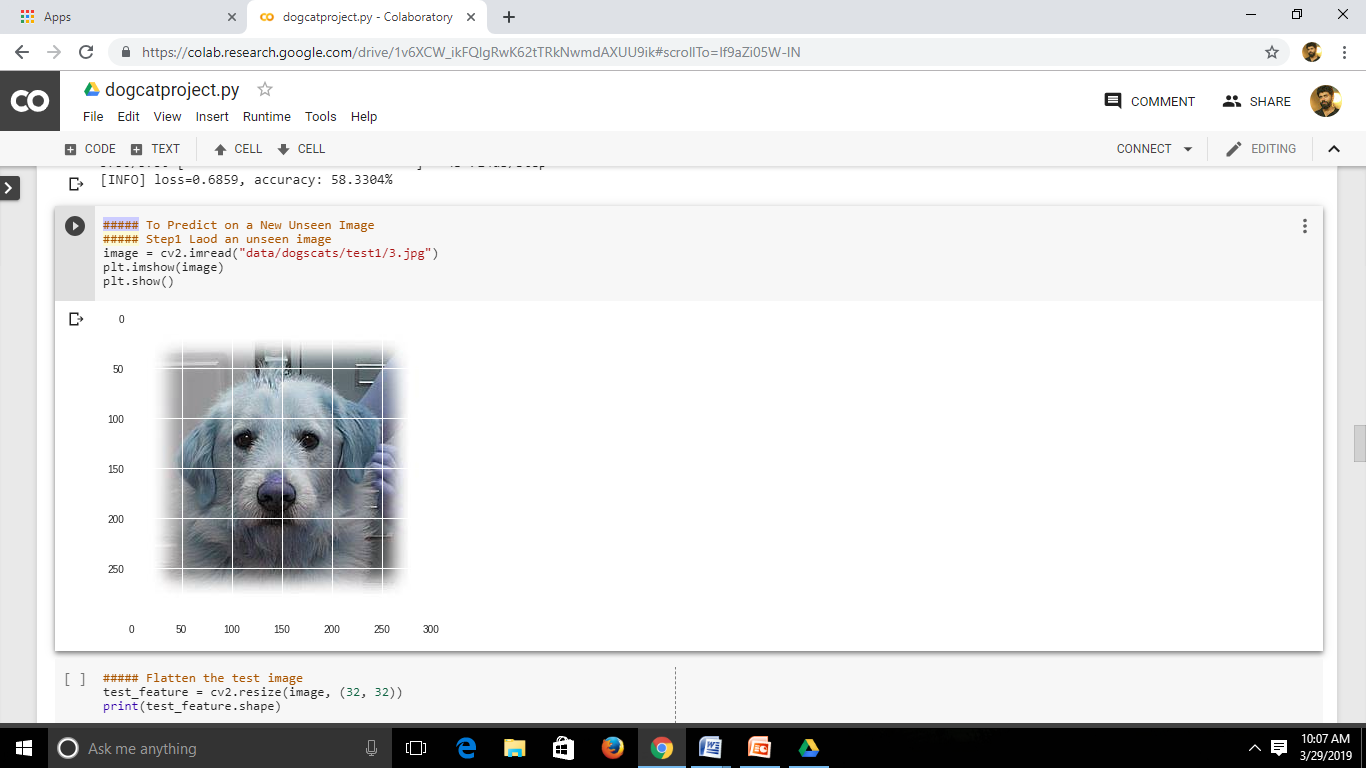
**Fig 13:** Extraction of dataset

****

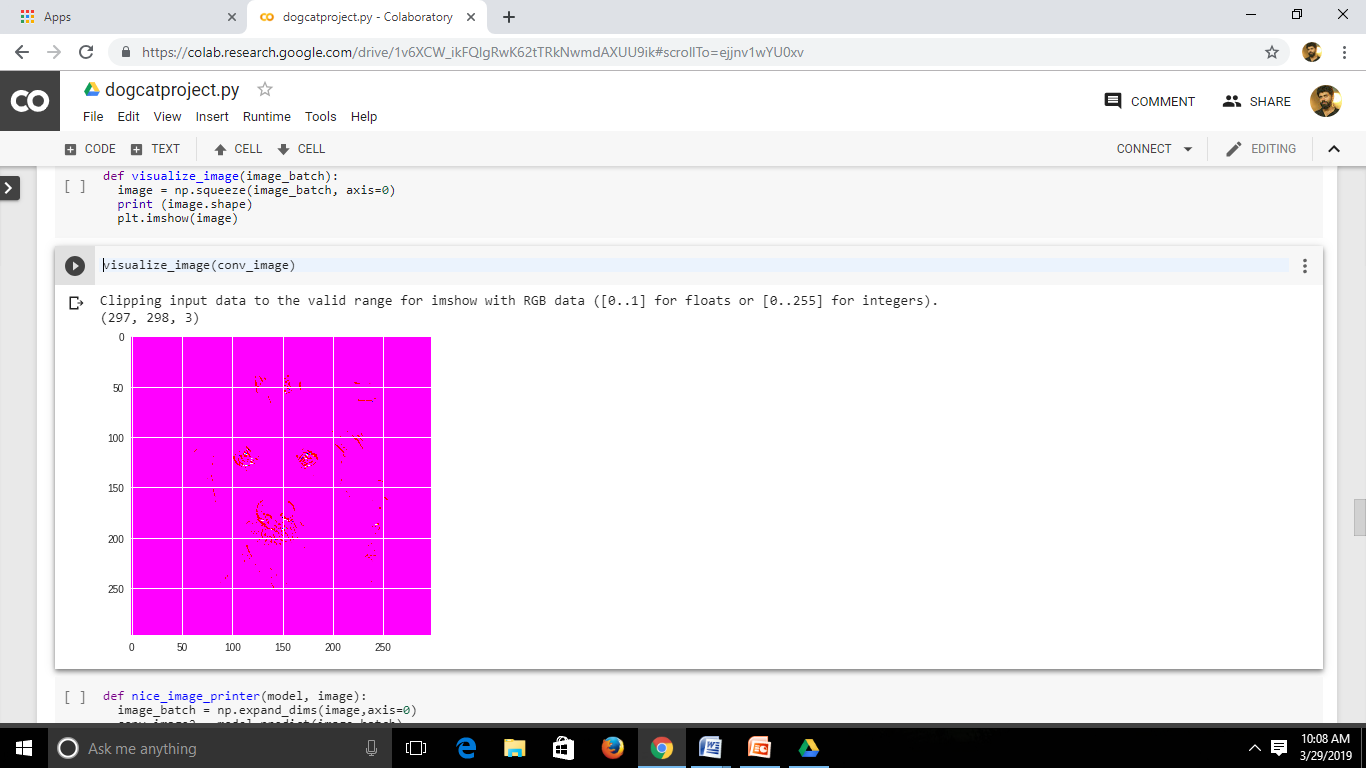
**Fig 14:** Dataset



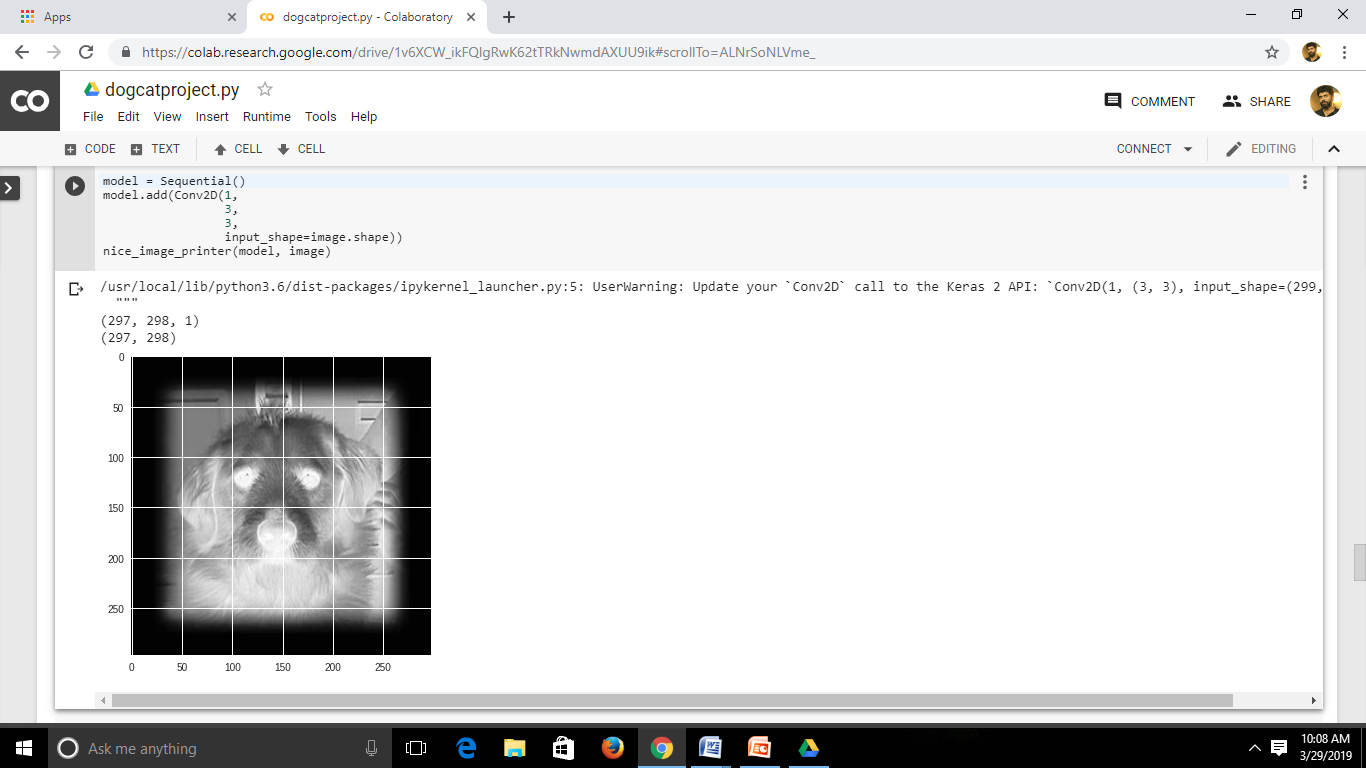
**Fig 15:** Image visualisation



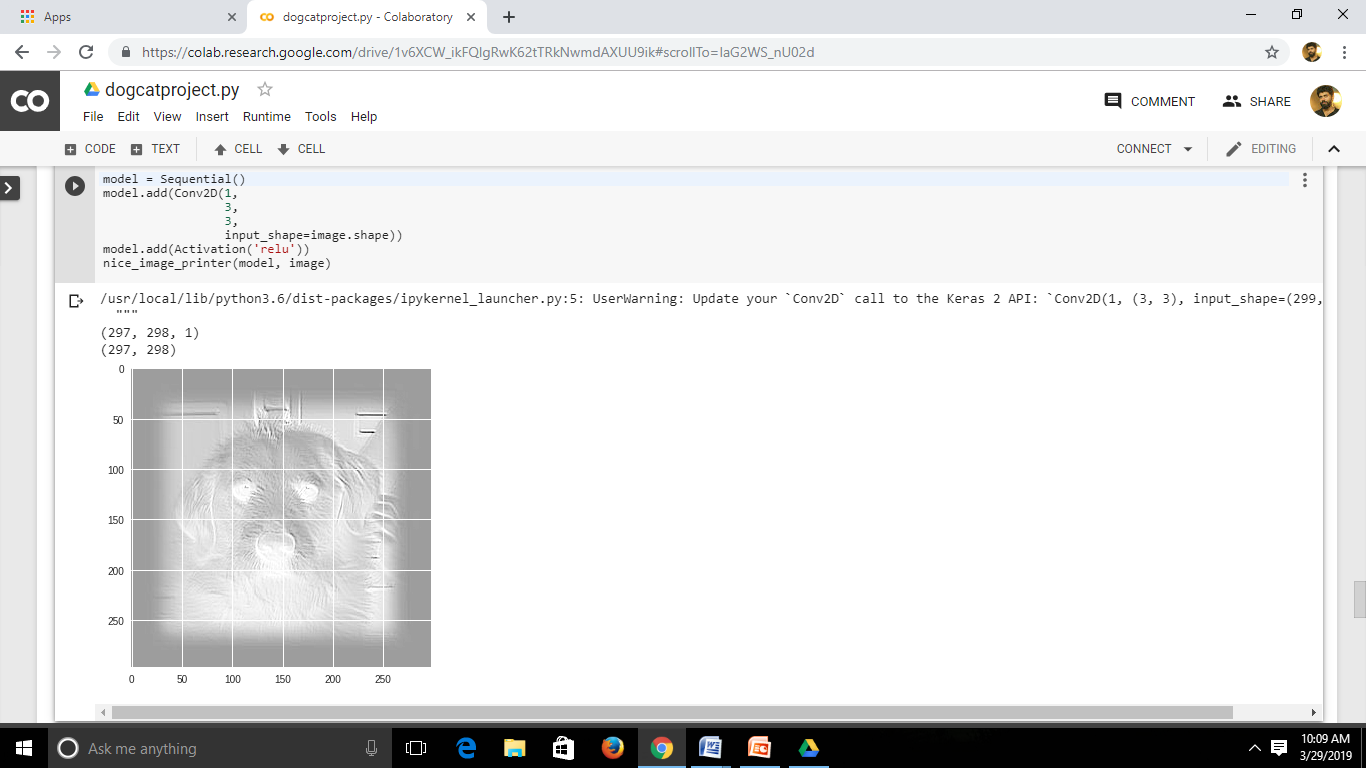
**Fig 16:** Test image visualisation



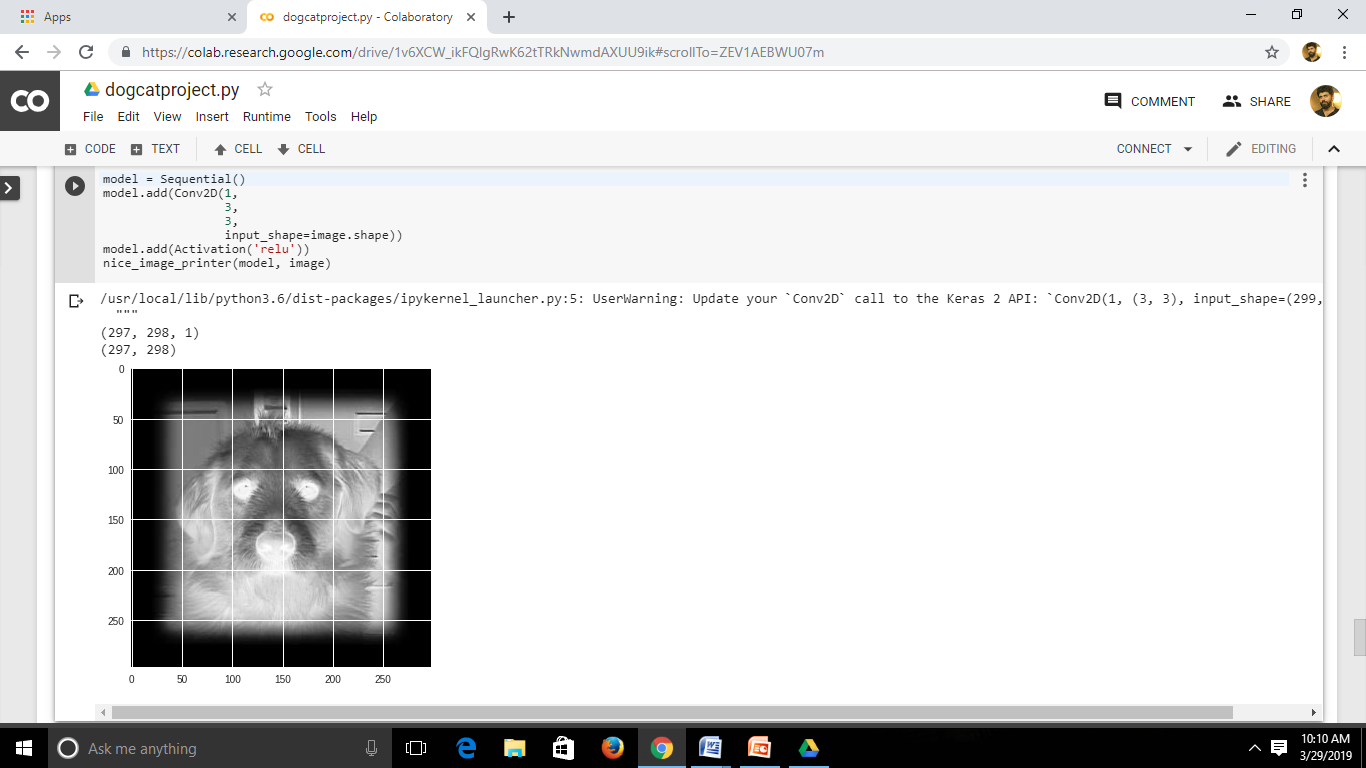
**Fig 17:** Convolution image



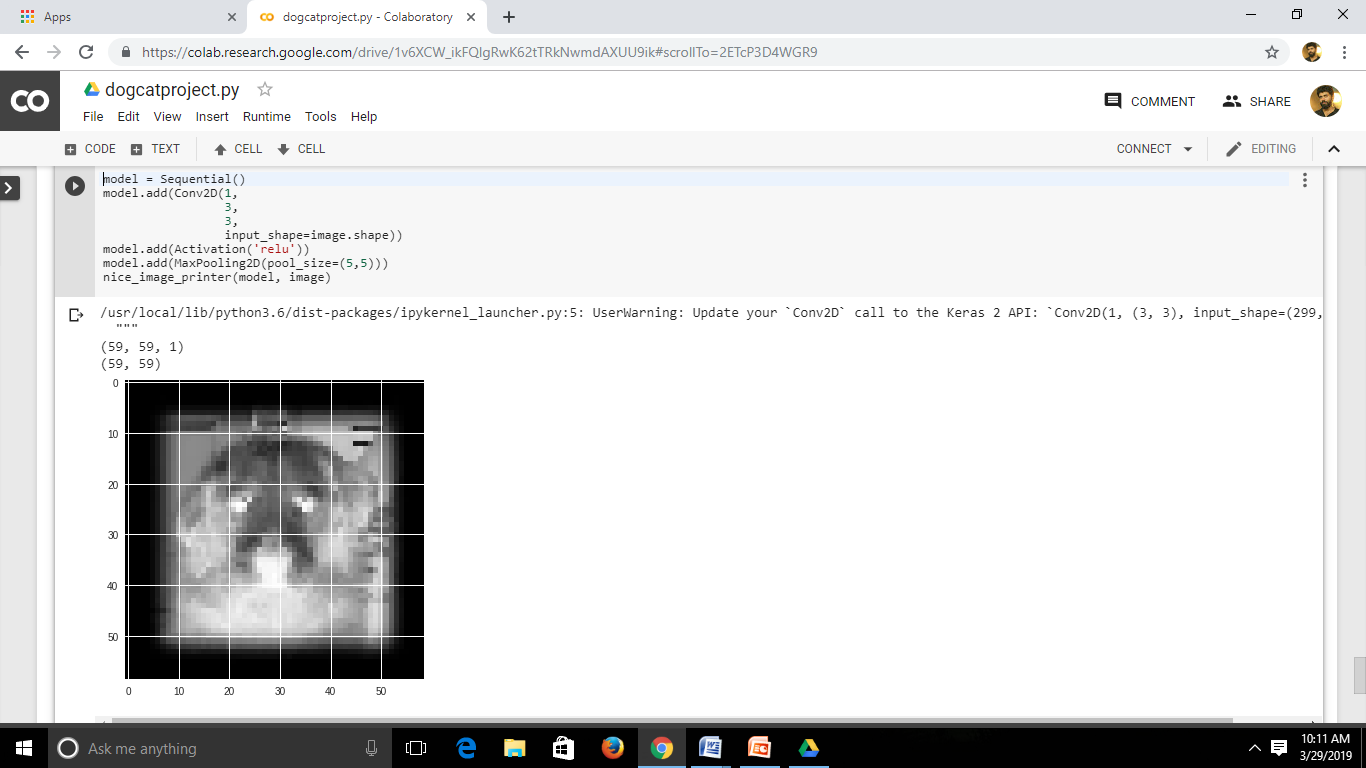
**Fig 18:** Conv2D image



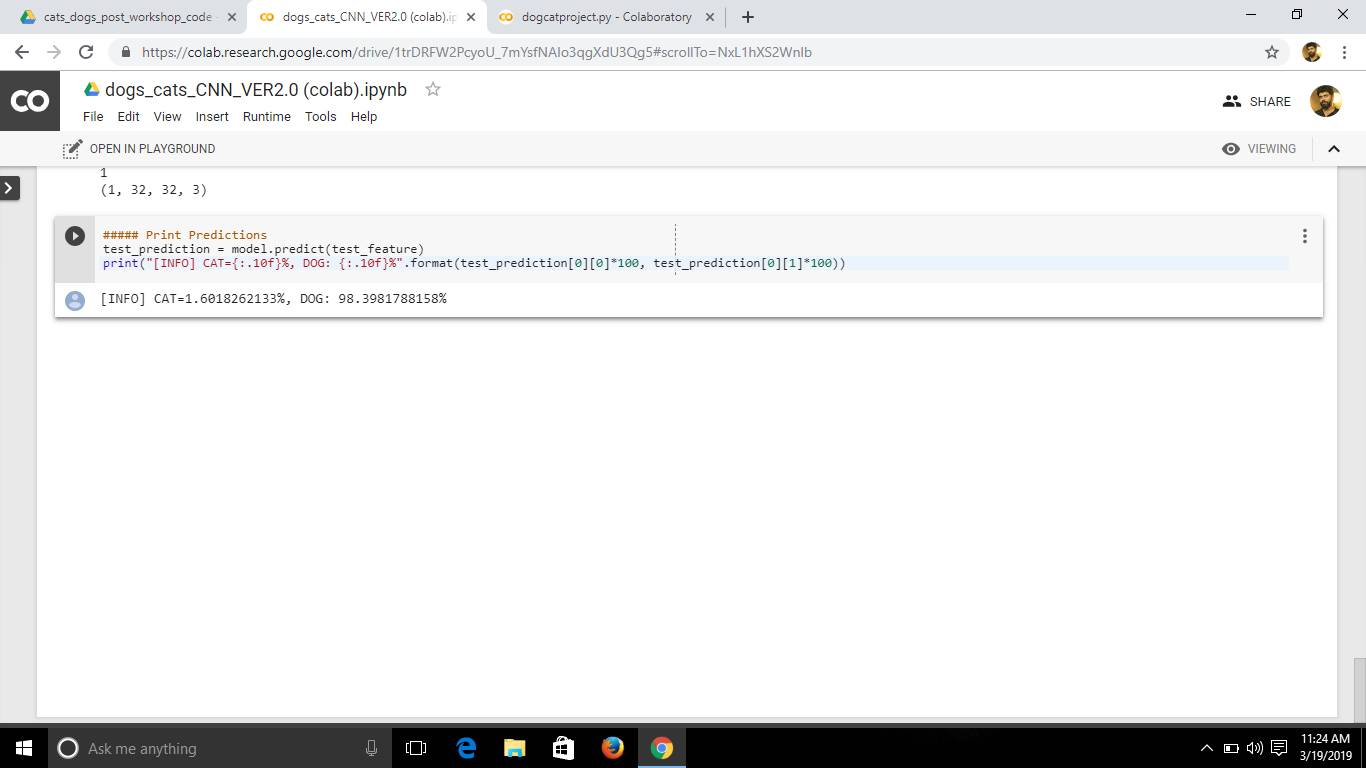
**Fig 19:** Image after ReLU



**Fig 20:** Convoluted image



**Fig 21:** Pooled image



**Fig 22:** Prediction

**REFERENCES**

1. A. pentland, B. Moghaddam , and T. Starner , “View-Based and Modular Eigenspaces for Face Recognition,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pp. 84-91, 1994.
2. A. Pirinen and C. Sminchisescu , “Deep reinforcement learning of region proposal networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 6945–6954.
3. Ahonen, T., Hadid, A., & Pietikinen, M. (2006). Face description with local binary patterns: Application to face recognition. ‘Pattern Analysis and Machine Intelligence’, *28*(14), 2037–2041.
4. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., & Malik, J. (2014). Multiscale combinatorial grouping*.* In Computer vision and pattern recognition, 2014.
5. Aylin Sevik, Pakize Erdogmus, Erdi Yalcin (2018) Font and Turkish Letter Recognition in Images with Deep Learning, 3-4 Dec 2018.
6. B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” Science, vol. 350, no. 6266, pp. 1332–1338, 2015.
7. Binbin Peng, Lin Chen, Mingsheng Shang, Jianjun Xu (2018) Fully Convolutional Neural Networks for Tissue Histopathology Image Classification and Segmentation, 2018.
8. C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, “Visual relationship detection with language priors,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 852–869.
9. Chen, Q., Song, Z., Huang, Z., Hua, Y., & Yan, S. (2014). ‘Contextualizing object detection and classification’, 2014.
10. Christian Szegedy1, Wei Liu2, Yangqing Jia1, Pierre Sermanet1, Scott Reed3, Dragomir Anguelov1, Dumitru Erhan1, Vincent Vanhoucke1, andAndrew Rabinovich4(2015) ‘Going Deeper with Convolutions’,2015.
11. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2013), ‘Decaf: A deep convolutional activation feature for generic visual recognition’, CoRR, abs/1310.1531.
12. G. Ding, W. Chen, S. Zhao, J. Han, and Q. Liu, “Real-time scalable visual tracking via quadrangle kernelized correlation ﬁlters,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 1, pp. 140–150, Jan. 2018.
13. H. Izadinia, F. Sadeghi, and A. Farhadi, “Incorporating scene context and object layout into appearance modeling,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 232–239.
14. Handbook of Face Recognition, S.Z. Li and A.K. Jain, eds. Springer, 2005.
15. Holger R. Roth, Christopher T. Lee, Hoo-Chang Shin, Ari Seff, Lauren Kim, Jianhua Yao, Le Lu, Ronald M. Summers, Anatomy-specific classification of medical images using deep convolutional nets.
16. J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4080–4090.
17. J. Yang, D. Zhang, A.F. Frangi, and J. Yang, “Two-dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 1, pp. 131-137, Jan 2004.
18. J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal, “Relationship proposal networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 1, Jul. 2017, pp. 5226–5234.
19. C. Galleguillos, A. Rabinovich, and S. Belongie, “Object categorization using co-occurrence, location and appearance,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.
20. Jinxin Li, Chao Wang, Shigang Wang, Hong Zhang , Bo Zhang, Classification of Very High Resolution SAR Image Based on Convolutional Neural Network.
21. K. Etemad and R. Chellappa, “Discriminant Analysis for Recognition of Human Face Images,” J. Optical Soc. Am., vol. 14, pp. 1724-1733, 1997.
22. K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2980–2988.
23. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun(2015), ‘Deep Residual Learning for Image Recognition’, 2015.
24. L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 594–611, Apr. 2006.
25. L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, “Face Recognition by Elastic Bunch Graph Matching,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, pp. 775-779, 1997.
26. Liu Hui, Song Yu-jie (2018) Research on face recognition algorithm based on improved convolution neural network, 2018.
27. M. Turk and A. Pentland, “Eigenfaces for Recognition,” J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.
28. O. Vinyals, C. Blundell, T. Lillicrap, K. K
29. P.S. Penev and J.J. Atick, “Local Feature Analysis: A General Statistical Theory for Object Representation,” Network-Computation in Neural Systems, vol. 7, no. 3, pp. 477-500, Aug. 1996.
30. S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, “Multi-class segmentation with relative location prior,” Int. J. Comput. Vis., vol. 80, no. 3, pp. 300–316, 2008.
31. Sameer Singhi, Markos Markou & John Haddon (2000) Natural object classification using Artificial Neural Networks, 2000.
32. Shuqiang Jiang , Senior Member, IEEE, Sisi Liang, Chengpeng Chen, Yaohui Zhu, and Xiangyang Li(2019) ‘Class Agnostic Image Common Object Detection’, VOL. 28, NO. 6, JUNE 2019.
33. Sunil Kumar Khatri, Shivali Dutta, Prashant Johri (2015) Recognizing Images of Handwritten Digits using Learning Vector Quantization Artificial Neural Network, 2015.
34. Tianmei Guo, Jiwen Dong ,Henjian LiˈYunxing Gao (2017) Simple Convolutional Neural Network on Image Classification, 2017 IEEE 2nd International Conference on Big Data Analysis.