Cats vs Dogs Classification

Industrial/In-house Training REPORT

Submitted in partial fulfilment of the requirements for the award of

the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION ENGINEERING

by

RAJA	Prince Sareen	Vikram Kumar
04951202819	04551202819	04851202819

Guided by

Adgaonker Shashank

Designation

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)
NEW DELHI – 110063

March 2021

CANDIDATE'S DECLARATION

It is hereby certified that the work which is being presented in the B. Tech Industrial/In-house training Report entitled **"Cats vs Dogs Classification"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** and submitted in the **Department of Electronics & Communication Engineering** of **BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING, New Delhi (Affiliated to Guru Gobind Singh Indraprastha University, Delhi)** is an authentic record of our own work carried out during a period from **February2021 to March 2021** under the guidance of **Adgaonker Shashank , Designation.**

The matter presented in the B. Tech Industrial/In-house training Report has not been submitted by me for the award of any other degree of this or any other Institute.

RAJA	**Prince Sareen**	**Vikram Kumar**
04951202819	**04551202819**	**04851202819**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge. He/She/They are permitted to appear in the External Industrial/In-house training Examination

Adgaonker Shashank) **Mrs. Prof. Kirti Gupta**
Designation **Head, ECE**

The B. Tech Industrial/In-house training Viva-Voce Examination of **RAJA, Prince Sareen, Vikram kumar (Enrollment No: 04951202819 , 04551202819, 04851202819),** has been held on …….

Dr S B Kumar

Industrial/In-house training Coordinator **(Signature of External Examiner)**

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

We express our deep gratitude to **Adgaonker Shashank**, Designation for his valuable guidance and suggestion throughout my project work. We are thankful to **Winter training** for their valuable guidance.

We would like to extend my sincere thanks to **Head of the Department, Prof. Kirti Gupta** for her time to time suggestions to complete my project work. I am also thankful to **Dharmender Saini, Principal** for providing me the facilities to carry out my project work.

RAJA	**Prince Sareen**	**Vikram Kumar**
04951202819	**04551202819**	**04851202819**

ABSTRACT

Artificial Intelligence is basically the mechanism to incorporate human intelligence into machines through a set of rules (algorithm). Machine Learning is basically the study/process which provides the system (computer) to learn automatically on its own through experiences it had and improve accordingly without being explicitly programmed. ML is an application or subset of AI. ML focuses on the development of programs so that it can access data to use it for themselves. The entire process makes observations on data to identify the possible patterns being formed and make better future decisions as per the examples provided to them. The major aim of ML is to allow the systems to learn by themselves through the experience without any kind of human intervention or assistance. Deep Learning is basically a sub-part of the broader family of Machine Learning which makes use of Neural Networks(similar to the neurons working in our brain) to mimic human brain-like behavior. DL algorithms focus on information processing patterns mechanism to possibly identify the patterns just like our human brain does and classifies the information accordingly. DL works on larger sets of data when compared to ML and prediction mechanism is self-administered by machines.

The topic of our project is Cats vs Dogs classification and for building the model we have used Deep Learning algorithm CNN (Convolutional Neural Network) A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. And we have used metrics to find accuracy to avulate our model

Chapter 1
Introduction

This project aims to classify the input image as either a dog or a cat image. The image input which you give to the system will be analyzed and the predicted result will be given as output. Machine learning algorithm [Convolutional Neural Networks] is used to classify the image. The model thus implemented can be extended to a mobile device or any website as per the developer's need..

1.1 Motivation and Background

The Dogs vs. Cats competition from Kaggle is trying to solve the CAPTCHA challenge, which relies on the problem of distinguishing images of dogs and cats. It is easy for humans, but evidence suggests that cats and dogs are particularly difficult to tell apart automatically. Many people has worked or are working on constructing machine learning classifiers to address this problem. In, a classifier based on color features got 56.9% accuracy on the Asirra dataset. In , an accuracy of 82.7% was achieved from a SVM classifier based on a combination of color and texture features. And in, they used the SIFT (Scale-Invariant Feature Transform) features to train a classifier and finally got an accuracy of 92.9%. In our project, we also would like to solve this problem and achieve higher performance. We tried different strategies. For instance, we tried Dense-SIFT features and the combination of DenseSIFT and color features, and features learned from CNN. Also, we employed SVMs on the learned features and finally achieved our best classication accuracy of 88.00%

1.2 Conceptual Framework:

The project is entirely implemented using Python3. The Conceptual Framework involved is mainly:

- Keras – Tensorflow backend
- OpenCV – Used to handle image operations

1.3 Importance of classification

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. ... Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output. In deep learning, neural networks have a significant role. These are a set of algorithms that we implement to identify relevant relationships in datasets, and they follow the process that imitates the human brain. Neural networks depict the behavior of the human brain and enable computer algorithms to identify trends. It also solves complex problems in the domain of machine learning, AI, and data science.

Deep learning deploys artificial neural networks to recognize the hidden patterns of data in the dataset provided. These algorithms are trained over an adequate amount of time and applied to a data set.

Deep learning uses artificial neural networks (ANN) to find the hidden patterns. These patterns are the connection between various variables present in a dataset.
ANN algorithms are trained over a high volume of sample data and then applied to a new dataset. Such algorithms stimulate the way for information processing and communicate experiences similar to the biological nervous system. Deep learning has become a part of our everyday lives: from search engines to self-driving cars that demand high computational power.

1.4 Algorithm

This model is created using the machine & and deep learning algorithms i.e CNN Convolutional Neural Network (CNN) is an algorithm taking an image as input then assigning weights and biases to all the aspects of an image and thus differentiates one from the other. Neural networks can be trained by using batches of images, each of them having a label to identify the real nature of the image (cat or dog here). A batch can contain few tenths to hundreds of images. For each and every image, the network prediction is compared with the corresponding existing label, and the distance between network prediction and the truth is evaluated for the whole batch. Then, the network parameters are modified to minimize the distance and thus the prediction capability of the network is increased. The training process continues for every batch similarly.
CNN does the processing of Images with the help of matrixes of weights known as filters. They detect low-level features like vertical and horizontal edges etc. Through each layer, the filters recognize high-level features.

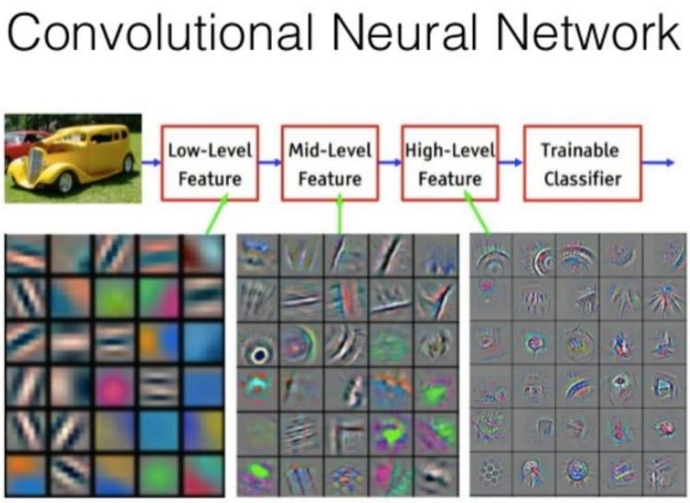

Fig 1.1 CNN

1.5 Sections

The proposed report organized as follows. In section 2, various works in the field are discussed and the gap in exploring using machine learning and deep learning techniques available in python has been highlighted. Section 3 discusses the methodology of the approaches applied in this report using a block diagram. Results and discussions are detailed in the section 4. This section explores the results for better understanding. It is also important that the performance metrics derived from the models is proving the high accuracy and efficiency of the built model. Section 5 concludes the work done in this report.

Chapter 2
Literature Review

This project aims to classify the input image as either a dog or a cat image. The image input which you give to the system will be analyzed and the predicted result will be given as output. Machine learning algorithm [Convolutional Neural Networks] is used to classify the image. The model thus implemented can be extended to a mobile device or any website as per the developer's need.

2.1 challenges

Image classification is one of the major task because of its data size, handling a image data is not an easy game. Image classification is where a computer can analyse an image and identify the 'class' the image falls under. (Or a probability of the image being part of a 'class'.) A class is essentially a label, for instance, 'car', 'animal', 'building' and so on. For example, you input an image of a sheep. Simply put, image classification is where machines can look at an image and assign a (correct) label to it. It's a key part of computer vision, allowing computers to see the world as we do. And with the invention of deep learning, image classification has become more widespread.
Deeper exploration into image classification and deep learning involves understanding convolutional neural networks. But for now, you have a simple overview of image classification and the clever computing behind it.

The Challenges of Deep Learning

- Learning without Supervision. ...
- Coping with data from outside the training distribution. ...
- Incorporating Logic. ...
- The Need for less data and higher efficiency. ...
- Attention and Transformers. ...
- Unsupervised and self-supervised learning. ...
- Generative Adversarial Networks (GANs) ...
- Auto-encoders.

2.2 Different techniques

Image classification is probably the most important part of digital image analysis. It uses AI-based deep learning models to analyze images with results that for specific tasks already surpass human-level accuracy, The 3 main types of image classification techniques in remote sensing are: Unsupervised image classification. Supervised image classification. Object-based image analysis.
Different techniques to classify an images are

Machine Learning (ML) image classification

1. Artificial Neural Network.
2. Convolutional Neural Network.
3. K nearest neighbor.
4. Decision tree.
5. Support Vector Machines.

In the classification, six different modalities, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), -nearest neighbour (NN), the Naïve Bayes approach, support vector machine (SVM), and artificial neural networks (ANN), were utilized.

2.3 CNN

In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks.

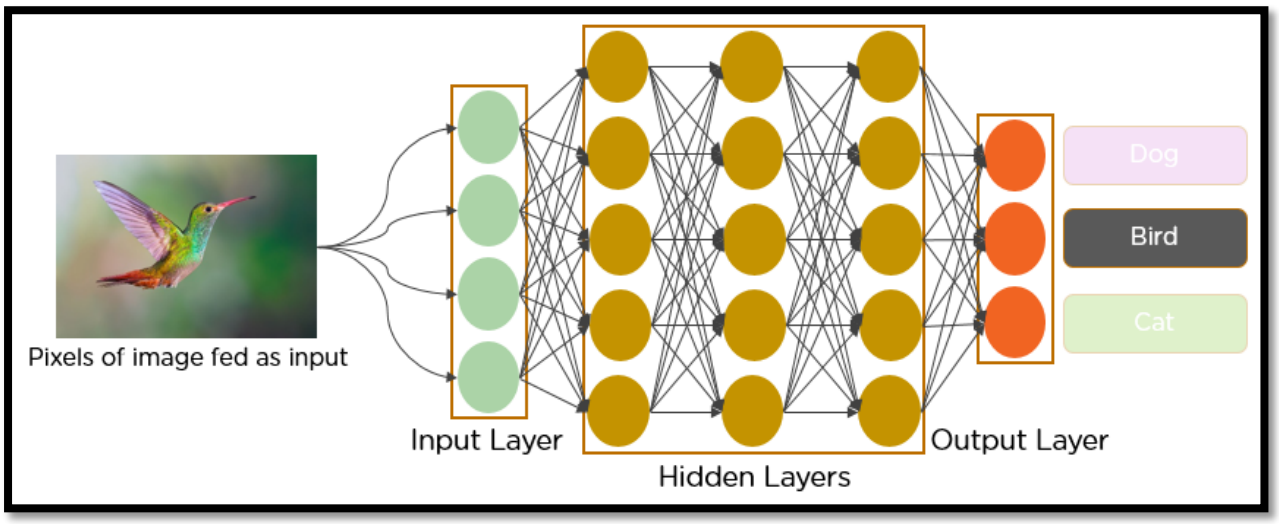

Fig. 2.1: Layers of CNN

Since the 1950s, the early days of AI, researchers have struggled to make a system that can understand visual data. In the following years, this field came to be known as Computer Vision. In 2012, computer vision took a quantum leap when a group of researchers from the University of Toronto developed an AI model that surpassed the best image recognition algorithms and that too by a large margin.

Background of CNNs

CNN's were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

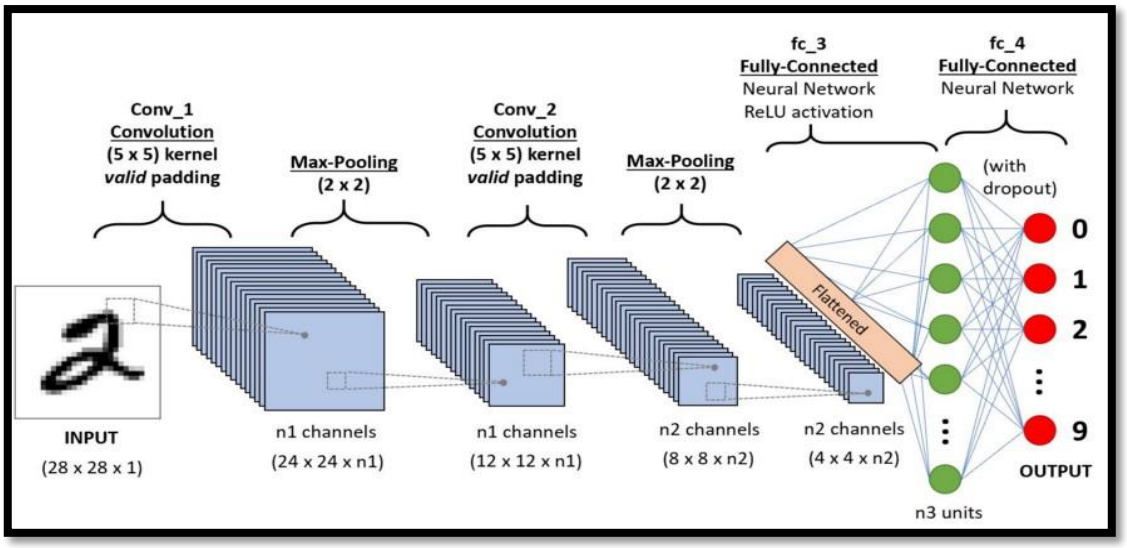

Fig. 2.2: Layering

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

Cnn working

This article was published as a part of the Data Science Blogathon. In 2012 Alex Krizhevsky realized that it was time to bring back the branch of deep learning that uses multi-layered neural networks. The availability of large sets of data, to be more specific ImageNet datasets with millions of labeled images and an abundance of computing resources enabled researchers to revive CNNs.

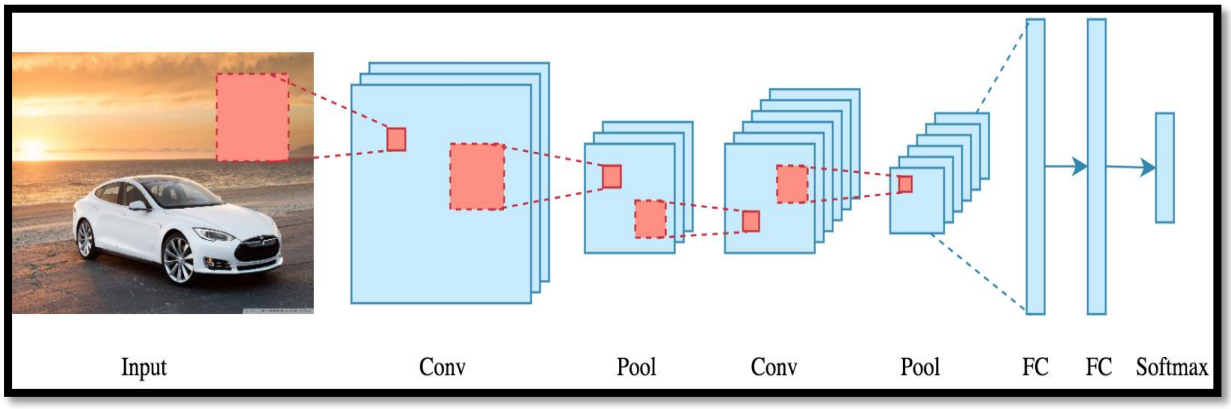

Fig.2.3: Working

But we don't really need to go behind the mathematics part to understand what a CNN is or how it works.

Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

How does it work?

Before we go to the working of CNN's let's cover the basics such as what is an image and how is it represented. An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same but it has a single plane. Take a look at this image to understand more.

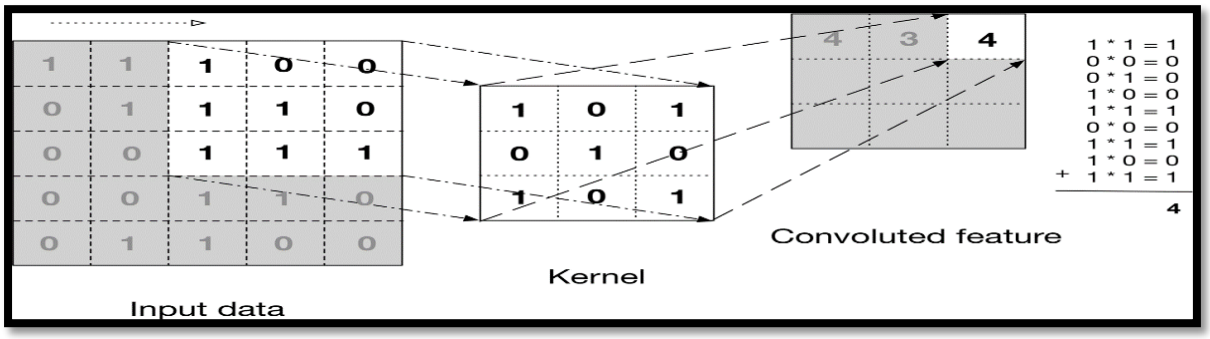

Fig.2.4: Applying Kernal

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

KERAS IN CNN

A great way to use deep learning to classify images is to build a convolutional neural network (CNN). The Keras library in Python makes it pretty simple to build a CNN. Computers see images using pixels. ... This dataset consists of 70,000 images of handwritten digits from 0–9. We will attempt to identify them using a CNN.

Prediction in cnn

They have proven so effective that they are the ready to use method for any type of prediction problem involving image data as an input. The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image, Keras is used for creating deep models which can be productized on smartphones. Keras is also used for distributed training of deep learning models. Keras is used by companies such as Netflix, Yelp, Uber, etc

Chapter 3
Work Carried Out

Architecture

This below architecture of our project shows that we extracted the key features from the raw data and then split it into a training dataset and testing dataset. The machine learning model then trained using the training dataset and tested using the testing dataset. The model then predicts the match outcome.

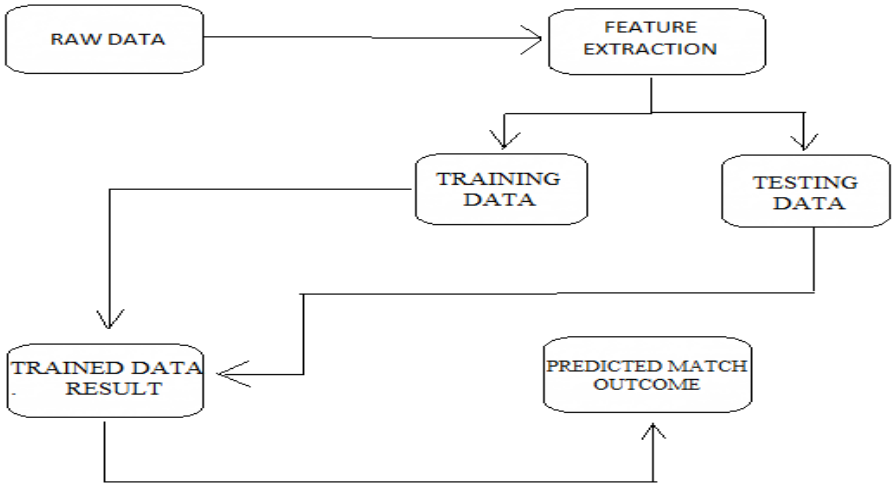

Figure 3.1 Architecture

Step by step approach:

Step 1: Getting the Dataset

The dataset is available on kaggle.
(!wget https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_3367a.zip)

Unzip the Dataset

!unzip kagglecatsanddogs_3367a.zip

Step 2: Installing Required Packages [Python 3.6]

1. OpenCV --> '3.4.0' [Used to handle image operations like reading the image, resizing, reshaping]
2. numpy --> '1.14.4' [Image that is read will be stored in an numpy array]
3. TensorFlow --> '1.8.0' [Tensorflow is the backend for Keras]
4. Keras --> '2.1.6' [Keras is used to implement the CNN]

Step 3: How the Model Works??

The dataset contains a lot of images of cats and dogs. Our aim is to make the model learn the distinguishing features between the cat and dog. Once the model has learned, i.e once the model got trained, it will be able to classify the input image as either cat or a dog.

Features Provided:

Own image can be tested to verify the accuracy of the model
This code can directly be integrated with your current project or can be extended as a mobile application or a site.
To extend the project to classify different entities, all you need to do is find the suitable dataset, change the dataset accordingly and train the model

Plot Dog and Cat Photos

Looking at a few random photos in the directory, you can see that the photos are color and have different shapes and sizes.

For example, let's load and plot the first nine photos of dogs in a single figure

```
[ ]  # to display grid of images
     plt.figure(figsize=(25,25))
     temp = df[df['label']==1]['images']
     start = random.randint(0, len(temp))
     files = temp[start:start+25]

     for index, file in enumerate(files):
         plt.subplot(5,5, index+1)
         img = load_img(file)
         img = np.array(img)
         plt.imshow(img)
         plt.title('Dogs')
         plt.axis('off')
```

Fig. 3.2: Code for display dog data

Fig. 3.3: Dog Data

Running the example creates a figure showing the first nine photos of dogs in the dataset.

We can see that some photos are landscape format, some are portrait format, and some are square

```
# to display grid of images
plt.figure(figsize=(25,25))
temp = df[df['label']==0]['images']
start = random.randint(0, len(temp))
files = temp[start:start+25]

for index, file in enumerate(files):
    plt.subplot(5,5, index+1)
    img = load_img(file)
    img = np.array(img)
    plt.imshow(img)
    plt.title('Cats')
    plt.axis('off')
```

Fig. 3.4: code for display cat data

Fig. 3.5: Cat Data

Again, we can see that the photos are all different sizes.

We can also see a photo where the cat is barely visible (bottom left corner) and another that has two cats (lower right corner). This suggests that any classifier fit on this problem will have to be robust.

Select Standardized Photo Size

The photos will have to be reshaped prior to modeling so that all images have the same shape. This is often a small square image.

There are many ways to achieve this, although the most common is a simple resize operation that will stretch and deform the aspect ratio of each image and force it into the new shape.

We could load all photos and look at the distribution of the photo widths and heights, then design a new photo size that best reflects what we are most likely to see in practice.

Smaller inputs mean a model that is faster to train, and typically this concern dominates the choice of image size. In this case, we will follow this approach and choose a fixed size of 200×200 pixels

Pre-Process Photo Sizes

If we want to load all of the images into memory, we can estimate that it would require about 12 gigabytes of RAM.

That is 25,000 images with 200x200x3 pixels each, or 3,000,000,000 32-bit pixel values. We could load all of the images, reshape them, and store them as a single NumPy array. This could fit into RAM on many modern machines, but not all, especially if you only have 8 gigabytes to work with. We can write custom code to load the images into memory and resize them as part of the loading process, then save them ready for modeling.

The example below uses the Keras image processing API to load all 25,000 photos in the training dataset and reshapes them to 200×200 square photos. The label is also determined for each photo based on the filenames. A tuple of photos and labels is then saved.

Pooling

The pooling operation provides spatial variance making the system capable of recognizing an object with some varied appearance. It involves adding a 2Dfilter over each channel of the feature map and thus summarise features lying in that region covered by the filter.

So, pooling basically helps reduce the number of parameters and computations present in the network. It progressively reduces the spatial size of the network and thus controls over fitting. There are two types of operations in this layer; Average pooling and maximum pooling. Here, we are using max-pooling which according to its name will only take out the maximum from a pool. This is possible with the help of filters sliding through the input and at each stride, the maximum parameter will be taken out and the rest will be dropped.

The pooling layer does not modify the depth of the network unlike in the convolution layer.

Chapter 4
Experimental Results and Comparison

The Dogs vs. Cats dataset is a standard computer vision dataset that involves classifying photos as either containing a dog or cat.
Although the problem sounds simple, it was only effectively addressed in the last few years using deep learning convolutional neural networks. While the dataset is effectively solved, it can be used as the basis for learning and practicing how to develop, evaluate, and use convolutional deep learning neural networks for image classification from scratch.

```
]: import random
   num=random.randint(0,len(Xtrain))
   pt.imshow(Xtrain[num,:])

]: <matplotlib.image.AxesImage at 0x2878f92c730>
```

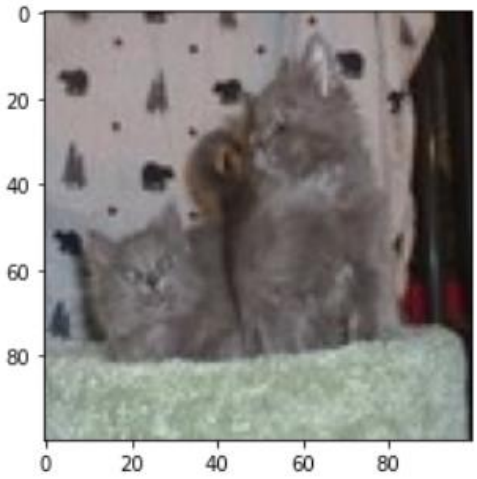

Fig. 4.1 X train data

This includes how to develop a robust test harness for estimating the performance of the model, how to explore improvements to the model, and how to save the model and later load it to make predictions on new data.
In this tutorial, you will discover how to develop a convolutional neural network to classify photos of dogs and cats. After completing this tutorial, you will know: How to load and prepare photos of dogs and cats for modeling.
How to develop a convolutional neural network for photo classification from scratch and improve model performance. How to develop a model for photo classification using transfer learning.

```
num=random.randint(0,len(ytest))
pt.imshow(Xtest[num,:])
pt.show()
```

Fig. 4.2 X test data

```
2]: ypr=model.predict(Xtest[num,:].reshape(1,100,100,3))
    print (ypr)

    [[0.5204475]]

3]: ypr=ypr>0.5
    if ypr==1:
        print("dog")
    else:
        print("cat")

    dog
```

Fig. 4.3 Prediction

Visualization

```
[ ]  acc = history.history['accuracy']
     val_acc = history.history['val_accuracy']
     epochs = range(len(acc))

     plt.plot(epochs, acc, 'b', label='Training Accuracy')
     plt.plot(epochs, val_acc, 'r', label='Validation Accuracy')
     plt.title('Accuracy Graph')
     plt.legend()
     plt.figure()

     loss = history.history['loss']
     val_loss = history.history['val_loss']
     plt.plot(epochs, loss, 'b', label='Training Loss')
     plt.plot(epochs, val_loss, 'r', label='Validation Loss')
     plt.title('Loss Graph')
     plt.legend()
     plt.show()
```

Fig. 4.4 Code for visualization

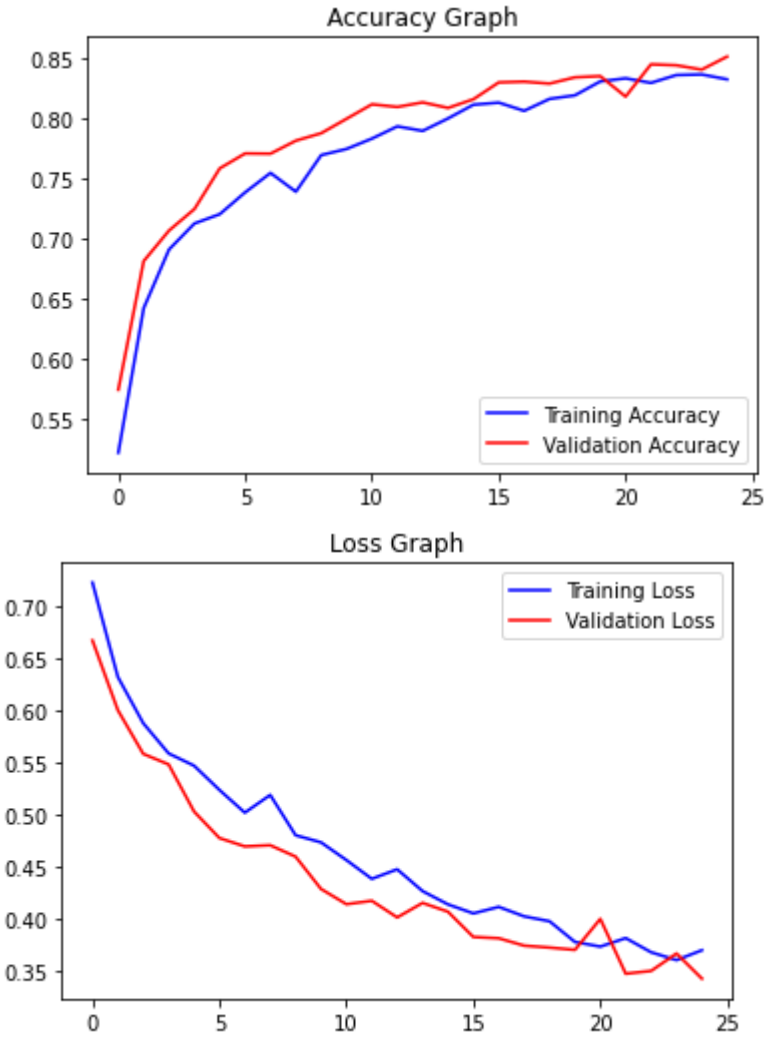

Fig. 4.5: Loss and Accuracy graph

Chapter 5
Summary &Conclusions

Summary

In this report, we first briefly explained our motivation of this project and showed some background materials. Then, we precisely illustrated our task, including the learning task and the performance task. After that, we introduced our solution in detail, mainly including two approaches. The first approach is a traditional pattern recognition model, by which we learned the classification model from some human-crafted features, mainly including color feature, Dense-SIFT feature, and a combination of the two. To improve the performance, we also applied image segmentation approach to preprocess the data. However, due to poor segmentation result, we did not achieve any improvement. The best accuracy we got from the first method is only 71.47% (from an SVM classifier).

To achieve better performance, we implemented our second approach, which is a trainable model that applies the CNN to learn features. We also looked insight into what Deep Networks learned from images and explained why they achieve good performance. The highest accuracy of this approach is 94.00% (from an SVM classifier), which is also our best result and helps us rank 9th in 91 teams in the Kaggle competition.

In terms of classifiers, we mainly considered SVMs and BP Neural Networks, taking our high dimensional feature space into account. Various parameter settings were explored to improve classification accuracy on the test dataset. For example, for the BP Neural Networks, we tried different hidden layers and hidden units; for the SVMs, different kernel functions and C parameters were used. Table 1 illustrates the best results of each model and related parameters.

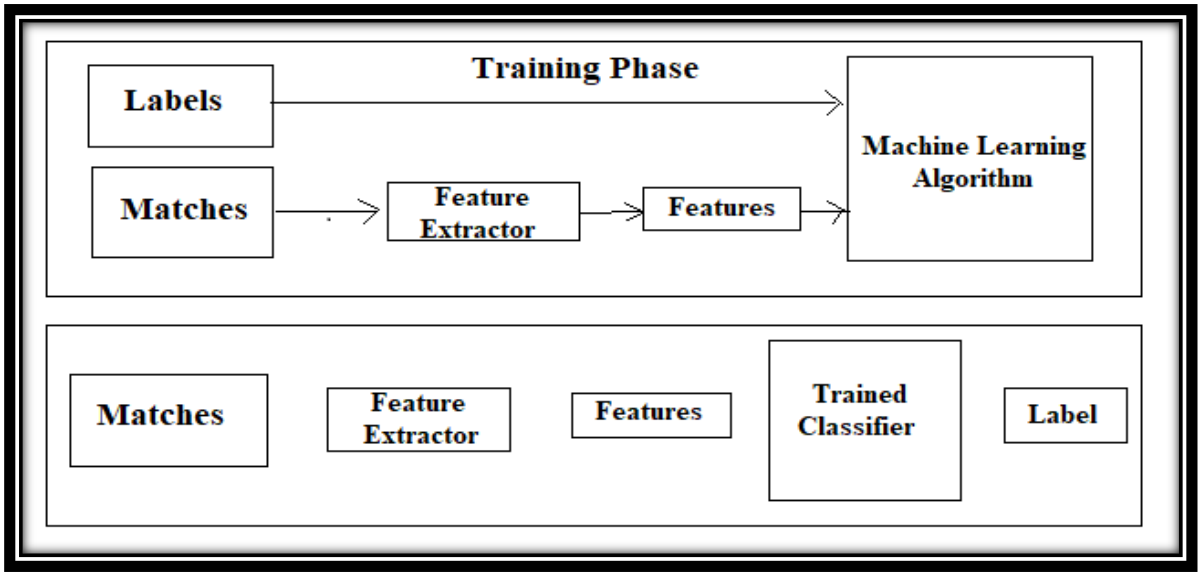

Figure 5.1 Process Flow

Conclusions

In this paper we build a deep convolutional neural network for image classification (cat and dog images). Despite of using only a subset of the images and accuracy of 88.10% was obtained. If the whole dataset was being used the accuracy would have been even better.

This paper shows a classification process using convolutional neural networks which is a deep learning architecture. Although there are many algorithms that perform image classification, convolutional neural network is considered to be a standard image classification technique. Convolutional neural network uses GPU technology because of large number of layers which increases the number of computers. Therefore, in this paper, we presented a very small convolutional neural network which can work on CPU as well. This network classifies the images into one the two predefined classes say Cat and Dog. This same network can be used for other datasets as well.

References

1) Golle, P. (2008, October). Machine learning attacks against the Asirra CAPTCHA. In Proceedings of the 15th ACM conference on Computer and communications security (pp. 535-542). ACM.

2) J. Elson, J. Douceur, J. Howell and J. Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. Proc. of ACM CCS 2007, pp. 366-374.

3) Muthukrishnan Ramprasath, M.Vijay Anand and Shanmugasundaram Hariharan, Image Classification using Convolutional Neural Networks. International Journal of Pure and Applied Mathematics, Volume 119 No. 17 2018, 1307-1319

4) Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012, June). Cats and dogs. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 3498-3505). IEEE.

5) Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Neural Networks. arXiv preprint arXiv:1311.2901.

6) [6] Bang Liu, Yan Liu, Kai Zhou "Image Classification for Dogs and Cats"

7) Tianmei Guo, Jiwen Dong ,Henjian Li'Yunxing Gao, "Simple Convolutional Neural Network on Image Classification" , IEEE 2nd International Conference on Big Data Analytics at Beijing, China on 10-12 March 2017

8) Emine CENGIL , Ahmet ÇINAR , Zafer GÜLER, "A GPU-Based Convolutional Neural Network Approach for Image Classification", International Conference on Artificial Intelligence and Data Processing Symposium at Malatya, Turkey on 16- 17 September 2017.

9) Travis Williams, Robert Li, "Advanced Image Classification using Wavelets and Convolutional Neural Networks", 15th International Conference on Machine Learning and Applications at Anaheim, CA, USA on 18-20 December 2016.

10) Sayali Jog , Mrudul Dixit, "Supervised Classification of Satellite Images", Conference on Advances on signal processing at Pune, India on 9-11 June 2016.

11) ER-XIN SHANG, HONG-GANG ZHANG, "Image Spam Classification Based On Convolutional Neural Network", International Conference on Machine Learning and Cybernetics at Jeju, South Korea on 10-13 July 2016.

12) Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng and Mei Chen, “Medical Image Classification with Convolutional Neural Network”, 13th International Conference on Control Automation Robotics and Vision at Singapore on 10-12 December 2014.

13) Mart´in Abadi et.al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.