learning provide a very better solution for different image classifications. But in thistype of project when there is a need to rebuild the project from the scratch it is difficult to start because it requires a lot of time and cost by using CNN, so we use transfer learning.

Hiroki Watabe et al [14] and Hiroki Watanabe et al [15] in the year 2017 implemented CNN by using Data Augmentation and Discriminative Generative Adversaria Network (DCGAN). Proposed Data Augmentation is more effective when compared to standard Augmentation. The major drawback is it generates both realistic and unrealistic images. The proposed approach was to apply DCGAN to CNN Data Augmentation. DCGAN has both a generator and a discriminator to create images. From this method, we can generate many more images and increase the efficiency of the CNN model. Using this model, we can achieve an accuracy of 82.7%.

2.1 INFERENCES FROM LITERATURE SURVEY

From the literature survey, Traditional classification makes the classification or identification less effective. To overcome this problem, we need to use pre-trained Convolutional Neural Networks for better efficiency and accuracy. CNN is a class of deep learning and gives good accuracy when there is a large dataset. When Convolutional Neural Networks work with different pre-trained models such as Xception, Augmentation, and Transfer Learning, and different layers of CNN such as RESNET18, VGG16, and DenseNet161 give us better results. From this, I came to know that CNN is the best method for image classification and they are known for their ability to reduce the computational time and adapt to different variations of images.

50 epochs achieved an accuracy of 82.36% from the Densenet161model. The major drawback is by using VGG16 overfitting occurs.

In Y. Liuetet al [7] all research paper, the authors perform domain adaptation, a branch of transfer learning, to adapt the data distributions of source and target so that the classification could be more efficient in a cross-subject scenario.

In this research paper, Zalan Raduly et al [8], Csaba Sulyok et al [9], Zsolt vadaszi et al [10], and Attila Zoldc et al [11] in the year 2018 implement a fine-grained image recognition problem. Used multiclass classification, Inception ResNetv2, mobile trained model. The dataset is taken from the Stanford dog dataset. The training data is split into train and validation folds. Fine-tuning and 5 – fold cross validation was also used in this project. This project consists of various experiments in CNN architectures, Data Augmentation, Learning and hyperparameters, and Frozen graphs. Accuracy, precision, recall, and confusion matrices were used to predict the accuracy of different methods and take the best one. Used one of the software called "sniff" for the prediction of trained Convolutional neural networks. Two different CNN architectures were used in this project. One is NASNET-A mobile architecture and Inception ResNet v2 deep architecture. By using Inception ResNetv2 we get an accuracy of 90.69%.

Vijaya Kumar et al [12] and B. Bhavya et al [13] focused on fine-tuning pre-trained models in the year 2019. This project was implemented by using Image classification, Transfer learning, Convolutional Neural Networks, Vgg16, Xception, and Inception V3. Finally, a multi-class classifier named logistic regression was used to identify the breed of the dog. The major drawback is by using CNN we need to have a large amount of dataset and images to reduce this drawback they used transfer learning to train the model to provide the best solution for this. The results that are produced by transfer learning are better than the results produced by Inception v3. Because a Multinomial linear classifier was applied to pre-train the model in transfer learning. From this project, we can conclude that Convolutionalneural networks with transfer

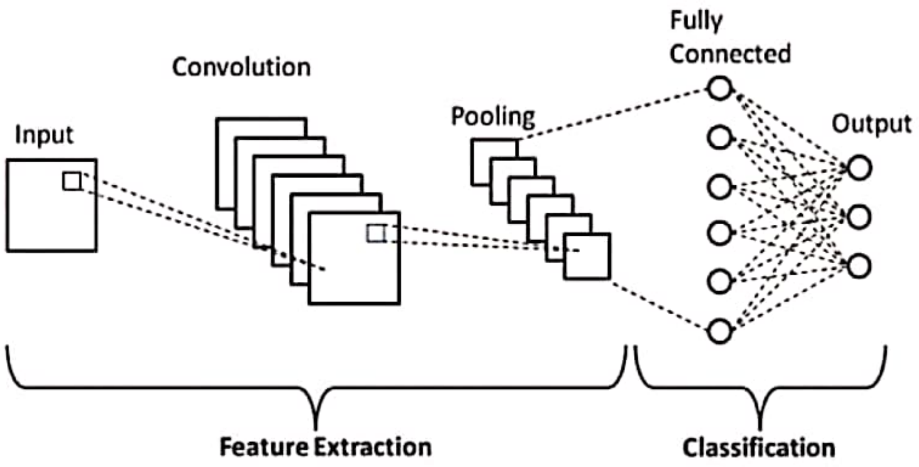
CHAPTER 2

LITERATURE SURVEY

This problem statement has been extensively studied over the past 5 years by researchers and automotive companies in a bid to create a solution, and all their solutions vary from analyzing the Dog breed identification using different methods some of the research was as follows.

The work of K. Mulligan et al [1] and P. Rivas et al [2] in the year 2019, July, conducted dog breed identification with the help of Xception Convolutional Neural Network architecture. This paper is mainly focused on classification tools. The dataset is downloaded from Kaggle. This classification is worked on CNN and Xception with multilayer perceptron. The methods used for Xception and MLP. Experimented on 120 unique breeds over 10,200 images of dogs. From this project, a confusion matrix was created over training and test set. Its major drawbacks were generating a diagonal pattern and the values incorrectly predicted. The methods were not passed through cross-validation. Achieved accuracy of 54.80 %. Later with the performance matrix, changing and increasing the number of splits utilized by both LogLoss and balance accuracy. After doing this achieved the correct prediction of describing the image belonging to which type of breed.

The work performed by Wenting Shi et al [3], Jiaquan Chen et al [4], Muyun Liu et al [5], and Fangyu Liu et al [6] in the year 2019 mostly focused on pattern recognition of which object belongs. This project is based on image or pattern recognition of identifying the dog's breed. Four models were used in this project such as ResNet18, VGG16, DenseNet161, and AlexNet. Data Augmentation was used to increase the number of training data parameters. Conducted some experiments by using Data Augmentation, Transfer Learning, Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), and parameter tuning. By comparing the four models Densenet161 gives the best accuracy. Comparing Loss Analysis, and Accuracy Analysis by designing a model using



1.1 BASIC CNN ARCHITECTURE

Dogs are considered to be man's best friend and they act as the best pet. They can be easily trained and used as service dogs to help handicapped persons and guide them. They are also used for military purposes. There were almost 350 dog breeds all over the world. They require care, food, and a habitat environment depending on the breed and for this reason, we are going to identify the breed of the dog. Our project is to identify different dogs from different breeds. Our dataset is downloaded from Kaggle: Dog Breed Identification. There isa total of 20,000+ images of dogs in our dataset.We separate our dataset into two parts for training and testing. Since our dataset is too large, we use Keras and TensorFlow to build, train and test a convolutional neural network capable of identifying the breed of a dog in a supplied image.

CHAPTER 1

INTRODUCTION

In the past few years, Deep learning has been focused on many fields, such as image processing and natural language process. Particularly Convolutional Neural Networks (CNN) increased high accuracy in the image processing field. This Convolutional Neural Network has become most famous for image classification problems. When compared to other classification techniques, CNN has unique features that start from raw data and thereby apply the modeling technique. CNN works like human vision, firstly we can classify things using our vision. We can present CNN in the form of an algorithm by building a mathematical model. Convolutional neural networks are effective in analyzing visual imagination. They use multi-layer perceptrons and require less preprocessing when compared with other image classification algorithms. CNN has at least one fully connected layer preceded by the desired number of fully Convolutional layers as a standard multi-layered network. One of the benefits of CNN is that it is translation invariant, also referred to as shift invariant or space invariant artificial neural networks.

There are three main layers used in the CNNs, namely the convolutional layer, pooling layer, and fully connected layer, (CONV-POOL-FC). Each layer can be repeated a required number of times to accomplish the desired output. CNN's divide the images into smaller parts/features and match them individually.

- The input image comprises raw pixel values represented in a matrix format (m x m x r, m rows, n columns, r channels, for an RGB image r=3).
- CONV layer computes the output of neurons connected to the local region by computing the dot product of the sub-image and k-filters (size - n x n x q) and finding the average value to obtain k-filtered images.
- POOL layer performs a downsampling operation along the spatial dimensions i.e., it takes the stack of filtered images and reduces the size of the image matrix to give an optimal output image.
- FC layer will compute the class scores, resulting in a volume of size [1x1x a]

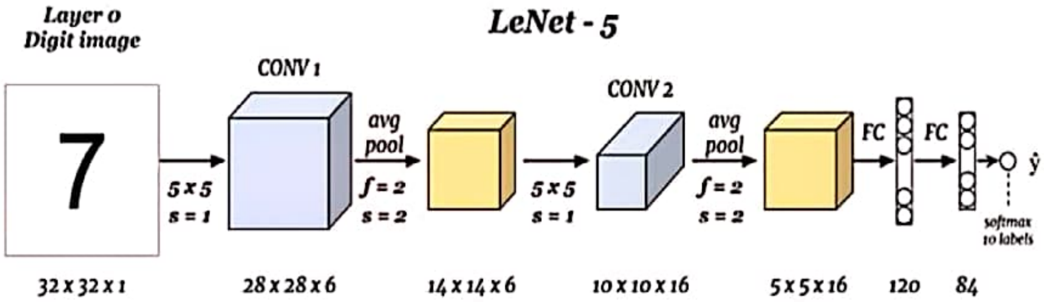
4. Layer 4: Again, similar to Layer 2, this layer is a pooling layer with 16 filters this time around. Remember, the outputs are passed through the sigmoid activation function. The input of size 10x10x16 from the previous layer gets sub-sampled to 5x5x16. Total params in layer = (1 + 1) * 16 = 32.

5. Layer 5: This time around we have a convolutional layer with a 5×5 kernel size and 120 filters. There is no need to even consider strides as the input size is 5x5x16 so we will get an output of 1x1x120. Total params in layer = 5 * 5 * 120 = 3000

6. Layer 6: This is a dense layer with 84 parameters. So, the input of 120 units is converted to 84 units. Total params = 84 * 120 + 84 = 10164. The activation function used here was rather a unique one.

7. Output Layer: Finally, a dense layer with 10 units is used. Total params = 84 * 10 + 10 = 924.

3.4.1 LeNet-5 Convolutional Neural Network



3.5 *LAYERS OF LeNet-5 ARCHITECTURE*

It is a seven layers Convolutional Neural Network and each layer has specific dimensions.

1. Layer 1: A convolutional layer with a kernel size of 5×5, a stride of 1×1, and 6 kernels in total. So, the input image of size 32x32x1 gives an output of 28x28x6. Total params in layer = 5 * 5 * 6 + 6.

2. Layer 2: A pooling layer with 2×2 kernel size, a stride of 2×2, and 6 kernels in total. This pooling layer acted a little differently than what we discussed in the previous post. The input values in the receptive were summed up and then multiplied to a trainable parameter (1 per filter), and the result was finally added to a trainable bias (1 per filter). Finally, sigmoid activation was applied to the output. So, the input from the previous layer of size 28x28x6 gets sub-sampled to 14x14x6. Total params in layer = [1 (trainable parameter) + 1 (trainable bias)] * 6 = 12

3. Layer 3: Similar to Layer 1, this layer is a convolutional layer with the same configuration except it has 16 filters instead of 6. So, the input from the previous layer of size 14x14x6 gives an output of 10x10x16. Total params in layer = 5 * 5 * 16 + 16 = 416.

b. Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH, and Sigmoid functions. Each of these functions has a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred and for multi-class classification, generally, softmax is used. In simple terms, activation functions in a CNN model determine whether a neuron should be activated or not. It decides whether the input to the work is important or not to predict using mathematical operations.

3.4 LeNet-5 CNN ARCHITECTURE WITH KERAS AND TENSERFLOW

Keras is a high-level python API that can be used to quickly build and train neural networks using either Tensorflow or Theano as the backend.Keras is one of the fastest-growing libraries for deep learning.The main data structure in Keras is the model which provides a way to define the complete graph. You can add layers to the existing model/graph to build the network you want.

Keras has two distinct ways of building models:

I. **Sequential models:** This is used to implement simple models. You simply keep adding layers to the existing model.

II. **Functional API:** Keras functional API is very powerful and you can build more complex models using it, models with multiple outputs, directed acyclic graphs, etc.

c. Fully Connected Layer

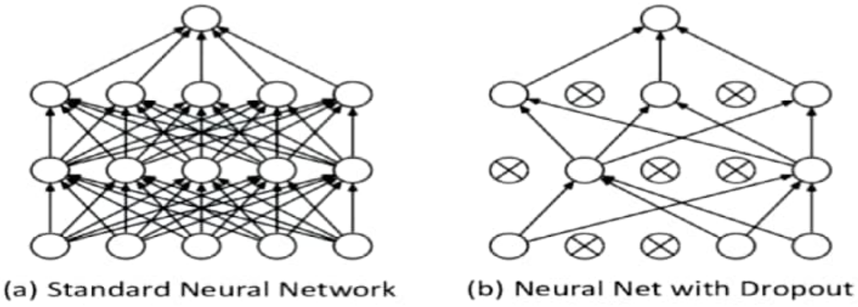
The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.In this, the input image from the previous layers is flattened and fed to the FC layer. The flattened vector then undergoes a few more FC layers where the mathematical function's operations usually take place. In this stage, the classification process begins to take place. The reason two layers are connected is that two fully connected layers will perform better than a single connected layer. These layers in CNN reduce human supervision.

Some important parameters in CNN architecture are

a. Dropout

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data harming the model’s performance when used on new data.To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during the training process resulting in a reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

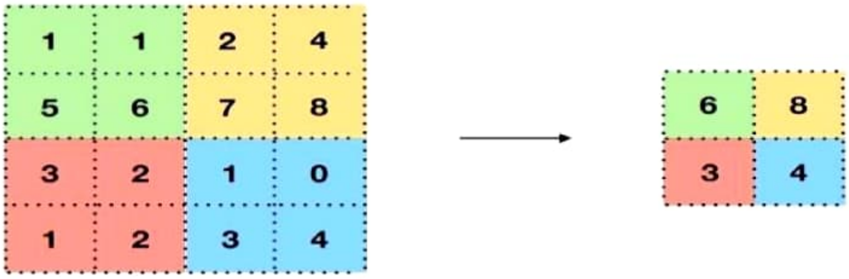
Dropout results in improving the performance of a machine learning model as it prevents overfitting by making the network simpler. It drops neurons from the neural networks during training.



3.4 EXAMPLE OF DROPOUT

map. Depending upon the method used, there are several types of Pooling operations. It summarises the features generated by a convolution layer.In Max Pooling, the largest element is taken from the feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

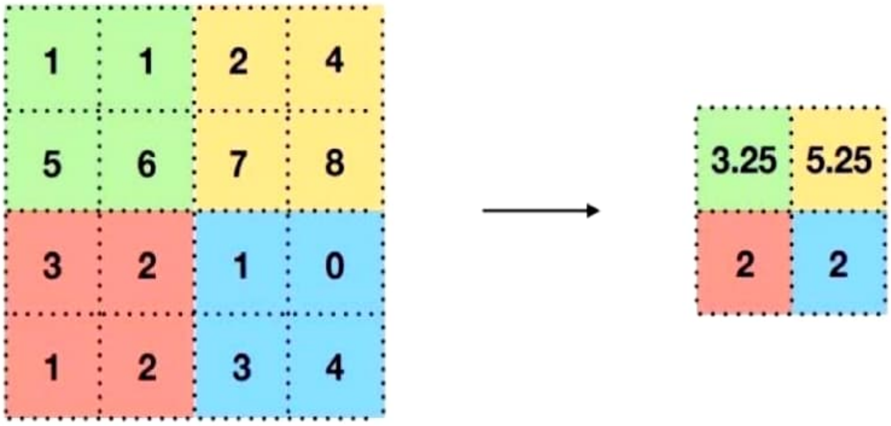
This CNN model generalizes the features extracted by the convolution layer and helps the networks to recognize the features independently. With the help of this, the computations are also reduced in a network.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

3.2 MAX POOLING EXAMPLE

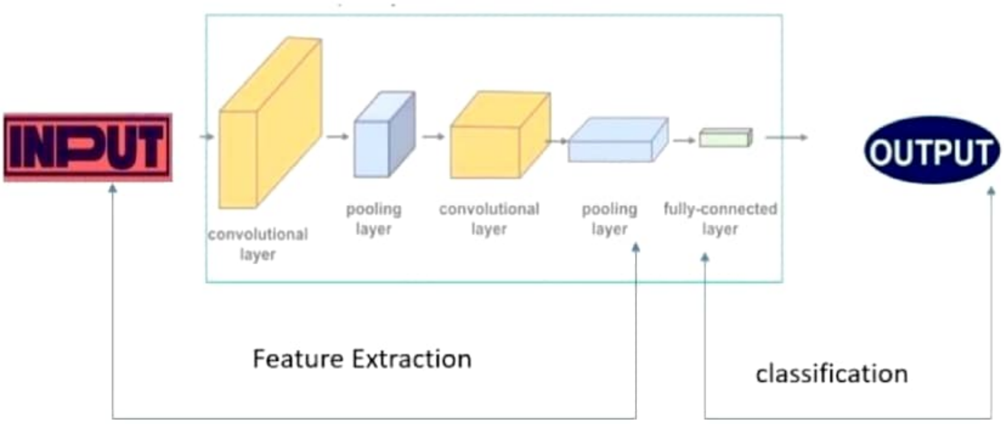


1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

3.25	5.25
2	2

3.3 AVERAGE POOLING EXAMPLE

CNN ARCHITECTURE



3.1 CONVOLUTIONAL NEURAL NETWORK

Types of CNN layers

a. Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product istaken between the filter and the parts of the input image concerning the size of the filter (MxM). The output is termed the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.
The convolution layer in CNN passes the result to the next layer once applying the convolution operation in the input. Convolutional layers in CNN benefit a lot as they ensure the spatial relationship between the pixels is intact.

b. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce computational costs. This is performed by decreasing the connections between layers and independently operating on each feature

3.3 METHODS AND ALGORITHM USED:

Since this is a multi-class classification problem under deep learning with image recognition. We are going to use the CNN algorithm with Keras and TensorFlow methods. As classification is a supervised learning approach in which the computer program learns from the data input and images and uses these learnings to classify new observations. Some examples of classification problems are speech recognition, handwriting recognition, biometric identification, document classification, etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on the pattern and associates the patterns with the unlabeled new data.

3.3.1 BASIC CNN ARCHITECTURE:

CNN's are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision, and natural language processing.CNN has high accuracy, and because of the same, it is useful in image recognition. Image recognition has a wide range of uses in various industries such as medical image analysis, phone, security, recommendation systems, etc.

Three types of layers make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function.

ENVIRONMENTAL REQUIREMENTS:

1. Software Requirements:

   Operating system: Windows

   Tool: Google Colab, spyder, MS Excel 2013

2. Hardware Requirements:

   A. Internet connection to download and activate.

   B. Minimum 10GB of free disk space

   C. Windows 8.1 or 10 (64-bit version only) is required.

   D. Minimum System Requirements

   To run Office Excel 2013, your computer needs to meet the following minimum hardware requirements:

   - 500-megahertz (MHz)
   - 1.5 gigabytes (GB) available space
   - 1024x768 or higher resolution monitor

3.2 PROBLEM STATEMENT:

Dogs are considered to be man's best friend and they act as the best pet. Now pet industry has increased a lot. Many people were interested to keep a pet like dogs in their homes. Some people may not be able to identify the breed of a dog by simply looking at it. So, the only way to find the breed of a dog is to ask the owner or some professionals in the industry. This takes a lot of time and makes a lot of confusion. So, for this purpose, we are going to build a project to identify the breed of a dog.

CHAPTER 3

EXPERIMENTAL MATERIALS AND ANALYSIS

3.1 MATERIALS OR REQUIREMENTS:

Requirements are the basic constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environmental requirements
   A. Hardware requirements
   B. Software requirements

FUNCTIONAL REQUIREMENTS:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists the requirements of a particular software system.

NON-FUNCTIONAL REQUIREMENTS:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction of the result

2.2 OPEN PROBLEMS IN THE EXISTING SYSTEM

Some of the drawbacks or problems with the existing system are:

- By using VGG16 in the model overfitting may occur and gives incorrect predictions.
- While using Convolutional Neural Networks the dataset must be bigger, if not we come across less accuracy.
- By using Data Augmentation, it produces both realistic as well as unrealistic images.
- While using the Xception method for smaller datasets sometimes the predicted outcomes may become wrong.

2.3 OVERALL OBJECTIVE:

This project examines how to use Keras and TensorFlow to build, train and test a Convolutional Neural Network capable of identifying the breed of a dog in a supplied image. By using different methods such as LeNet-5, VGG16, and Xception, by predicting the model and finding the accuracy of the model. Later on, building a website with flask which will be more user-friendly for the end user to identify the breed of the dog.

REFERENCES

- Mulligan, K. and Rivas, P., 2019, July. Dog breed identification with a neural network over learned representations from the xception CNN architecture. In *21st International conference on artificial intelligence (ICAI 2019).*

- Z. Ráduly, C. Sulyok, Z. Vadászi and A. Zölde, "Dog Breed Identification Using Deep Learning," 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY), 2018, pp. 000271-000276, doi: 10.1109/SISY.2018.8524715.

- Shi, Wenting, Jiaquan Chen, Muyun Liu, and Fangyu Liu. "Dog Breed Identification." (2018).

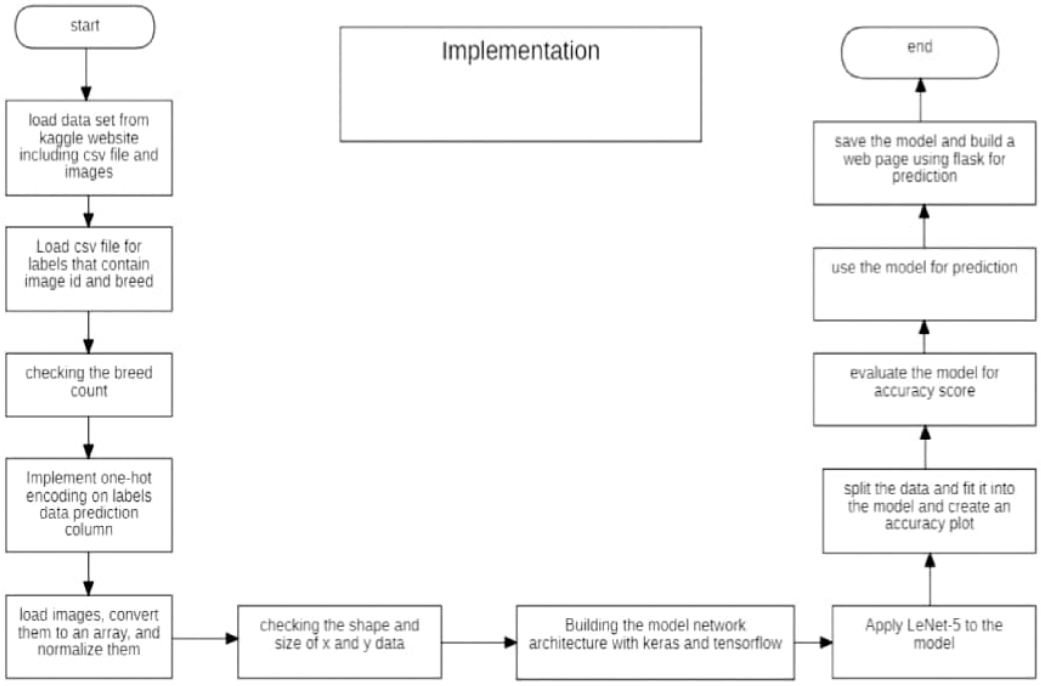
- B. Vijaya Kumar, K. Bhavya., Dog Breed Identification with Fine tuning of pre-trained models. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S11, September 2019.

- Watabe, Hiroki and Hiroshi Watanabe. “USING DCGAN FOR BREED IDENTIFICATION.” (2017).

- M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, “A High-Speed and Low-Complexity Architecture for Softmax Function in Deep Learning,” 2018 IEEE Asia Pacific Conf. Circuits Syst. APCCAS 2018, pp. 223-226, 2019, doi: 10.1109/APCCAS.2018.8605654.

- Split the dataset into training and test datasets and create accuracy and value accuracy of the model and plot the accuracy model between them.
- Predicting the accuracy score of the model and we take the best one as our model and save it.
- We use the best model for the prediction of dog breeds.
- By saving the model, creating some HTML files with python code using the flask application, and creating a web application for the prediction of dog breeds.

4.3 PROJECT MANAGEMENT PLAN:

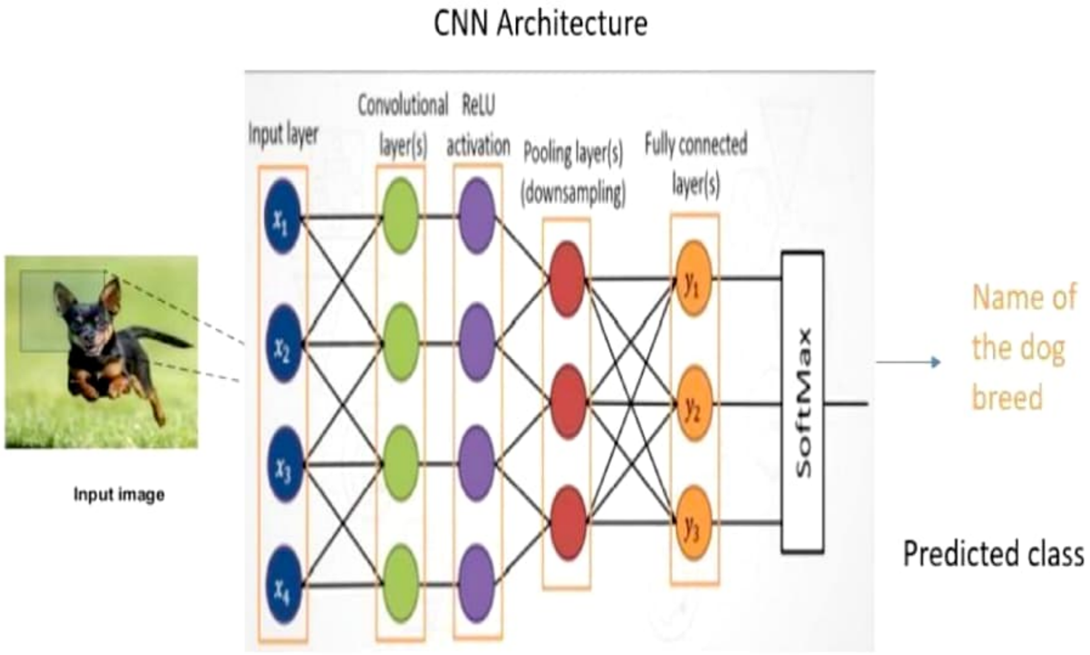

start
load data set from kaggle website including csv file and images
Load csv file for labels that contain image id and breed
checking the breed count
Implement one-hot encoding on labels data prediction column
load images, convert them to an array, and normalize them
checking the shape and size of x and y data
Building the model network architecture with keras and tensorflow
Apply LeNet-5 to the model
split the data and fit it into the model and create an accuracy plot
evaluate the model for accuracy score
use the model for prediction
save the model and build a web page using flask for prediction
end
Implementation

4.2 PROJECT MANAGEMENT PLAN

- we are collecting the data from the Kaggle platform. We require dataset dog breeds with images of each dog breed in distinguishing proof just as in picture order. It consists of 133 different types of dog breeds and consists of 20,400 images of dogs.
- In this, we will import necessary modules such as NumPy, pandas, Keras, etc., and read the CSV file, check the first rows of a dataset and count the number of breeds in the CSV file.
- Load images from the dataset and the images are then converted into a NumPy array and finally normalizing the array.
- Here we use Convolutional Neural Network architecture for image classification because in recent years it earned much popularity inthe image processing field. Here we are going to study different layers such as the Convolutional layer, Pooling layer, fully connected layer, Dropout, and Activation functions**.**

- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 7+, and others.
- System architecture: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2 / arm64), 64-bit Power8/Power9, s390x (Linux on IBM Z &Linux ONE).
- Minimum 5 GB disk space to download and install.
- Python Libraries - Keras, TensorFlow, NumPy, pandas, and matplotlib.

4.2 ARCHITECTURE / OVERALL DESIGN OF THE PROPOSED SYSTEM



4.1 SYSTEM ARCHITECTURE

CHAPTER 4

DESCRIPTION OF THE PROPOSED SYSTEM

This is a multiclass classification problem which is an example of supervised learning. We have been provided with a fixed number of features for each data point, and we will aim to train a variety of methods on the Convolutional Neural network algorithm, so that, when a new data point arises, our best-performing classifier can be used to categorize the data point as a positive example or negative.

This project focuses on the related works of various authors on dog breed identification such that algorithms were implemented using Google colab which is a machine learning as well as deep learning software written in Python. Various attributes that are essential in the prediction were examined and the dataset of 133 types of dog breeds was also evaluated.

We are going to build Convolutional Neural Networks with Keras and TensorFlow. We are going to examine different models by applying them to CNN architecture. As of now we are using the LeNet-5 CNN architecture model and predict the accuracy and use the model for prediction. Then going to build a flask application by saving this model.

4.1 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLANOF THE PROPOSED MODEL/SYSTEM

- Installing Anaconda Individual edition 64-bit (PY 3.8)
- Use Jupyter notebook in Anaconda Navigator for running project Python notebook.
- Python Notebook also works in Google Colab and Kaggle Notebook editor.