**IMAGE CLASSIFICATION USING CIFAR-10 DATASET**



A mini project-1 report submitted in partial fulfillment of requirements for the award of Degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**V. ANJALI (199X1A05G3)**

**V. BHARATH KUMAR (199X1A05J8)**

**S. RAJ KUMAR (199X1A05J1)**

**Under the Esteemed guidance of**

**Smt. P. SREEDEVI**

**Assistant Professor**

**Department of C.S.E.**

**Department of Computer Science and Engineering**

**G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL**

**(Affiliated to JNTUA, ANANTAPURAMU)**

**2020 - 2021**

**Department of Computer Science and Engineering**

**G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL**

**(Affiliated to JNTUA, ANANTAPURAMU)**



**CERTIFICATE**

This is to certify that the mini project-1 work entitled ‘IMAGE CLASSIFICATION USING CIFAR-10 DATASET’ is a bona fide record of work carried out by

**V. ANJALI (199X1A05G3)**

**V. BHARATH KUMAR (199X1A05J8)**

**S. RAJ KUMAR (199X1A05J1)**

Under my guidance and supervision in fulfillment of the requirements for the award of degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Smt. P. Sreedevi Dr. N. Kasiviswanath,**

Assistant Professor, Professor & Head of the Department,

Department of CSE, Department of CSE.

G. PullaReddy Engineering College, G. PullaReddy Engineering College,

Kurnool. Kurnool.

# DECLARATION

We hereby declare that the project titled “IMAGE CLASSIFICATION USING CIFAR-10 DATASET” is the authentic work carried out by us as students of G. PULLA REDDY ENGINEERING COLLEGE (Autonomous) Kurnool, during June – October 2019 and has not been submitted elsewhere for the award of degree in part or in full to any institute.

**V. ANJALI**

**(199X1A05G3)**

**V. BHARTH KUMAR**

**(199X1A05J8)**

**S. RAJ KUMAR**

**(199X1A05J1)**

**Acknowledge****ment**

We wish to express our deep sense of gratitude to our project guide, **Smt. P. Sreedevi,** Assistant Professor of CSE Department, G. Pulla Reddy Engineering College, for his immaculate guidance, constant encouragement and cooperation which have made possible to bring out this project work.

We are grateful to our project incharge, **Sri. S Vinay Kumar**, Asst Professor of CSE Department, G. Pulla Reddy Engineering College, for helping us and giving us the required information needed for our project work.

We are thankful to our Head of the Department **Dr. N. Kasi Viswanath**, for his whole hearted support and encouragement during the project sessions.

We are grateful to our respected Principal **Dr. B. Sreenivasa Reddy**, for providing requisite facilities and helping us in providing such a good environment.

We wish to convey our acknowledgements to all the staff members of the Computer Science Engineering Department for giving the required information needed for our project work.

Finally we wish to thank all our friends and well wishers who have helped us directly or indirectly during the course of this project work.

**ABSTRACT**

In this project, we work on image classification of the CIFAR-10 dataset using supervised machine learning techniques. The dataset consists of 60,000 32x32RGB images containing one of 10 object classes, with 6000 images per class. We experiment with convolution neural networks (CNN). We use cross-validation by splitting the 50,000training data into 49,000 training samples and 1,000 validation samples to select the optimized hyperparameters for each parametric classifier.

1. INTRODUCTION

1. INTRODUCTION

Image classification refers to the task of extracting information classes from a multibandrasterimage. The resulting raster from image classification can be used to create thematic maps. It is the task of associating one (*single-label classification)* or more (*multi-label classification)* labels to a given image.

* 1. **CIFAR-10 DATASET**

CIFAR is an acronym that stands for the **Canadian Institute For Advanced Research** and the CIFAR-10 dataset was developed along with the CIFAR-100 dataset by researchers at the CIFAR institute.

The dataset is comprised of 60,000 32×32-pixel colour photographs of objects from 10 classes, such as frogs, birds, cats, ships, etc. The class labels and their standard associated integer values are listed below.

* 0: airplane
* 1: automobile
* 2: bird
* 3: cat
* 4: deer
* 5: dog
* 6: frog
* 7: horse
* 8: ship
* 9: truck

These are very small images, much smaller than a typical photograph, and the dataset was intended for computer vision research.

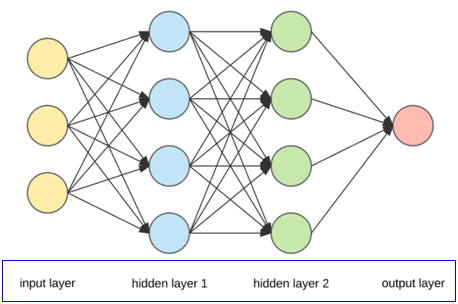
**1.2 NEURAL NETWORKS**

Neural Networks are complex structures made of artificial neurons that can take in multiple inputs to produce a single output.

●This is the primary job of a Neural Network

●Simple terms, to transform input into a meaningful output

●NN consists of an input and output layer with one or more hidden layers.



**Fig 1.2.1 Artificial Neural Networks (ANN)**

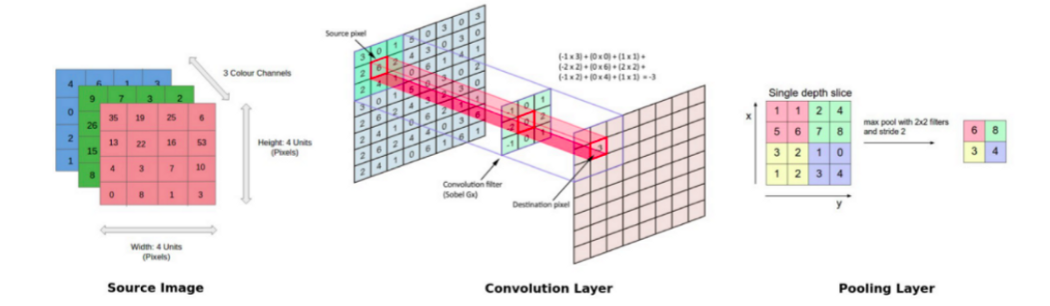
**1.3 CONVOLUTIONAL NEURAL NETWORKS**

Convolutional neural networks, also called as ConvNets, were first introduced in the 1980’s by Yann LeCun, a postdoctoral computer science researcher. Before that we used to use ANN(Artificial Neural Network) to classify images but there was much error rate. So, after CNN was invented, people started to train data based on this technique and minimized the error rate and made our lives so simpler.

Convolutional Neural Network has had ground breaking results over the past decade in a variety of fields related to pattern recognition; from image processing to voice recognition. The most beneficial aspect of CNNs is reducing the number of parameters in ANN. This achievement has prompted both researchers and developers to approach larger models in order to solve complex tasks, which was not possible with classic ANNs. The most important assumption about problems that are solved by CNN should not have features which are spatially dependent. In other words, for example, in a face detection application, we do not need to pay attention to where the faces are located in the images. The only concern is to detect them regardless of their position in the given images. Another important aspect of CNN, is to obtain abstract features when input propagates toward the deeper layers. For example, in image classification, the edge might be detected in the first layers, and then the simpler shapes in the second layers, and then the higher-level features such as faces in the next layers.

CNNs typically consist of multiple convolution and pooling layers which help the deep learning model in automatically extracting relevant features from visual data like images. Due to this multi-layered architecture, CNNs learn a robust hierarchy of features, which are spatial, rotation, and translation invariant.

The key operations in a CNN model are depicted in the figure above. Any image can be represented as a tensor of pixel values. The convolution layers help in extracting features from this image (forms feature maps). Shallower layers (closer to the input data) in the network learn very generic features like edges, corners and so on. Deeper layers in the network (closer to the output layer) learn very specific features pertaining to the input image. The following graphic helps summarize the key aspects of any CNN model.



**Fig 1.3.1 Convolution Neural Networks (CNN)**

**1.4 OBJECTIVE OF THE PROJECT**

Image recognition is a vital component in robotics such as driverless vehicles or domestic robots. Image recognition is also important in image search engines such as Google or Bing image search whereby we use rich image content to query for similar stuff. Like in Google photos where the system uses image recognition to categorize your images into things like cats, dogs, people and so on so that we can quickly search our albums for things like, “give me photos of my cat”, that’s awesome. If we are working on a robotics project where we are required to train our robot so that it can recognize different images. The objective of the project is to build a deep learning model that helps us recognize the object in images and predict the class of the image. (Class ranges from 0 to 9).

**1.5 ORGANISATION OF THE REPORT**

The first chapter deals with the introduction of the image classification, convolutional neural networks, motivating for developing this project, objective of the project.  The second chapter deals with the system specifications.  The third chapter gives us the Design and implementations which includes introduction, source code and description of the languages used for the project, methods of implementation. Finally, the fourth chapter deals with the conclusion.

2. SOFTWARE REQUIREMENTS SPECIFICATION

**2. SOFTWARE REQUIREMENTS SPECIFICATION**

The project requires the following hardware and software capabilities to train a computer how to classify images and to make the computer classify other images. These are the minimal

requirements required for the computer to run the test smoothly.

**2.1 SOFTWARE SPECIFICATIONS:**

* Operating System : Windows/Ubuntu/Linux
* Programming Language : Python
* Development Environment : Anaconda Navigator
* Integrated Development Environment : Jupyter Notebook
* Datasets Required : CIFAR-10

**2.2 HARDWARE SPECIFICATIONS:**

* Processor : Intel core i3/i5
* Hard disk capacity : 40GB minimum
* RAM : 256GB minimum
* Monitor : 15 VGAcolour

3. LITERATURE SURVEY

**3. LITERATURE SURVEY**

**3.1 EXISTING SYSTEM**

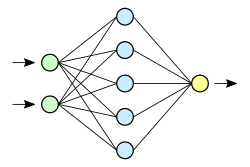
* The existing system is based on classifying images using Artificial Neural Networks(ANN).
* ANNs are computational models inspired by an animal's central nervous system.
* A neural network is an oriented graph. It consists of nodes which in the biological analogy represent neurons, connected by arcs.
* ANN is a set of connected input output networks in which weight is associated with each connection.
* Artificial neural network is an example of supervised learning.
* A neural network contains the following three layers:

**Input Layer:** The activity of input units represents the raw information that can feed into the network.

**Hidden Layer:**

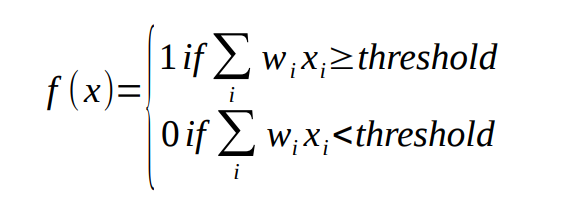
* Hidden Layer is used to determine the activity of each hidden unit.
* The activities of the input units and the weights depend on the connections between the input and the hidden units.
* There may be one or more hidden layers.

**Output Layer:** The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

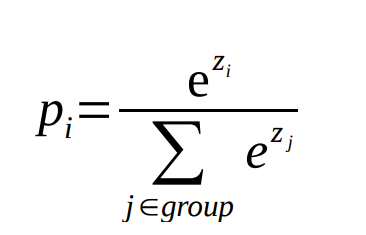


**Fig 3.1.1. Artificial Neural Networks (ANN)**

* The Activation function for training the data set is



* The accuracy formula for calculating the efficiency of the program after training is



**3.2 DISADVANTAGES OF EXISTING SYSTEM**

* Artificial Neural Networks require processors with parallel processing power, by their structure.
* Unexplained functioning of the network.
* There is no specific rule for determining the structure of artificial neural networks.
* Problems have to be translated into numerical values before being introduced to ANN.
* The display mechanism to be determined will directly influence the performance of the network.

**3.3 PROPOSED SYSTEM**

The proposed system uses Convolutional Neural Networks (CNN) which were first introduced by Yann LeCun in the 1980's. Convolution Neural Networks has many hidden layers besides ANN has only one hidden layer. The hidden layers do all the computation between the input and the output. The accuracy of classifying images using CNN is far better than the accuracy of classifying images using CNN. Also, the pressure on the CPU is decreased in CNN compared to ANN. To further improve the accuracy of the model, we use data augmentation which increases the capability of the machine to classify the images correctly.

**3.4 ADVANTAGES OF PROPOSED SYSTEM**

* No need for feature extraction.
* extract informative features from images, eliminating the need of traditional manual image classifying methods.
* CNN compared to its predecessors is that it automatically detects the important features without any human supervision.
* High accuracy than many traditional methods.

**4. ANALYSIS**

**4. ANALYSIS**

**4.1 MODULE ORGANISATION**

This project requires Anaconda Navigator, Jupyter Notebook and CIFAR-10 dataset to proceed for further implementation. We can use Google Colab or Anaconda Navigator to train the dataset. But, to understand the basics of Anaconda Navigator, we are using it rather than Google Colab. Let us take a peek at how to install Anaconda Navigator.

**4.1.1 Anaconda Navigator**

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. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called **Miniconda**, which includes only conda, Python, the packages they depend on, and a small number of other packages.

The following applications are available by default in Navigator:

* JupyterLab
* Jupyter Notebook
* QtConsole
* Spyder
* Glue
* Orange
* RStudio
* Visual Studio Code

The steps involved for creating the environment are

* Go to the official Anaconda Website.
* Install it according to the required specification.
* Open it and go to the Environment tab.
* Create a new environment of your choice.
* Install the required datasets in the created environment and start coding to train the model.

**4.1.2 TensorFlow**

Created by the Google Brain team, TensorFlow is an open-source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

**Advantages of TensorFlow**

* Open-source Platform
* Better Data Visualization
* Keras friendly
* Scalable to develop on any machine
* Compatible with C, C++, Java and Python
* Architectural support with TPU

**Disadvantages of TensorFlow**

* Frequent updates every 2-3 months
* Architectural limitation
* Every code needs to be executed using any platform for its support which increases the dependency for the execution
* Slow speed

TensorFlow is a machine learning platform but it requires an interface to work with known as **KERAS**. Let us take a look into it.

**4.1.3 KERAS**

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.

**Frameworks supported by Keras**

* TensorFlow
* Theano
* PlaidML
* MXNet
* CNTK

**Building a model in Keras**

* Define a network
* Compile the network
* Fit the network
* Evaluate the network
* Make prediction

**4.1.4 PYTHON**

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don’t know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to “big name” languages. Hopefully I can explain it to you.

### Python Concepts

### If you are not interested in the how and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it’s a great one to start programming with.

* Open-source general-purpose language.
* Object Oriented, Procedural, Functional
* Easy to interface with C/ObjC/Java/Fortran
* Easy-ish to interface with C++ (via SWIG)
* Great interactive environment

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

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

### Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data types.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

### Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated. Tuples can be thought of as **read-only** lists.

#### **4.1.5 NumPy**

NumPy’s main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called *axes*. The number of axes is *rank*.

* Offers Matlab-ish capabilities within Python
* Fast array operations
* 2D arrays, multi-D arrays, linear algebra etc.

**4.1.6 Matplot**

Plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK**.**

**4.2 FEASIBILITY STUDY**

In this project, we have only two choices. The system has to detect the class to which it belongs. It may be right or wrong. It totally depends on how and which method the system is totally trained.

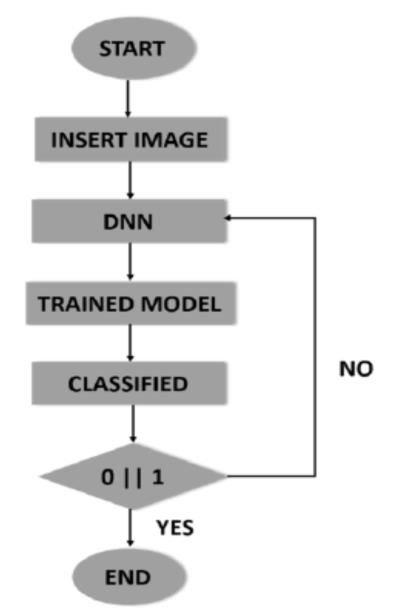
**CNN**

* Has high accuracy in image classification problems.
* Automatically detects the important features without any human supervision.
* Is of weight sharing type.

Due to these factors, the project is more feasible to make it work with CNN than with ANN.

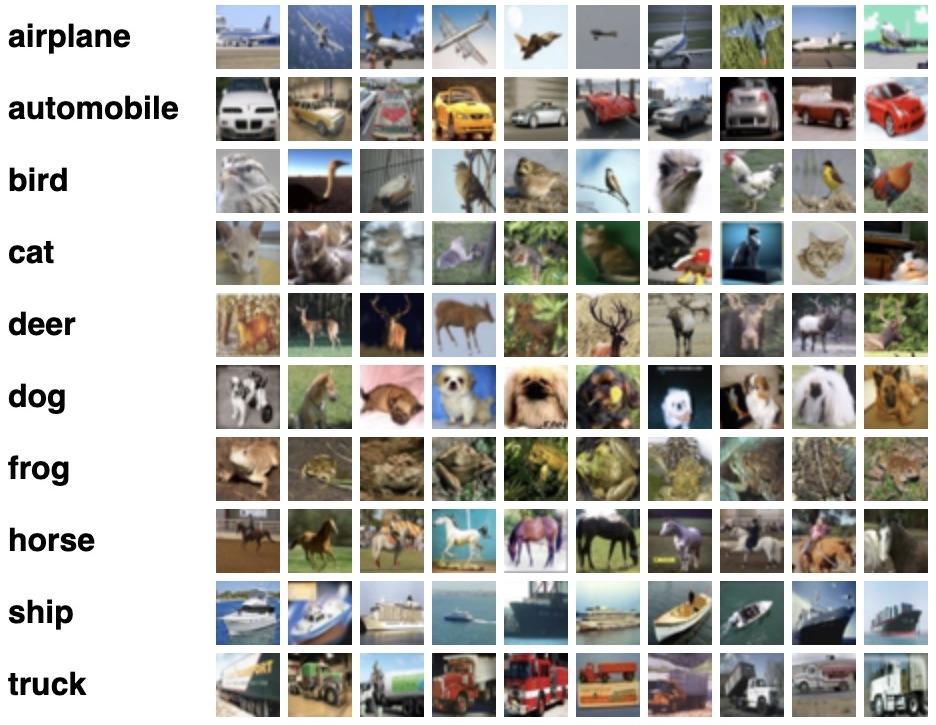
**5. DESIGN**

**5.1 FLOW CHART FOR IMAGE CLASSIFICATION**

****

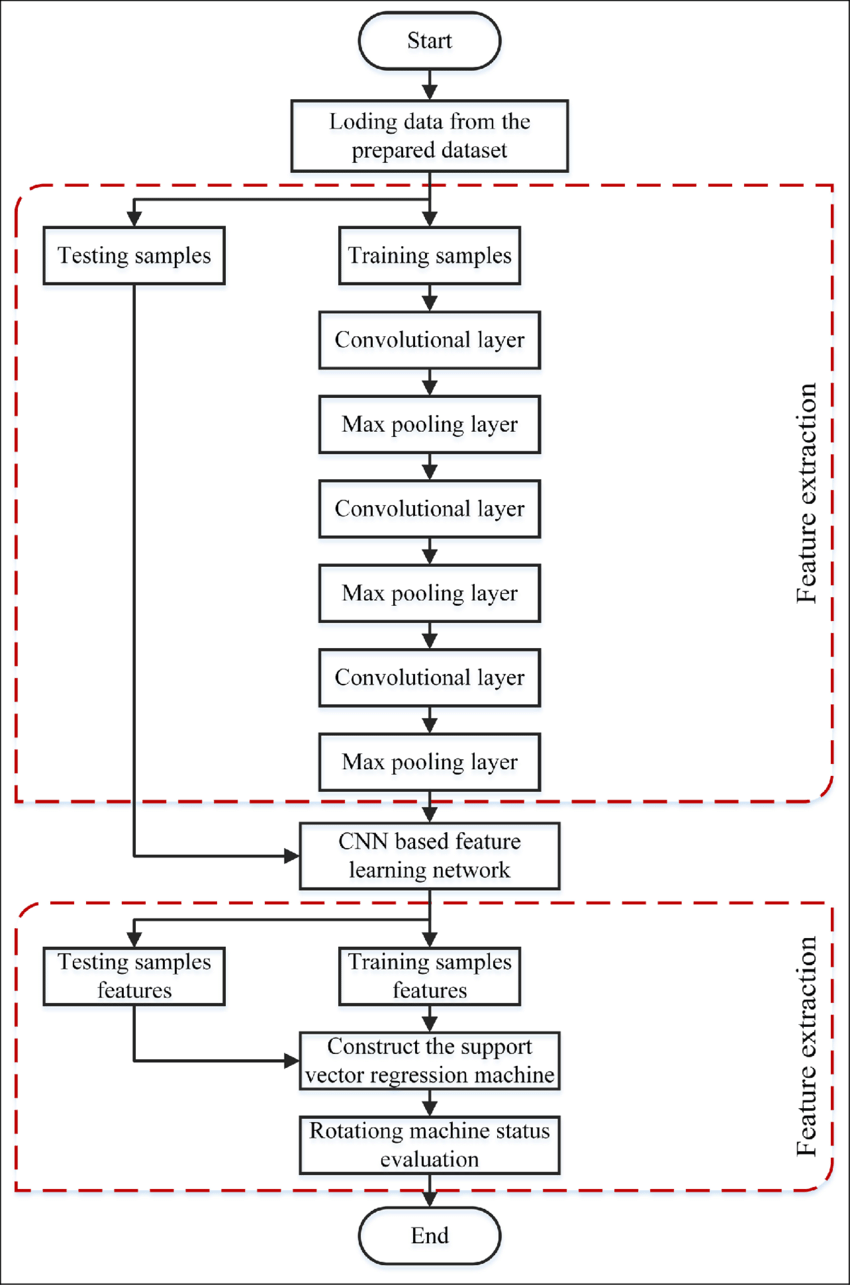
**Fig 5.1.1 Flow chart for Image Classification**

**5.2 CIFAR-10 DATASET STRUCTURE**

****

**Fig 5.2.1 CIFAR-10 Dataset representing its classes**

**5.3 FLOWCHART FOR IMAGE CLASSIFICATION USING CNN**

****

**Fig 5.3.1 Flow chart representing Image Classification using CNN**

**6. IMPLEMENTATION AND**

**RESULT ANALYSIS**

**6. IMPLEMENTATION AND RESULT ANALYSIS**

**6.1 ANACONDA NAVIGATOR**

* Open the Anaconda Navigator.
* Go to the Environments tab.
* Create a new environment of your choice.
* In that environment, go to the downloads section.
* Install TensorFlow, Keras Library, Matplot Library, Numpy from the not installed section.
* Go home and install the Jupyter Notebook.
* Complete the installation setup and open it.

**6.2 JUPYTER NOTEBOOK**

* Create as many cells as you want for your project source code.
* Each cell can be made to run individually.
* The below source code is copied in the Jupyter Notebook to train the model.

**6.3 SOURCE CODE**

import tensorflow as tf

print(tf.\_\_version\_\_) #Verifies that tensorflow is installed

import numpy as np

print(np.\_\_version\_\_) #Verifies that numpy is installed

import matplotlib.pyplot as plt

from tensorflow.keras import datasets, layers, models #Imports all the required things from Keras

(x\_train, y\_train), (x\_test, y\_test) = datasets.cifar10.load\_data() #Loading the CIFAR- 10 dataset

x\_train.shape #size of x\_train dataset

y\_train.shape #size of y\_train dataset

x\_test.shape #size of x\_test dataset

y\_test.shape #size of y\_test dataset

y\_train[:5]

y\_train = y\_train.reshape(-1,)

y\_train[:5]

y\_test = y\_test.reshape(-1,)

classes = ["airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"]

def plot\_sample(x, y, index):

plt.figure(figsize = (15,2))

plt.imshow(x[index])

plt.xlabel(classes[y[index]])

plot\_sample(x\_train, y\_train, 0)

plot\_sample(x\_train, y\_train, 1)

x\_train = x\_train / 255.0

x\_test = x\_test / 255.0

ann = models.Sequential([ #Training with ann model

layers.Flatten(input\_shape=(32,32,3)),

layers.Dense(3000, activation='relu'),

layers.Dense(1000, activation='relu'),

layers.Dense(10, activation='softmax')

])

ann.compile(optimizer='SGD',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

ann.fit(x\_train, y\_train, epochs=5)

import numpy as np

from sklearn.metrics import confusion\_matrix , classification\_report

y\_pred = ann.predict(x\_test)

y\_pred\_classes = [np.argmax(element) for element in y\_pred]

print("Classification Report: \n", classification\_report(y\_test, y\_pred\_classes))

#Generating Classification report for ANN model

cnn = models.Sequential([ #Training with CNN model

layers.Conv2D(filters=32, kernel\_size=(3, 3), activation='relu', input\_shape=(32, 32, 3)),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(filters=64, kernel\_size=(3, 3), activation='relu'),

l

layers.MaxPooling2D((2, 2)),

layers.Flatten(),

layers.Dense(64, activation='relu'),

layers.Dense(10, activation='softmax')

])

cnn.compile(optimizer='adam',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

cnn.fit(x\_train, y\_train, epochs=10)

cnn.evaluate(x\_test,y\_test)

y\_pred = cnn.predict(x\_test)

y\_pred[:5]

y\_classes = [np.argmax(element) for element in y\_pred]

y\_classes[:5]

y\_test[:5]

plot\_sample(x\_test, y\_test,5)

classes[y\_classes[5]]

plot\_sample(x\_test, y\_test,25)

classes[y\_classes[25]]

plot\_sample(x\_test, y\_test,100)

classes[y\_classes[100]]

plot\_sample(x\_test, y\_test,1729)

classes[y\_classes[1729]]

**6.4 OUTPUT SCREENS**

****

**Fig 6.4.1 Output screen for right prediction (1)**

****

**Fig 6.4.2 Output screen for right prediction (2)**

****

**Fig 6.4.3 Output screen for right prediction (3)**

** Fig 6.4.4 Output screen for wrong prediction**

**7. CONCLUSION AND FUTURE ENHANCEMENTS**

**CONCLUSION AND FUTURE ENCHANCEMENTS**

**7.1 CONCLUSION**

We have walked a long way from using Simple Neural Networks to understanding complex Convolution Neural Networks and make the computer classify them without human interruption. Although, the training dataset made an accuracy of more than 73%, it is so far less to apply them in real life. It should at least match 90% accuracy to be used in normal life to prevent mistakes which costs a significant loss if used. The CIFAR-10 dataset we have used has some images in which the objects are so blurred, which may be interpreted that it is so usual the training data set is making a mistake because it is so hard for even a human to predict them correctly.

**7.2 FUTURE ENCHANCEMENTS**

The accuracy of the model can be increased by using data augmentation techniques like for each input image, we generate a “duplicate” image that is shifted, zoomed in/out, rotated, flipped, distorted, or shaded with a hue. Both image and duplicate are fed into the neural net. For a dataset of size N, we generate a dataset of 2N size. By using these methods, we can increase the overall accuracy of the model but the shift of accuracy is not much skyrocketing. If testing time is the constraint, then we can use fuzzy classifiers. If the classes are much distinct, then we can use decision trees, which produces a high rate of accuracy.

REFERENCES

1. <https://www.kaggle.com/c/cifar-10/leaderboard>
2. <http://www.cs.toronto.edu/~kriz/cifar.html>
3. M. Manoj Krishna, M. Neelima, M. Harshali, M. Venu Gopala Rao, “Image Classification using Deep Learning”, International Journal of Engineering & Technology, vol 7, 2018.
4. Rajani Bala, Dr.Dharmendar Kumar, “Classification Using ANN”, International Journal of Computational Intelligence Research, ISSN 0973-1873 Volume 13, Number 7 (2017).
5. Mohd Azlan Abu, Nurul Hazirah Indra, Abdul Halim Abd Rahman , Nor Amalia Sapiee and Izanoordina Ahmad, “A study on Image Classification based on Deep Learning and Tensorflow”, International Journal of Engineering Research and Technology, Volume 12, Number 4 (2019).
6. Dr. Leena Patil, Miss. Anagha M. Pawar, “A Literature Survey on Classification of Images using State of the Art Machine Learning Techniques”, IJRECE Volume 7, Issue 1, 2019.
7. <https://www.tensorflow.org/install>
8. <https://numpy.org/doc/stable/user/whatisnumpy.html>
9. <https://en.wikipedia.org/wiki/Keras#:~:text=Keras%20is%20an%20open%2Dsource,interface%20for%20the%20TensorFlow%20library.&text=Designed%20to%20enable%20fast%20experimentation,friendly%2C%20modular%2C%20and%20extensible>.
10. <https://www.techopedia.com/definition/5967/artificial-neural-network-ann>
11. M. Seetha, I.V. Muralikrishna, B.L. Deekshatulu, B.L. Malleswari, Nagaratna P. Hegde, “Artificial Neural Networks and Other methods of Image Classification”, Journal of Theoretical and Applied Information Technology.