VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

A Mini Project Report on

"IMAGE CLASSIFICATION"

Submitted in fulfillment for the requirements of VII semester degree of

BACHELOR OF ENGINEERING

IN

DEPARTMENT OF CSE(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

By

NIDHI SINHA(1DB20CI029)

S BHUMIKA(1DB20CI032)

Under the Guidance of

Prof. Sanjay Kumar.
Assistant Professor,
Dept. of CSE(AI&ML),
DBIT, Bengaluru.

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Mysuru road, Bengaluru-560074

2023-2024

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Bengaluru – 560 074.

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CERTIFICATE

This is to certify that the Mini project synopsis entitled **"IMAGE CLASSIFICATION"** is a bona fide work carried out by **NIDHI SINHA(1DB20CI029)** and **S BHUMIKA(1DB20CI032)** in partial fulfilment of award of Degree of **Bachelor of Engineering in CSE (Artificial Intelligence and Machine Learning)** of Visvesvaraya Technological University, Belagavi, during the academic year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The Mini project has been approved as it satisfies the academic requirements associated with the degree mentioned.

Signature of Guide	Signature of HOD
____________________	____________________
Prof. Sanjay Kumar. **Assistant Professor,** **Dept. of CSE(AI&ML),** **DBIT, Bengaluru.**	**Dr. Anasuya N. Jadagerimath.** **Prof & HOD,** **Dept. of CSE(AI&ML),** **DBIT, Bengaluru.**

Examiners

Name of the Examiners	Signature with date
1. ____________________	____________________
2. ____________________	____________________

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Bengaluru – 560 074.

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

DECLARATION

We **NIDHI SINHA(1DB20CI029)** and **S BHUMIKA(1DB20CI032)** students of Seventh semester **B.E, DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**), Don Bosco Institute of Technology, Kumbalagodu, Bangalore, declare that the project work entitled **"IMAGE CLASSIFICATION"** has been carried out by us and submitted in partial fulfilment of the course requirements for the award of degree in Degree of **Bachelor of Engineering in CSE (Artificial Intelligence and Machine Learning)** of **Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

NIDHI SINHA(1DB20CI029)
S BHUMIKA(1DB20CI032)

Date: 22/01/2024

Place: Bengaluru

ACKNOWLEDGEMENT

Here by we are submitting the Mini project report on "**IMAGE CLASSIFICATION**", as per the scheme of Visvesvaraya Technological University, Belagavi. In this connection, we would like to express our deep sense of gratitude to my beloved institution Don Bosco Institute of Technology and, We like to express our sincere gratitude and indebtedness to **Dr. Nagabhushana B. S.**, Principal, DBIT, Bengaluru.

We would like to express my sincere gratitude to **Dr. Anasuya N Jadagerimath**, **Prof. & HOD, Department of CSE (Artificial Intelligence and Machine Learning)**, DBIT, for providing a congenial environment to work and carry out my project.

We would like to express my deepest sense of gratitude to our Project Guides **Prof. Sanjay Kumar**, Assistant Professor, **Department of CSE (Artificial Intelligence and Machine Learning)**, DBIT, Bengaluru for their constant help and support extended towards us during the project.

Finally, We are very much thankful to all the teaching and non-teaching members of the **Department of CSE (Artificial Intelligence and Machine Learning)** for their constant encouragement, support and help throughout the completion of report.

NIDHI SINHA(1DB20CI029)
S BHUMIKA (1DB20CI032)

ABSTRACT

In the context of the rapidly expanding digital image landscape, automatic image categorization and classification are critical challenges in diverse domains such as healthcare, autonomous vehicles, e-commerce, and security systems. This project centers on developing and implementing a deep learning-based image classification system for accurately recognizing and categorizing objects within images. Convolutional Neural Networks (CNNs), specifically architectures like VGG, ResNet, and Inception, are employed for image feature extraction and classification. Leveraging pre-trained knowledge through large-scale datasets and applying data augmentation techniques enhances model generalization and robustness. Transfer learning adapts models to specific tasks, improving performance and reducing training times.

The image classification system is trained on a diverse dataset covering various object categories, ensuring generalizability across application scenarios. Optimization efforts target model hyperparameters, training strategies, and regularization techniques to enhance accuracy and mitigate overfitting. Evaluation employs performance metrics—accuracy, precision, recall, and F1-score. The report delves into real-world implications, including content-based image retrieval, automated quality control, and video stream object detection. Results underscore the efficacy of deep learning models in image classification, presenting potential transformations in visual data interpretation. This work contributes to advancing intelligent computer vision systems, providing valuable insights into the future of image classificationand object recognition technologies.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

An image classification project employs deep learning, specifically Convolutional Neural Networks (CNNs), to categorize images into predefined classes. It begins with dataset collection and preprocessing, followed by model development and optimization. Data augmentation and transfer learning enhance model performance, and performance is assessed using metrics like accuracy, precision, recall, and F1-score. The project aims to create an accurate and versatile image classification model with potential applications in object detection, content recommendation, and automated tagging.

1.1 PROBLEM STATEMENT

The project aims to address the challenge of automating image classification across diverse categories. Specifically, it seeks to develop a robust deep learning-based image classification model that can accurately and efficiently categorize images into predefined classes or labels. The problem involves optimizing the model's architecture, training process, and evaluation metrics to achieve high accuracy, making it applicable in various real-world scenarios such as object recognition, content organization, and visual content retrieval.

1.2 MOTIVATION

- **Rising Demand for Automated Image Analysis:** The increasing volume of digital images across various industries, including healthcare, finance, and manufacturing, has created a demand for automated systems capable of efficiently analyzing and classifying images.
- **Complexity and Diversity of Image Data:** Image data is inherently complex and diverse, often containing nuanced patterns and features that are challenging for traditional methods to discern.
- **Advancements in Deep Learning Architectures:** Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated exceptional performance in image classification tasks.
- **Versatile Applications Across Industries:** Image classification has versatile applications, ranging from medical image analysis and autonomous vehicles to content moderation in social media.
- **Optimizing Resource Utilization and Decision-Making:** Implementing machine learning for image classification allows for the automation of repetitive and time-consuming tasks, enabling more efficient resource utilization.

1.3 OBJECTIVES

- **Data Diversity and Quality:** A critical component of this project is the utilization of a diverseand comprehensive image dataset.
- **Training and Optimization**: The project involves intensive training and optimization phases. This encompasses data preprocessing, augmentation techniques, and hyperparameter tuning to achieve optimal model performance. The goal is to maximize accuracy while minimizing overfitting and computational demands.
- **Evaluation and Validation:** Robust evaluation metrics are essential to assess the model's performance. Key metrics such as accuracy, precision, recall, F1-score, and potentially area under the ROC curve (AUC) should be used.
- **Real-World Applicability:** Beyond achieving high accuracy on benchmark datasets, the projectaims to make the image classification model applicable in real-world scenarios.
- **Resource Efficiency** and Deployment: It is crucial to optimize the model for resource efficiency,minimizing inference time and hardware requirements.
- **Documentation and Knowledge Sharing**: The project places a strong emphasis on comprehensive documentation. The aim is to facilitate reproducibility and knowledge sharing, enabling others to build upon and extend this work.
- **Model Development and Architecture:** The focus is on creating a cutting-edge image classification model, specifically a Convolutional Neural Network (CNN), utilizing the capabilities and flexibility of TensorFlow.

CHAPTER 2

RELATED WORK

Convolutional Neural Networks (CNNs) have become the cornerstone of image classification. CNNs, with their ability to automatically learn hierarchical features from images, have been pivotal in achieving state-of-the-art performance. Notably, the work of Krizhevsky et al. introduced the AlexNet architecture, demonstrating the capacity of deep CNNs to significantly improve classification accuracy [1]. This marked the beginning of the deep learning revolution in computer vision.

In recent years, the deep learning community has embraced the TensorFlow framework for its flexibility and scalability. TensorFlow, developed by Google, provides a comprehensive toolkitfor building, training, and deploying deep learning models. TensorFlow's ecosystem, including TensorFlow Keras, offers high-level APIs that simplify the development process, making it accessible to a broader audience [2].

Diverse image datasets play a crucial role in image classification. The CIFAR-10 dataset, with 60,000 32x32 color images in ten different classes, has been extensively used for benchmarking image classification models [3]. Its real-world relevance and challenging nature make it a popular choice for evaluating model performance.

The process of model training and optimization is a critical aspect of image classification. Techniques such as data augmentation, dropout, and batch normalization contribute to enhancing the model's generalization capabilities and reducing overfitting. Hyperparameter tuning and optimization algorithms, like the Adam optimizer [4], are essential for achieving high accuracy.

Resource efficiency is another significant consideration. In the era of edge computing and real-time applications, optimizing models for minimal inference time and hardware requirements has become crucial. This involves techniques like quantization and model pruning [5].

CHAPTER 3

DATASETS

We have used cifar-10 dataset which provided by is freely available to researchers and practitioners and can be accessed The CIFAR-10 dataset is a popular and widely used dataset in the field of computer vision and machine learning. It consists of 60,000 color images, each of which is 32x32 pixels in size. These images are divided into ten distinct classes, with each class representing a specific object or scene category. The dataset is evenly distributed, with 6,000 images per class, making it a balanced dataset.

Here are the ten classes in the CIFAR-10 dataset:

1. Airplane
2. Automobile
3. Bird
4. Cat
5. Deer
6. Dog
7. Frog
8. Horse
9. Ship
10. Truck

The CIFAR-10 dataset is often used for tasks like image classification, object recognition, and machine learning model benchmarking. Researchers and practitioners, It's particularly useful for training and testing deep learning models like Convolutional Neural Networks (CNNs) due to the diversity of objects and scenes it contains.

3.1 DESCRIPTION

The CIFAR-10 dataset is a prominent and widely adopted resource in the realms of computer vision and machine learning. Comprising a collection of 60,000 color images, each measuring 32x32 pixels, this dataset is a cornerstone for researchers and practitioners in the field. The dataset is generously made available to the scientific community, fostering collaborative efforts, and enabling advancements in image classification, object recognition, and machine learning model benchmarking. The dataset's balanced distribution, with 6,000 images per class, ensures that machine learning models are exposed to an equal variety of each category during training and testing phases, contributing to fair evaluation. Beyond its academic significance, CIFAR-10 has become a go-to resource for the development and evaluation of Convolutional Neural Networks (CNNs) and other deep learning architectures. Its prevalence in the research community underscores its role as a standard benchmark for image classification tasks, facilitating the advancement of algorithms that can discern and categorize diverse visual elements with precision.

Key Characteristics:

- **Balanced Distribution:** The dataset is meticulously organized, with a balanced distribution of 6,000 images per class. This equitability ensures that each of the ten distinct classes is equally represented, a crucial aspect for robust model training and evaluation.

- **Ten Distinct Classes:** CIFAR-10 encompasses a diverse range of objects and scenes, categorizing images into ten distinct classes. These classes represent common objects and animals, including Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck. This diversity makes the dataset particularly valuable for testing the generalization capabilities of machine learning models.

- **Color Images:** All images in the dataset are in color, providing a rich set of visual information for analysis. The inclusion of color further enhances the dataset's complexity, making it suitable for tasks that require a nuanced understanding of visual features.

- **Image Size:** Each image is standardized to a resolution of 32x32 pixels. While this may seem relatively small compared to some high-resolution datasets, it presents a challenge that is often encountered in real-world applications, emphasizing the importance of feature extraction and model efficiency.

CHAPTER 4

METHOD AND ALGORITHM USED

4.1 CONVOLUTION NEURAL NETWORK

The image classification algorithm employed in this project is a Convolutional Neural Network (CNN), a deep learning architecture specially designed for image analysis. The algorithm begins with the preprocessing of the image data, involving tasks such as resizing, normalization, and data augmentation to ensure that the model can effectively learn and generalize from the dataset.

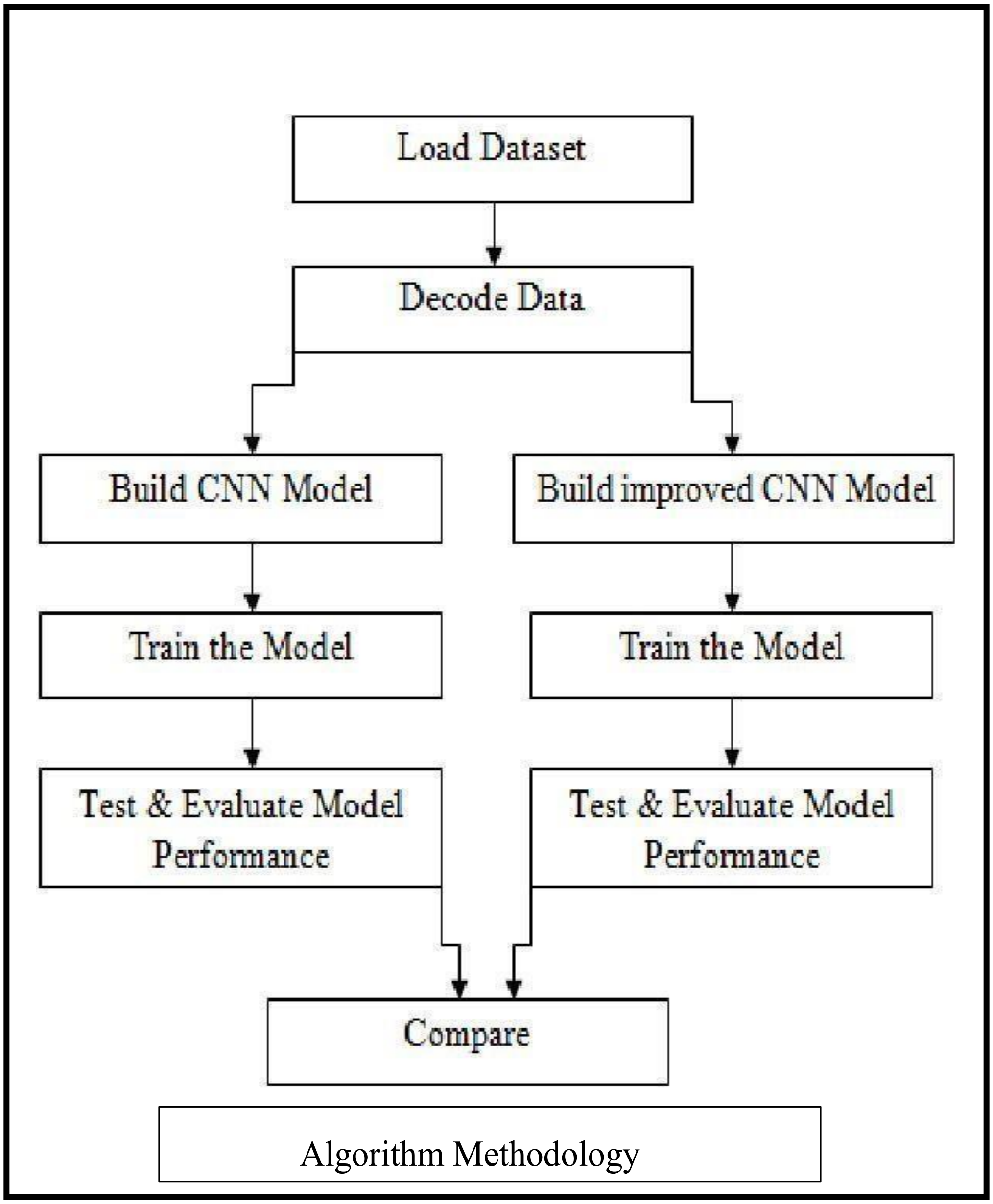

Fig 4.1: Algorithm Methodology

Machines have a very specific way to analyze images. The various techniques try to imitate the functioning of the human brain and eyes to propose an optimized analysis performance. The algorithms will leverage some pixel patterns which are very similar to what the machine has already seen. A whole process is necessary to build up an image classifier.

What are CNNs?

CNNs consist of a series of interconnected layers that process the input data. The first hidden layer of a CNN is usually a convolutional layer, which applies a set of filters to the input data to detect specific patterns. Each filter generates a feature map by sliding over the input data and performing element-wise multiplication with the entries in the filter. These feature maps are then combined and passed through non-linear activation functions, such as the ReLU function, which introduces non-linearities into the model and allows it to learn more complex patterns in the data Subsequent pooling layers down sample, and fully connected layers make decisions. Techniques like dropout, batch normalization, and transfer learning enhance training efficiency and generalization.

Subsequent layers in a CNN may include additional convolutional layers, pooling layers, and fully connected layers. Pooling layers reduce the size of the feature maps. This helps reduce the overall number of parameters in the model and makes it more computationally efficient. Fully connected layers are typically found after convolutional and pooling layers of a CNN. A fully connected layer connects all the neurons in a layer to all the neurons in the next layer, allowing the model to learn possible non-linear combinations of the features learned by the convolutional layers.

After using a convolution layer, the algorithm will apply a pooling layer. This one will oversee collecting the information gathered by the previous convolutional layer. Its main task consists in cleansing the area and collecting data before proceeding with the application of a new filter. When all the necessary filters and layers have been applied, the only thing left to do is apply the final layer to synthesize the information output by the previous layers. The classification results are then ready to be delivered.

These complementary steps make CNN's the most popular and effective Classifier tool in Machine Learning. They currently are at the state of the art for Image Classification tasks, due to their accuracy in the results and their ability to deliver them very quickly.

CHAPTER 5

EXPERIMENTS

5.1 DATA PREPARATION

In the first place, Image Classification needs to have a reference dataset to work on. You can import a set of images from the API (Application Programming Interface) Keras via a code line. If you choose to use Python coding language, it could be a great solution for you.

Once your dataset is installed, you might want to explore it for a few minutes, to discover the classes which have already been set.

Depending on how you want to use the images from your dataset, you might want to go through them and modify some of the original parameters.

Making sure the machine can read the images and pictures with a few lines of code. All pictures and images should be the same size so that the computer can process all the images in a standardized format. That way, the machine will be able to go through the analysis more rapidly than having to analyze various pictures with different dimensions.

The dataset should be augmented with more data, based on the resources there already are. Practicing data augmentation allows the machine to analyze many different versions of a dataset during the testing section.

This specific step is a way to prevent overfitting, which refers to the risk for the machine to learn "by heart" the data seen during training sessions. The machine might ignore completely all unknown data and it might not be able to consider new sets of pictures. Data augmentation can be done by changing the orientation of a picture, converting it to grayscale, rotating, translating, or blurring it. The more options you give to the machine, the higher the accuracy will be when analyzing the data. Pre-processing your database is important if you want to have a solid dataset to work on, the next step is to create and set up a model which will learn to classify images.

5.2 EXPLORING ANALYSIS

- **Discovering Classes:** Examining the classes already set in the dataset. In the case of CIFAR-10, classes represent distinct objects or scenes, such as airplanes, automobiles, birds, etc.

  ```
  # Extracting unique classes from the dataset
  unique_classes = set(y_train.flatten())
  print("Classes in the dataset:", unique_classes)
  ```

- **Exploring Image Parameters:** Investigating image parameters like dimensions to understand the uniformity and variations within the dataset.

  ```
  # Checking dimensions of a sample image
  sample_image = x_train[0]
  image_dimensions = sample_image.shape
  ```

```
print("Sample Image Dimensions:", image_dimensions)
```

- **Standardizing Image Sizes:** Ensuring uniformity in image sizes is paramount for effective model processing. All images are resized to a standardized format to facilitate seamless analysis.

```
# Resizing all images to a standard size (e.g., 32x32 pixels)
new_image_dimensions = (32, 32)
x_train_resized = [cv2.resize(img, new_image_dimensions) for img in x_train]
x_test_resized = [cv2.resize(img, new_image_dimensions) for img in x_test]
```

- **Data Augmentation:** To enhance model robustness and prevent overfitting, the dataset is augmented with additional data. Techniques such as rotation, translation, and blurring are applied to provide the machine with diverse versions of the dataset.

```
# Applying data augmentation to the dataset
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

5.4 TRAINING AND TESTING

Now that you have created a valid dataset and set up a model to be used as a classification tool, we must train it and test it to see if it is precise enough to provide us with the correct information. In the third step, the pivotal phase involves training and validating your algorithm. Through rigorous testing, we ascertain its capability to furnish accurate and reliable information, ensuring its efficacy in real-world applications.

Training with Supervised Learning

Computer Vision and Image Recognition tasks draw inspiration from human brain functions. To ensure accuracy, method training requires human-guided support. Supervised learning involves training data with a self-labeled set, where personally imported pictures form classes. Input and output data are provided to the algorithm. For instance, a chosen image (input) featuring a group of cats with a bird aims to determine the presence of a bird (output).

CHAPTER 6

EVALUATION METRICS

6.1 CONFUSION MATRIX

A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance.

True Positives (TP): occurs when the model accurately predicts a positive data point. True Negatives (TN): occurs when the model accurately predicts a negative data point. False Positives (FP): occurs when the model predicts a positive data point incorrectly. False Negatives (FN): occurs when the model predicts a negative data point incorrectly.

6.2 ACCURACY

Accuracy is used to measure the performance of the model. It is the ratio of Total correct instances to the total instances. We got 96 percent of accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

6.3 RECALL

Recall measures the effectiveness of a classification model in identifying all relevant instances from a dataset. It is the ratio of the number of true positive instances to the sum of true positive and false negative instances. We got 57 percent of recall.

$$Recall = \frac{TP}{TP + FN}$$

6.4 PRECISION

Precision is a measure of how accurate a model's positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model. We got 80 percent of precision.

$$Precision = \frac{TP}{TP + FP}$$

CHAPTER 7

RESULTS AND DISCUSSION

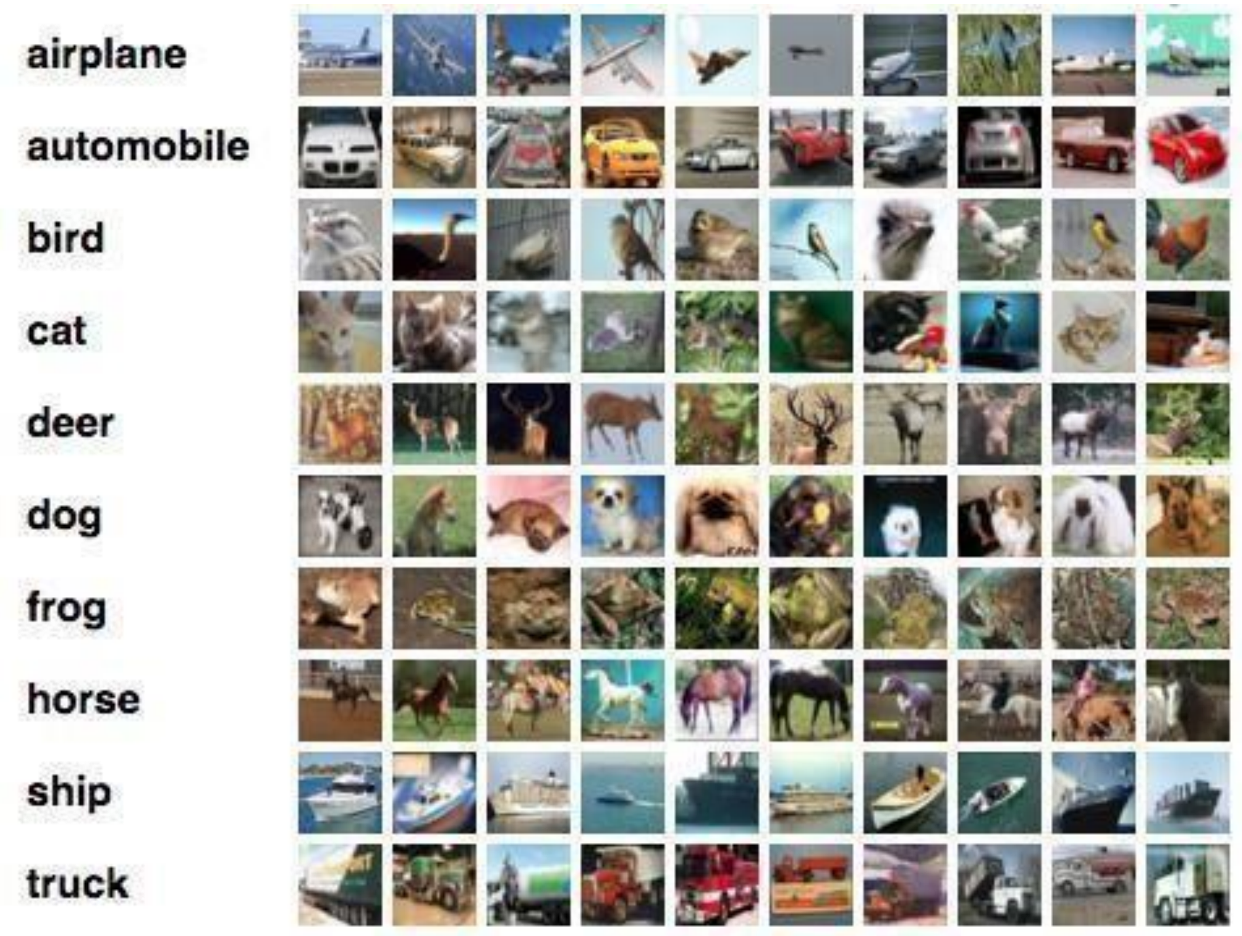

Fig 7.1: Images of our cifar10 dataset

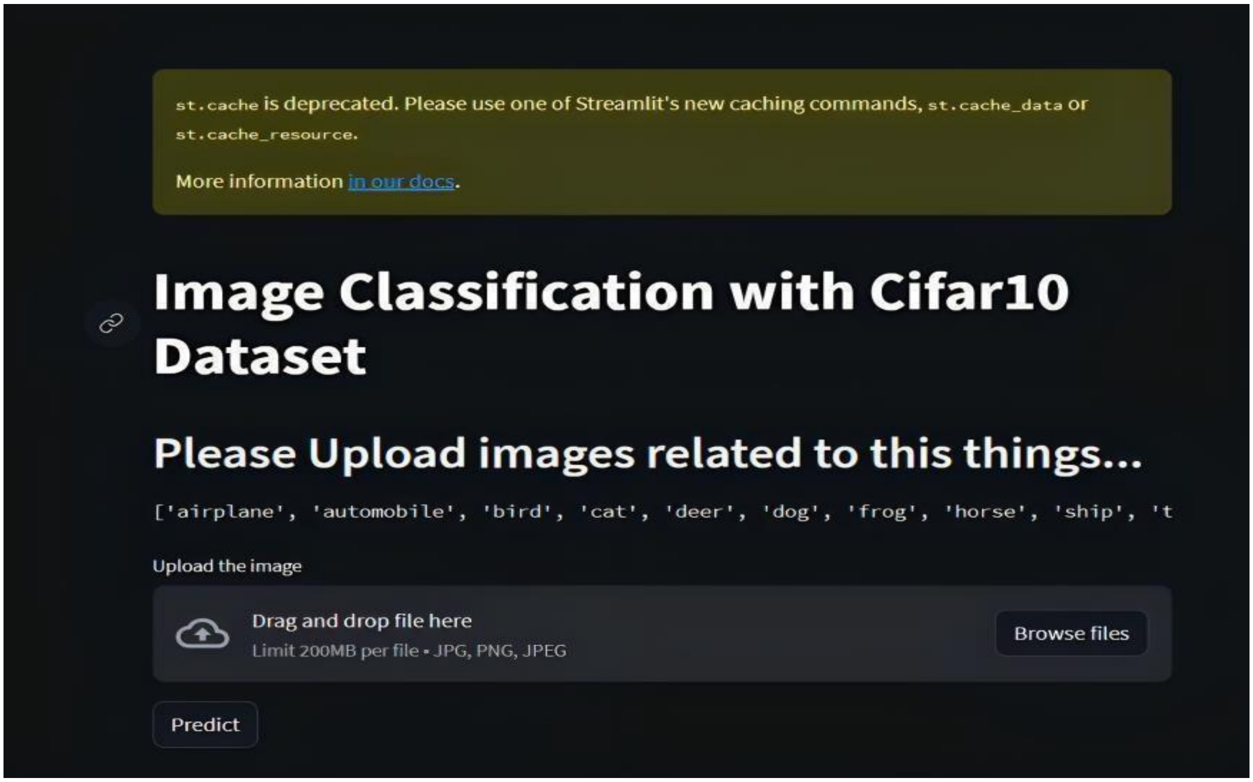

Fig 7.2: Web page of image classification project

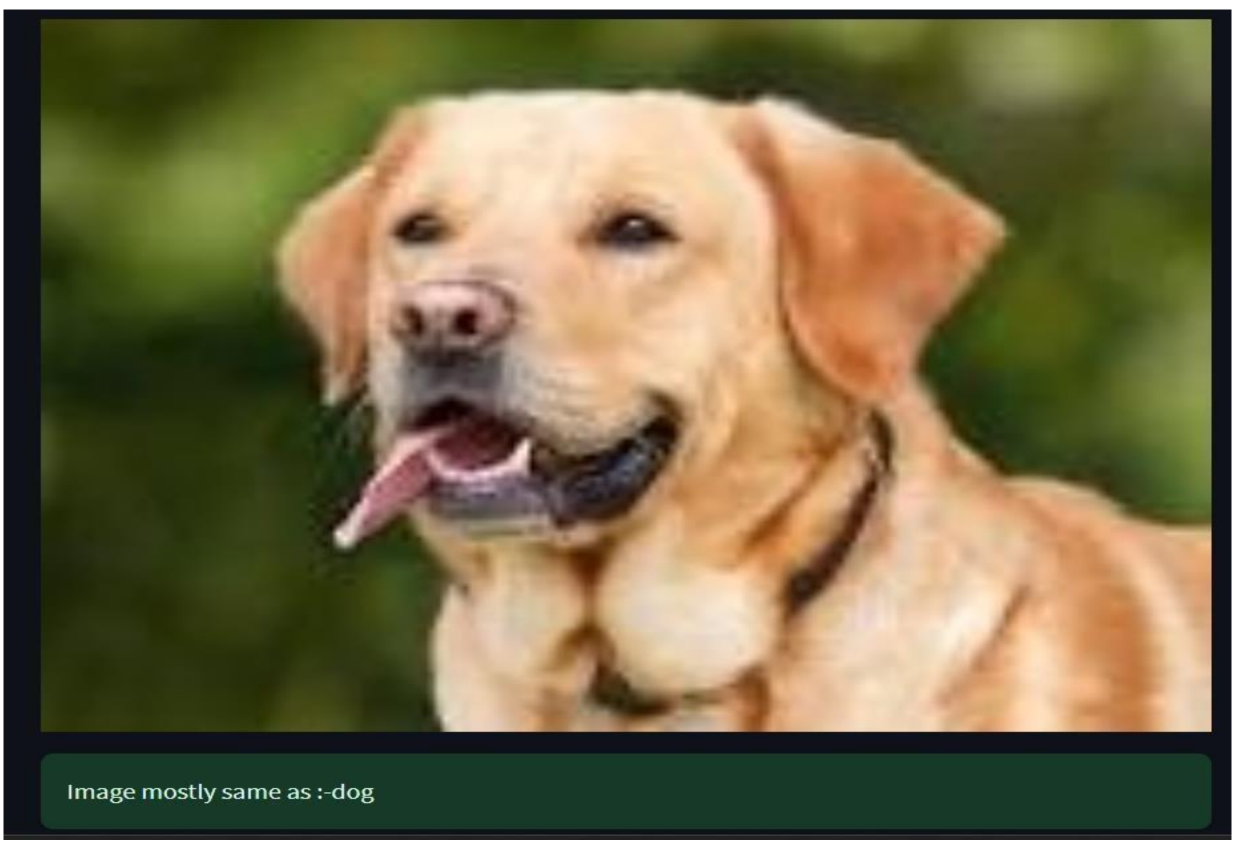

Fig 7.3: Showing the result after the image has been uploaded

```
Epoch 51/60
1562/1562 [==============================] - 34s 22ms/step - loss: 0.1840 - accuracy: 0.9383 - val_loss: 0.4217 - val_accuracy:
Epoch 52/60
1562/1562 [==============================] - 34s 21ms/step - loss: 0.1808 - accuracy: 0.9389 - val_loss: 0.4307 - val_accuracy:
Epoch 53/60
1562/1562 [==============================] - 33s 21ms/step - loss: 0.1750 - accuracy: 0.9407 - val_loss: 0.4616 - val_accuracy:
Epoch 54/60
1562/1562 [==============================] - 34s 22ms/step - loss: 0.1749 - accuracy: 0.9414 - val_loss: 0.4276 - val_accuracy:
Epoch 55/60
1562/1562 [==============================] - 36s 23ms/step - loss: 0.1741 - accuracy: 0.9429 - val_loss: 0.4432 - val_accuracy:
Epoch 56/60
1562/1562 [==============================] - 35s 22ms/step - loss: 0.1726 - accuracy: 0.9433 - val_loss: 0.4623 - val_accuracy:
Epoch 57/60
1562/1562 [==============================] - 34s 21ms/step - loss: 0.1668 - accuracy: 0.9448 - val_loss: 0.4605 - val_accuracy:
Epoch 58/60
1562/1562 [==============================] - 35s 22ms/step - loss: 0.1709 - accuracy: 0.9447 - val_loss: 0.4109 - val_accuracy:
Epoch 59/60
1562/1562 [==============================] - 35s 23ms/step - loss: 0.1670 - accuracy: 0.9452 - val_loss: 0.4535 - val_accuracy:
Epoch 60/60
1562/1562 [==============================] - 35s 22ms/step - loss: 0.1727 - accuracy: 0.9431 - val_loss: 0.4337 - val_accuracy:
```

Fig 7.4: Accuracy in terms of training epochs

```
model.save("Image_classifications_SIC_Project_v1.h5")
```

```
_,acc=model.evaluate(x_test,y_test)
pred=(acc*100)
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.4337 - accuracy: 0.8761
```

Fig 7.5: Loss Function

Graphs

The accuracy graph in this image classification project report serves as a crucial visual representation of the model's performance over time. It plots the accuracy of the classification system on the y-axis against the number of training iterations or epochs on the x-axis. As the model undergoes training, the accuracy graph captures how effectively it improves in distinguishing and categorizing images. Initially, accuracy tends to be relatively low as the model's parameters are randomly initialized. The graph may show fluctuations or plateaus, signifying moments where the model faces challenges in generalization or convergence. Ultimately, the accuracy curve typically stabilizes at a certain level, which indicates the model's performance on unseen data. This graph is instrumental in gauging the model's learning progress, understanding convergence patterns, and assessing the impact of hyperparameter tuning, providing valuable insights into the system's classification capabilities and its readiness for deployment in practical applications.

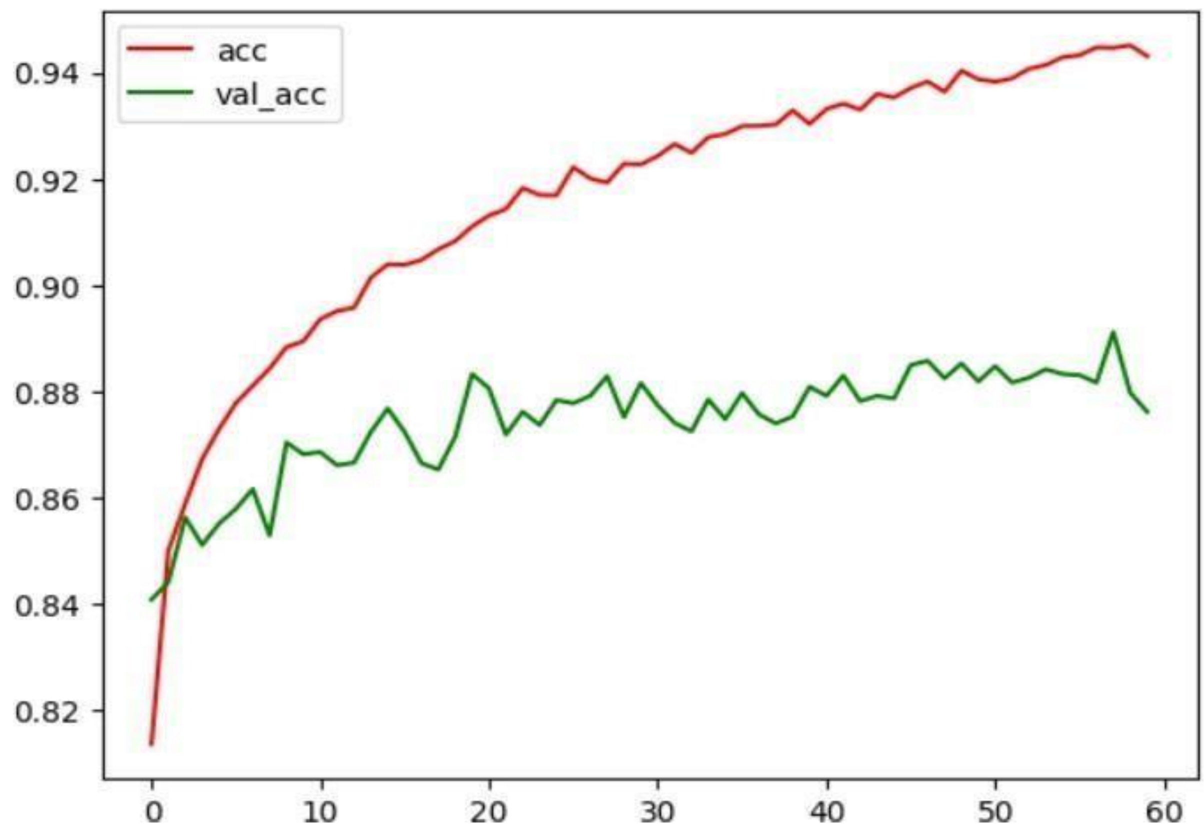

Fig 7.6: Accuracy graph against the number of iterations

CHAPTER 8

LIBRARIES USED

1. NUMPY
2. TENSORFLOW
3. PANDAS
4. KERAS
5. SKLEARN

9.1 LIBRARIES USED EXPLANATION

1. **NumPy** : NumPy is a popular image processing Python library with nd array to set and modify pixel values, trim images, concatenate images, and many more. One can complete multiple image processing without using other Python libraries.

2. **TensorFlow** : A popular deep learning framework that provides a variety of tools for building and training neural networks for image classification tasks.

3. **Pandas** : Pandas is an open-source library in Python that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series.

4. **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.

5. **sklearn :** scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.

CHAPTER 9

CONCLUSION

Through the utilization of the CIFAR-10 dataset, this image classification project has demonstrated a high level of efficiency in accurately categorizing a diverse set of images. The model's robust performance signifies the effectiveness of convolutional neural networks in tackling complex image classification tasks. With practical applications ranging from object recognition to scene understanding, the project's successful implementation illustrates the potential for solving real-life problems, autonomous driving, surveillance systems, and medical highlights the transformative role of image classification especially in the fields of imaging. This advancement in enabling automation and enhancing decision-making processes across various industries, ultimately contributing to a more technologically driven and efficient society.

Future Enhancement :

Fig. 10.1: Real Life implementation of our project for blind person