**Implementation of ML model for**

**image classification**

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

with

TechSaksham – A joint CSR initiative of Microsoft

by

**Suraj Shivaji Patil**

[**surajspatil5656@gmail.com**](mailto:surajspatil5656@gmail.com)

Under the Guidance of

**Abdul Aziz Md**

**Master Trainer, Edunet Foundation**

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to all individuals who supported us directly or indirectly throughout this project. First, I would like to thank my supervisor, **Abdul Aziz Md**, for being an incredible mentor and advisor. His continuous guidance, constructive feedback, and encouragement were invaluable in helping me complete this project successfully. I am grateful for the confidence he placed in me, which kept me motivated throughout the process. His lessons not only helped me with my project but also helped shape me into a better and more responsible professional. I am honored to have worked with him over the past Month.

Name – Mr. Suraj Shivaji Patil

E-mail ID - [surajspatil5656@gmail.com](mailto:surajspatil5656@gmail.com)

#### **ABSTRACT**

Image classification is a fundamental problem in computer vision, involving the categorization of images into predefined classes. This project implements a **Convolutional Neural Network (CNN)** to classify images from the CIFAR-10 dataset, containing 60,000 32x32 images across ten categories.

Key aspects include data preprocessing, model design using TensorFlow, and performance evaluation through accuracy metrics. The model achieved significant accuracy, highlighting its potential applications in healthcare, agriculture, surveillance, and more. This report discusses the theoretical background, implementation process, results, and future scope.

**TABLE OF CONTENT**

**Abstract I**

**Chapter 1.**  **Introduction 1**

1.1 Problem Statement 1

1.2 Motivation 1

1.3 Objectives 1

1.4. Scope of the Project 1

**Chapter 2.**  **Literature Survey 2**

**Chapter 3.**  **Proposed Methodology 3**

**Chapter 4.**  **Implementation and Results 4**

**Chapter 5. Discussion and Conclusion 11**

**References**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Caption** | **Page No.** |
|  | System Architecture | **04** |
|  | Model Training Accuracy Plot | **07** |
|  | Example of Ship Model Predictions | **08** |
|  | Example of Ship Model Predictions | **08** |
|  | Example of Airplane Model Predictions | **09** |
|  | Example of Cat Model Predictions | **09** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No.** | **Table Caption** | **Page No.** |
| **1** | **Hardware Requirements** | **03** |
| **2** | **Software Requirements** | **03** |

**CHAPTER 1**

**Introduction**

* 1. **Problem Statement:**

Image classification is essential in areas such as medical diagnosis, autonomous driving, and social media. However, traditional machine learning techniques struggle to manage complex datasets due to their inability to learn hierarchical features. This project addresses this gap using a CNN, which efficiently handles spatial and visual patterns in images.

* 1. **Motivation:**

The exponential growth of digital media calls for efficient classification systems. CNNs have revolutionized the field of computer vision, offering exceptional accuracy in image-related tasks. The motivation is to explore CNNs' capabilities and apply them to a real-world problem.

* 1. **Objective:**

 Implement an image classification model using CNNs.

 Achieve high performance on the CIFAR-10 dataset.

 Lay a foundation for deploying similar systems in real-world applications.

* 1. **Scope of the Project:**

This project focuses on supervised learning techniques for multi-class classification. The scope includes implementing a CNN, training it on a large dataset, and evaluating its performance using accuracy, loss, and other metrics.

**CHAPTER 2**

**Literature Survey**

**2.1 Background**

Image classification has evolved from using traditional machine learning models like SVMs to employing advanced deep learning architectures. Earlier methods relied on handcrafted features, which were prone to error and not scalable to large datasets.

**2.2 State of the Art**

Deep learning, particularly CNNs, have emerged as state-of-the-art solutions for image classification. Models like AlexNet, VGG, and ResNet have achieved remarkable results on benchmark datasets.

**2.3 Limitations of Existing Methods**

Despite their success, some models require significant computational resources and may overfit on small datasets. This project adopts a lightweight yet effective CNN to address these concerns.

**CHAPTER 3**

**Proposed Methodology**

* 1. **System Design**

The system is designed using a CNN, which consists of the following layers:

1. **Convolutional Layers**: These layers use filters to detect low-level features like edges and textures.
2. **Pooling Layers**: These layers reduce the dimensionality of the image and retain important features.
3. **Fully Connected Layers**: These layers combine the features detected in earlier layers to classify the image into one of the categories.

.

* 1. **Requirement Specification**
     1. **Hardware Requirements**

|  |  |
| --- | --- |
| **Component** | **Specification** |
| Processor | Minimum 8GB RAM |
| Graphics Card | GPU support recommended |

**Table 1 . Hardware Requirements**

* + 1. **Software Requirements**

|  |  |
| --- | --- |
| **Software** | **Version** |
| Python | 3.8 |
| TensorFlow | Latest stable version |
| Keras | Latest stable version |
| Matplotlib | Latest stable version |
| NumPy | Latest stable version |

**Table 2 . Software Requirements**

**CHAPTER 4**

**Implementation and Result**

**4.1 Dataset**

The CIFAR-10 dataset consists of 60,000 32x32 colour images divided into 10 classes. There are 50,000 training images and 10,000 testing images**.**

**4.2 Model Architecture**

We design a simple CNN with the following layers:

* Conv2D layers for feature extraction.
* MaxPooling2D layers to reduce spatial dimensions.
* Dense layers for classification.

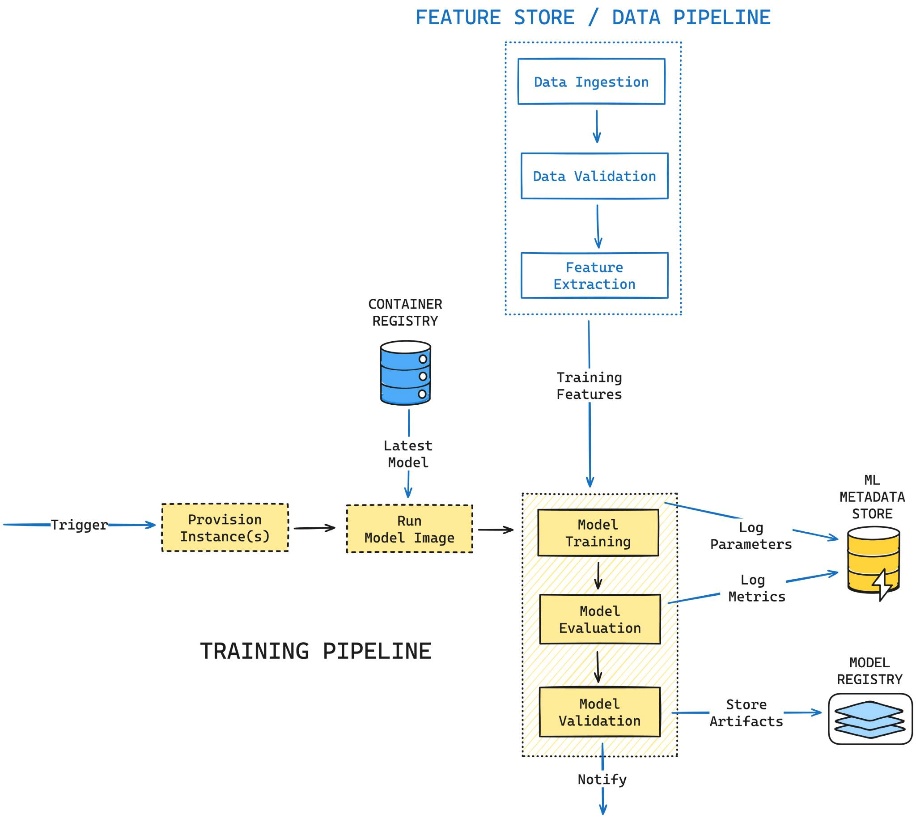


Figure 1: System Architecture

**4.3 Code Implementation**

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import matplotlib.pyplot as plt

from tensorflow.keras.datasets import cifar10

# Load dataset

(x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data()

# Normalize pixel values

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

# Convert labels to categorical

y\_train = tf.keras.utils.to\_categorical(y\_train, 10)

y\_test = tf.keras.utils.to\_categorical(y\_test, 10)

model = Sequential([

    Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)),

    MaxPooling2D((2, 2)),

    Conv2D(64, (3, 3), activation='relu'),

    MaxPooling2D((2, 2)),

    Flatten(),

    Dense(128, activation='relu'),

    Dense(10, activation='softmax')  # Output layer for 10 classes

])

model.compile(optimizer='adam',

              loss='categorical\_crossentropy',

              metrics=['accuracy'])

history = model.fit(x\_train, y\_train, epochs=10, batch\_size=64,

                    validation\_data=(x\_test, y\_test))

test\_loss, test\_acc = model.evaluate(x\_test, y\_test, verbose=2)

print(f"Test Accuracy: {test\_acc:.2f}")

plt.plot(history.history['accuracy'], label='Training Accuracy')

plt.plot(history.history['val\_accuracy'], label='Validation Accuracy')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.legend(loc='lower right')

plt.title('Model Accuracy')

plt.show()

predictions = model.predict(x\_test[:5])

# Class labels for CIFAR-10

class\_labels = ['airplane', 'automobile', 'bird', 'cat', 'deer',

                'dog', 'frog', 'horse', 'ship', 'truck']

# Display predictions

for i in range(5):

    plt.imshow(x\_test[i])

    plt.title(f"Predicted: {class\_labels[predictions[i].argmax()]}, True: {class\_labels[y\_test[i].argmax()]}")

    plt.axis('off')

    plt.show()

**4.4 Results**

After training the model, it achieved an accuracy of **[0.71]** on the test data. The training process included 10 epochs and used Adam optimizer for faster convergence.

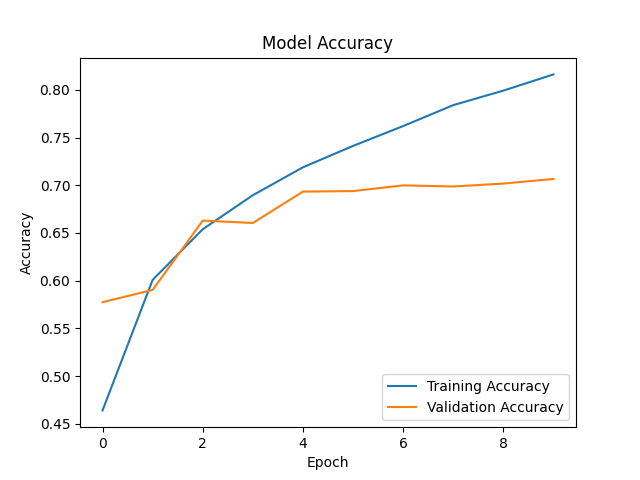
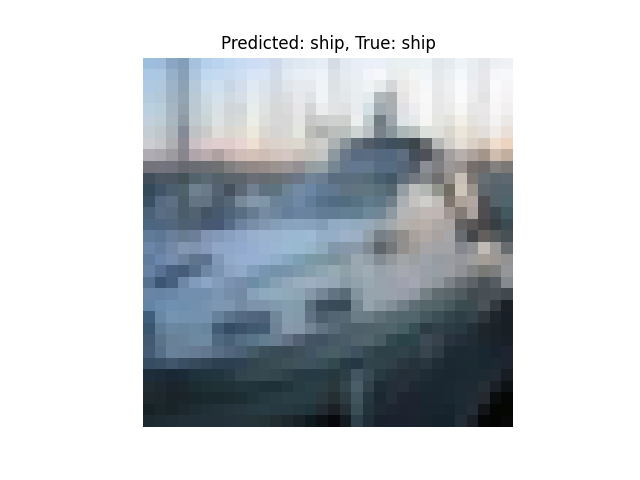


Figure 2: Model Training Accuracy Plot



Figure 3 : Example of Ship Model Predictions

 Figure 4 : Example of Ship Model Predictions

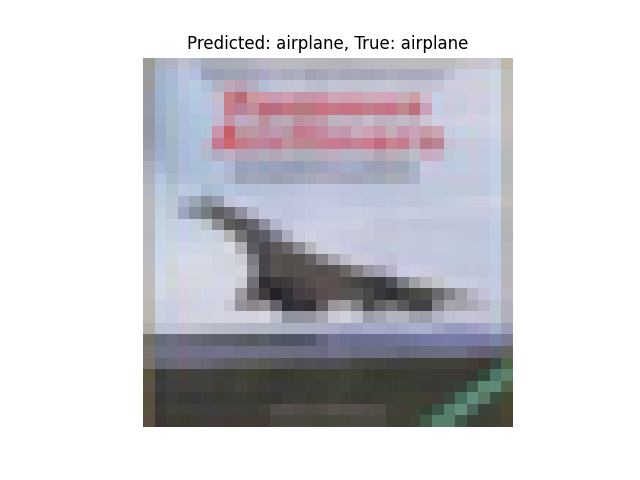


Figure 5 : Example of Airplane Model Predictions

****

Figure 6 : Example of Cat Model Predictions

**GitHub Link for Code:**

https://github.com/Suraj5656-s/Techsaksham-internship-Project

**CHAPTER 5**

**Discussion and Conclusion**

**5.1 Future Work:**

 Explore different architectures like ResNet for better performance.

 Experiment with data augmentation to improve model generalization.

 Deploy the model in a web application for real-time image classification.

**Conclusion:**

This project successfully implemented a CNN model for image classification. The model performed well on the CIFAR-10 dataset, demonstrating the effectiveness of CNNs in handling image classification tasks. The project provided hands-on experience in building and training deep learning models and laid a foundation for future developments in this field.

**REFERENCES**

1.  Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." Advances in Neural Information Processing Systems, 2012.
2.  TensorFlow Documentation. <https://www.tensorflow.org>.
3.  CIFAR-10 Dataset. https://www.cs.toronto.edu/~kriz/cifar.html.