Bharati Vidyapeeth (Deemed to be University)
Department of Engineering and Technology
Kharghar, Navi Mumbai

Department of Computer Science and Engineering (AIML)

Mini Project Report

On

IMAGE CLASSIFICATION SYSTEM

Subject-: - COMPUTING LAB-2

Presented By

Roll No.	Name	PRN
03	Smit More	2243110030
10	Rishi Chilveri	2243110037
18	Nirmalya Chatterjee	2243110046

Signature of Internal Examiner **Signature of External Examiner**

Bharati Vidyapeeth (Deemed to be University)
Department of Engineering and Technology
Kharghar, Navi Mumbai

Department of Computer Science and Engineering (AIML)

This is to certify that the project entitled, "**IMAGE CLASSIICATION SYSTEM** ", which is being submitted here with for the award of B.Tech., is the result of the workcompleted by **SMIT MORE** under my supervision and guidance within the four walls of theinstitute and the same has not been submitted elsewhere for the award of any degree.

(SUBJECT INCHARGE)

DEPARTMENT OF ENGINEERING & TECHNOLOGY OFFCAMPUS, KHARGHAR, NAVI MUMBAI

(HEAD OF DEPARTMENT)

DEPARTMENT OF ENGINEERING & TECHNOLOGY OFFCAMPUS KHARGHAR, NAVI MUMBAI

Bharati Vidyapeeth (Deemed to be University)
Department of Engineering and Technology
Kharghar, Navi Mumbai

Department of Computer Science and Engineering (AIML)

This is to certify that the project entitled, "**IMAGE CLASSIICATION SYSTEM** ", which is being submitted here with for the award of B.Tech., is the result of the workcompleted by **RISHI CHILVERI** under my supervision and guidance within the four walls of the institute and the same has not been submitted elsewhere for the award of any degree.

(SUBJECT INCHARGE)

DEPARTMENT OF ENGINEERING &
TECHNOLOGY OFFCAMPUS,
KHARGHAR, NAVI MUMBAI

(HEAD OF DEPARTMENT)

DEPARTMENT OF ENGINEERING &
TECHNOLOGY OFFCAMPUS
KHARGHAR, NAVI MUMBAI

Bharati Vidyapeeth (Deemed to be University)
Department of Engineering and Technology
Kharghar, Navi Mumbai

Department of Computer Science and Engineering (AIML)

This is to certify that the project entitled, "**IMAGE CLASSIICATION SYSTEM** ", which is being submitted here with for the award of B.Tech., is the result of the workcompleted by **NIRMALYA CHATTERJEE** under my supervision and guidance within the four walls of theinstitute and the same has not been submitted elsewhere for the award of any degree.

(SUBJECT INCHARGE)

DEPARTMENT OF ENGINEERING & TECHNOLOGY OFFCAMPUS,
KHARGHAR, NAVI MUMBAI

(HEAD OF DEPARTMENT)

DEPARTMENT OF ENGINEERING & TECHNOLOGY OFFCAMPUS
KHARGHAR, NAVI MUMBAI

Abstract

This mini-project focuses on the development of an Image Classification System that automatically identifies and classifies objects within images into predefined categories. Image classification is a key task in computer vision and has numerous applications in fields like healthcare, retail, security, and more.

The system has practical applications in various fields, such as healthcare and security, where fast and accurate image recognition is important. Future improvements could focus on enhancing model accuracy using techniques like data augmentation and transfer learning.

Index

Chapter 1

Introduction

Image classification is a fundamental task in computer vision that involves automatically identifying and categorizing objects in an image. With the advancement of artificial intelligence (AI) and machine learning, image classification has become an essential tool for various industries, including healthcare, retail, autonomous driving, and security. This mini project aims to develop a robust **Image Classification System** capable of classifying images into predefined categories based on their visual content.

The system utilizes **Convolutional Neural Networks (CNNs)**, a class of deep learning algorithms specifically designed for analyzing visual data. CNNs are highly effective in extracting hierarchical features from images, making them ideal for tasks such as object detection, face recognition, and image classification.

This Image Classification System can be extended to various applications, including automating tasks like medical diagnosis (classifying X-rays or MRI scans), product identification in e-commerce, or even real-time surveillance.

Chapter 2

System Design

❖ Model Architecture

- **Convolutional Layers**: Extract features from the images by applying filters to capture edges, textures, and shapes.
- **Pooling Layers**: Reduce the dimensionality of the data while retaining important features, improving computational efficiency.
- **Fully Connected Layers**: After the convolutional and pooling layers, the extracted features are flattened and passed through fully connected layers to make final predictions.
- **Output Layer**: The final layer applies a softmax function to classify the input image into one of the predefined categories.

❖ Model Training

- **Loss Function**: The cross-entropy loss function is used to quantify the difference between predicted and actual labels.
- **Optimizer**: An optimization algorithm like Adam or SGD (Stochastic Gradient Descent) is used to update the model weights during training.
- **Backpropagation**: The model learns through backpropagation, adjusting its parameters based on the error it makes during each iteration.

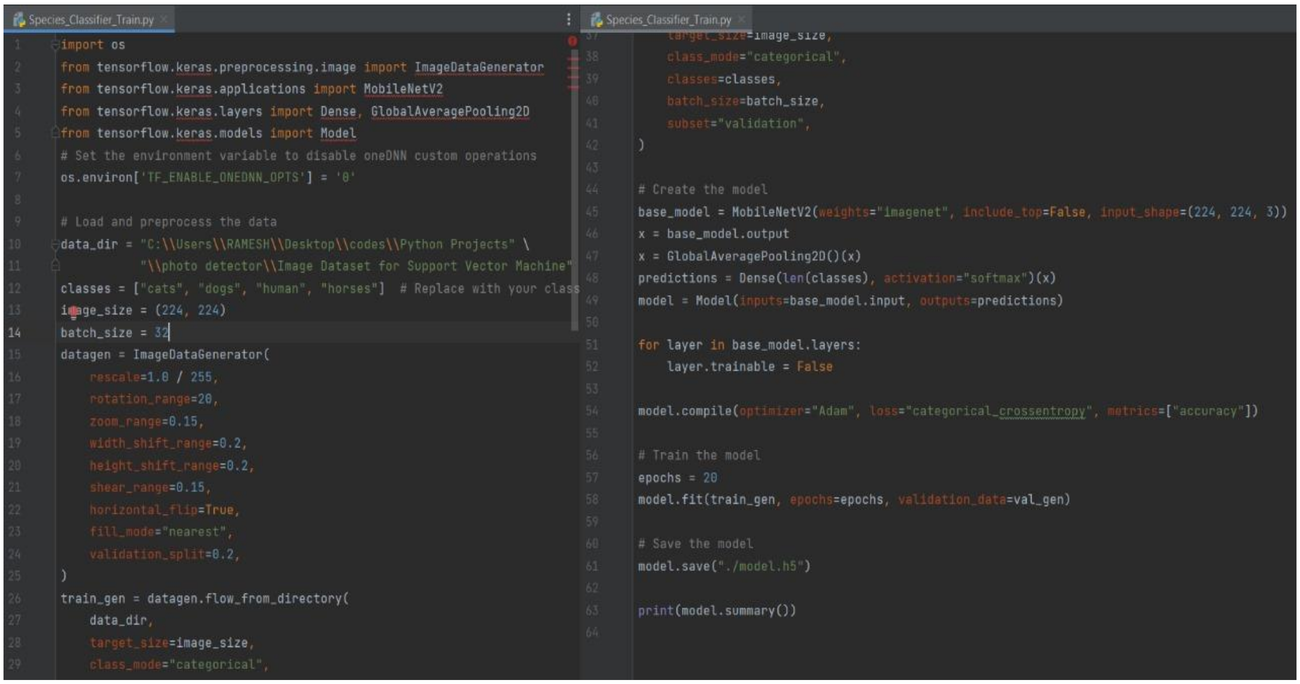
❖ Deployment

- Once the model is trained and evaluated, it can be deployed in real-world applications. Deployment options include integrating the model into a mobile or web app, where users can upload images for classification.

The system design provides a clear structure for building and deploying an Image Classification System, ensuring both scalability and accuracy in recognizing and classifying images.

Chapter 3

Implementation

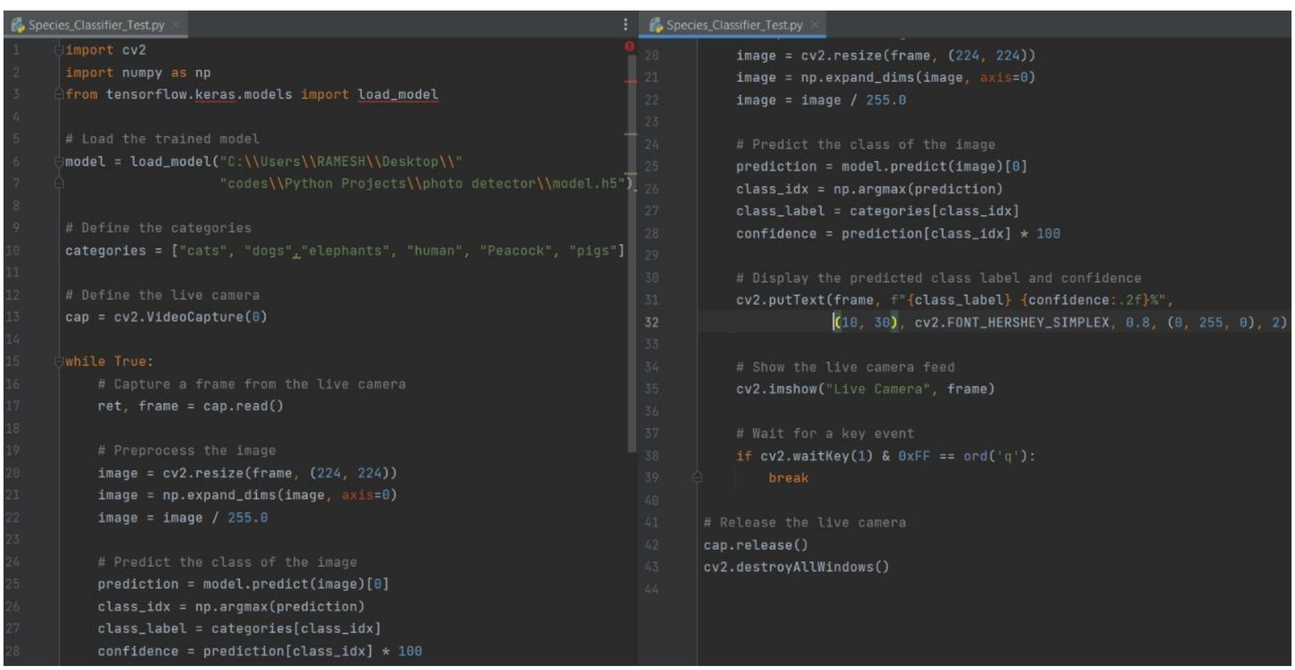


Species_Classifier_Train.py

```
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
# Set the environment variable to disable oneDNN custom operations
os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'

# Load and preprocess the data
data_dir = "C:\\Users\\RAMESH\\Desktop\\codes\\Python Projects" \
           "\\photo detector\\Image Dataset for Support Vector Machine"
classes = ["cats", "dogs", "human", "horses"]  # Replace with your class
image_size = (224, 224)
batch_size = 32
datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest",
    validation_split=0.2,
)
train_gen = datagen.flow_from_directory(
    data_dir,
    target_size=image_size,
    class_mode="categorical",
```

Species_Classifier_Train.py

```
    target_size=image_size,
    class_mode="categorical",
    classes=classes,
    batch_size=batch_size,
    subset="validation",
)

# Create the model
base_model = MobileNetV2(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
predictions = Dense(len(classes), activation="softmax")(x)
model = Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer="Adam", loss="categorical_crossentropy", metrics=["accuracy"])

# Train the model
epochs = 20
model.fit(train_gen, epochs=epochs, validation_data=val_gen)

# Save the model
model.save("./model.h5")

print(model.summary())
```



Species_Classifier_Test.py

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model

# Load the trained model
model = load_model("C:\\Users\\RAMESH\\Desktop\\"
                   "codes\\Python Projects\\photo detector\\model.h5")

# Define the categories
categories = ["cats", "dogs","elephants", "human", "Peacock", "pigs"]

# Define the live camera
cap = cv2.VideoCapture(0)

while True:
    # Capture a frame from the live camera
    ret, frame = cap.read()

    # Preprocess the image
    image = cv2.resize(frame, (224, 224))
    image = np.expand_dims(image, axis=0)
    image = image / 255.0

    # Predict the class of the image
    prediction = model.predict(image)[0]
    class_idx = np.argmax(prediction)
    class_label = categories[class_idx]
    confidence = prediction[class_idx] * 100
```

Species_Classifier_Test.py

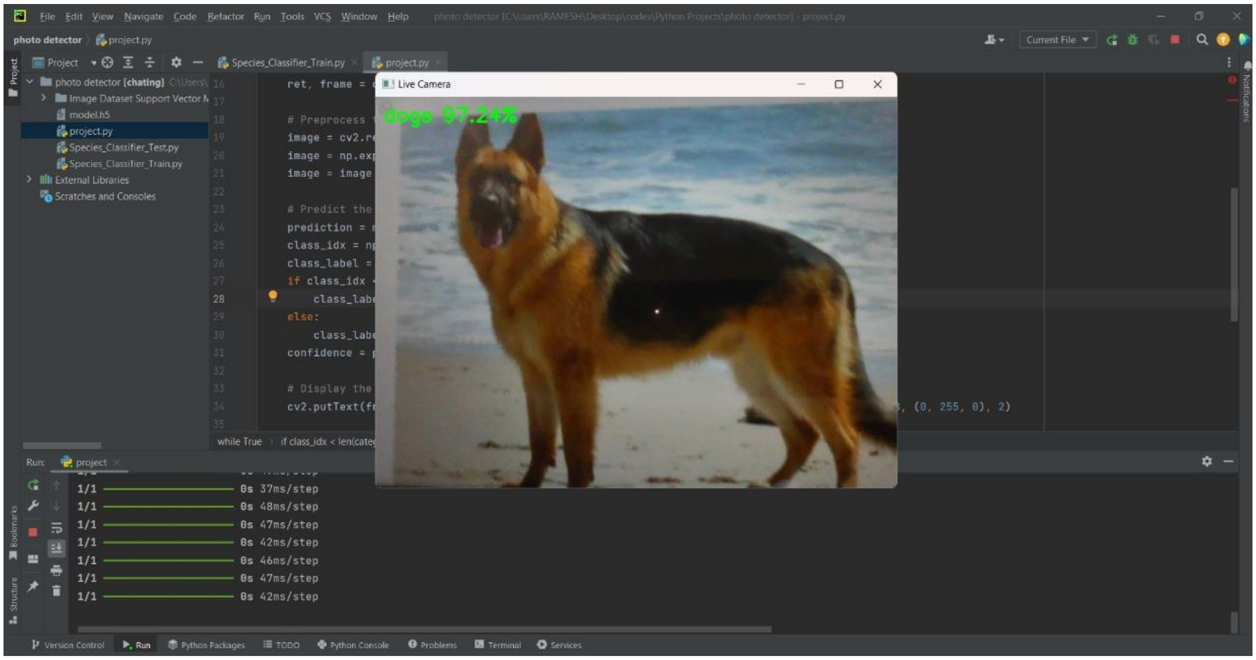
```
    image = cv2.resize(frame, (224, 224))
    image = np.expand_dims(image, axis=0)
    image = image / 255.0

    # Predict the class of the image
    prediction = model.predict(image)[0]
    class_idx = np.argmax(prediction)
    class_label = categories[class_idx]
    confidence = prediction[class_idx] * 100

    # Display the predicted class label and confidence
    cv2.putText(frame, f"{class_label} {confidence:.2f}%",
                (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2)

    # Show the live camera feed
    cv2.imshow("Live Camera", frame)

    # Wait for a key event
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the live camera
cap.release()
cv2.destroyAllWindows()
```

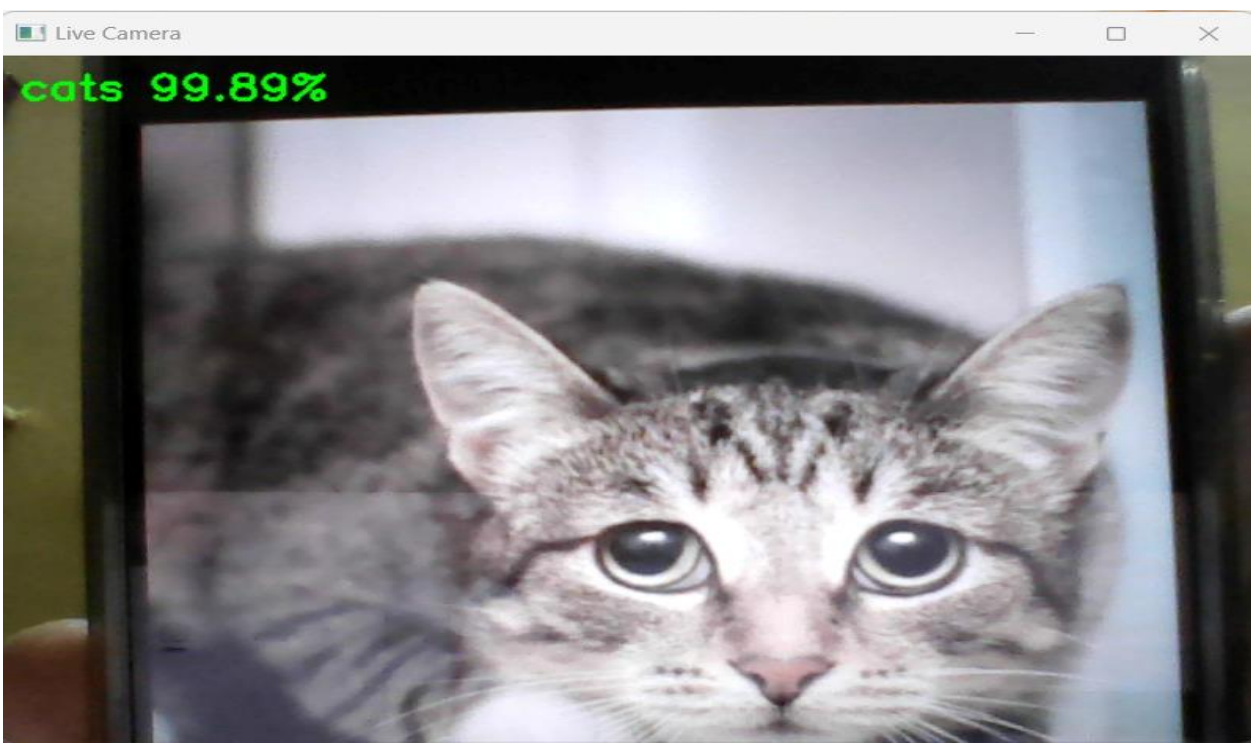
project test.py

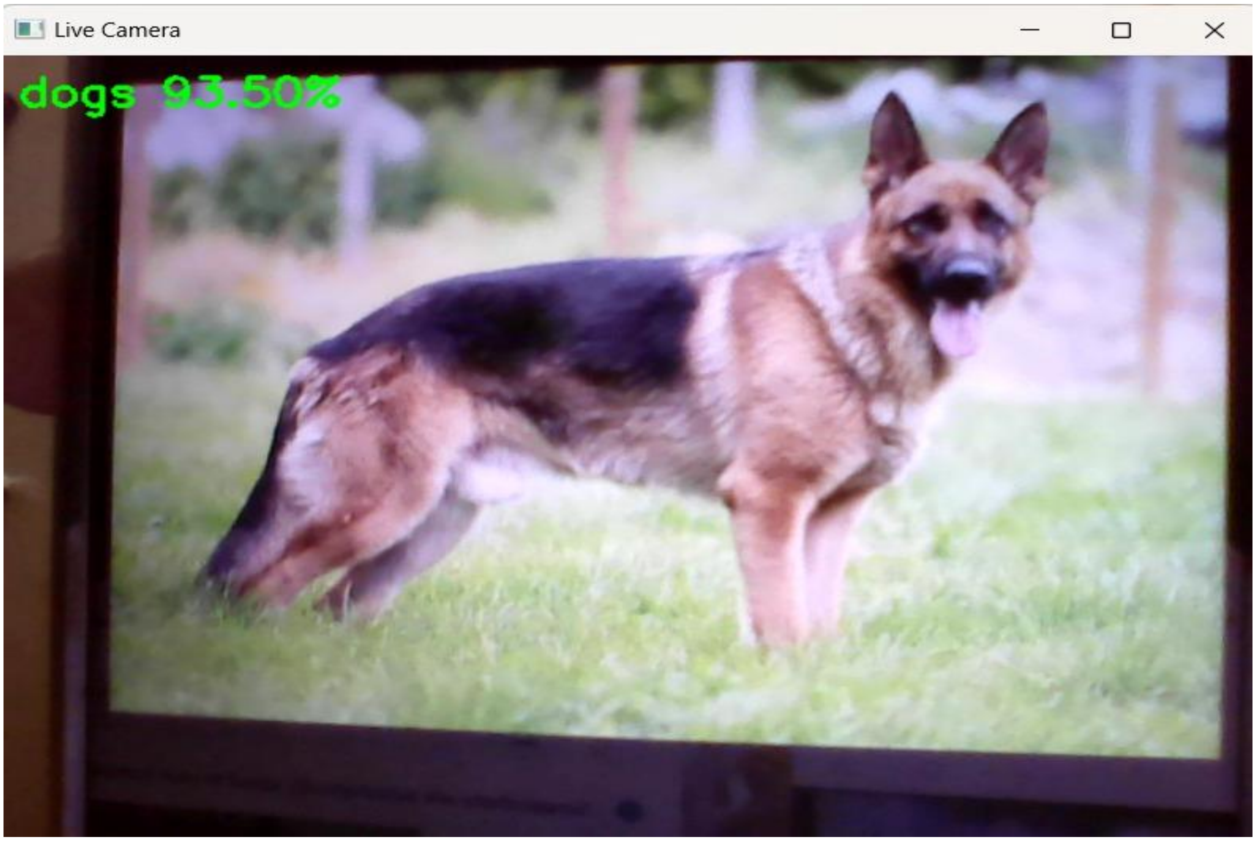
```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model

# Load the trained model
model = load_model("C:\\Users\\RAMESH\\Desktop\\"
                   "codes\\Python Projects\\photo detector\\model.h5")

# Define the categories
categories = ["cats", "dogs", "horses"]

# Define the live camera
cap = cv2.VideoCapture(0)

while True:
    # Capture a frame from the live camera
    ret, frame = cap.read()

    # Preprocess the image
    image = cv2.resize(frame, (224, 224))
    image = np.expand_dims(image, axis=0)
    image = image / 255.0

    # Predict the class of the image
    prediction = model.predict(image)[0]
    class_idx = np.argmax(prediction)
    class_label = categories[class_idx]
    confidence = prediction[class_idx] * 100

    # Display the predicted class label and confidence
    cv2.putText(frame, f"{class_label} {confidence:.2f}%",
(10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2)

    # Show the live camera feed
    cv2.imshow("Live Camera", frame)

    # Wait for a key event
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the live camera
cap.release()
cv2.destroyAllWindows()
```

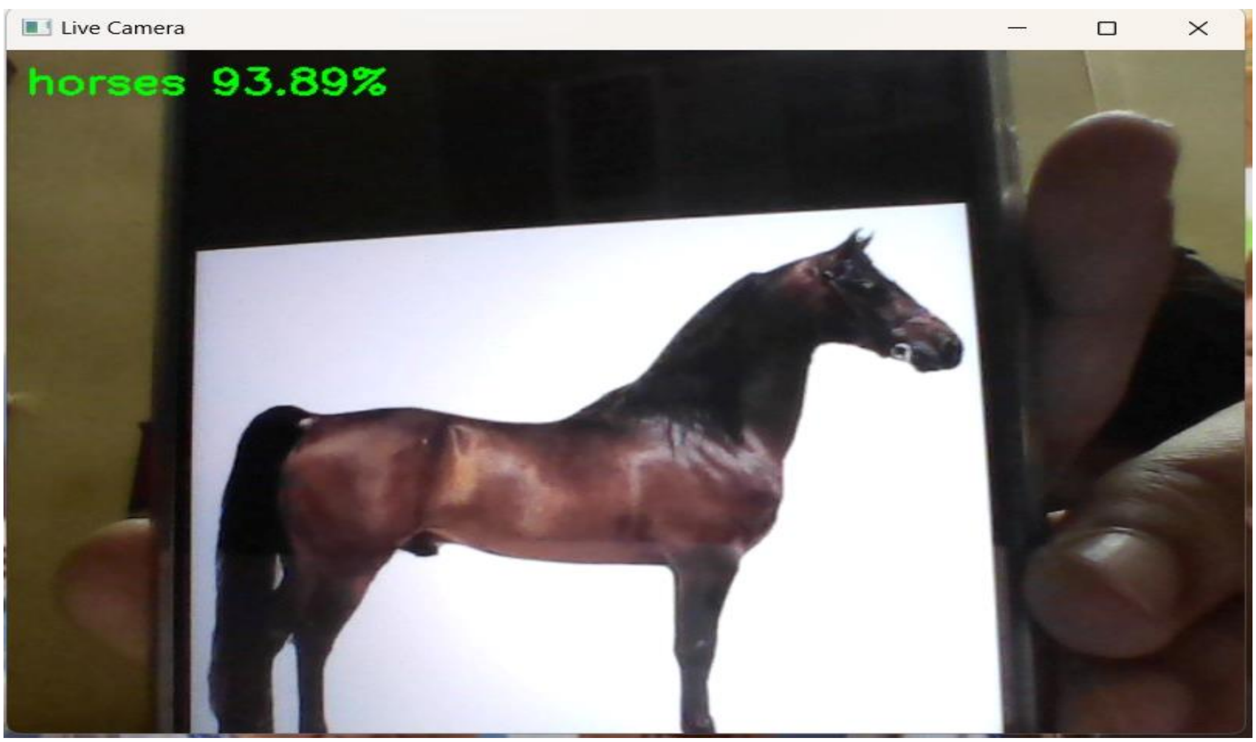
Chapter 4

Result/Output



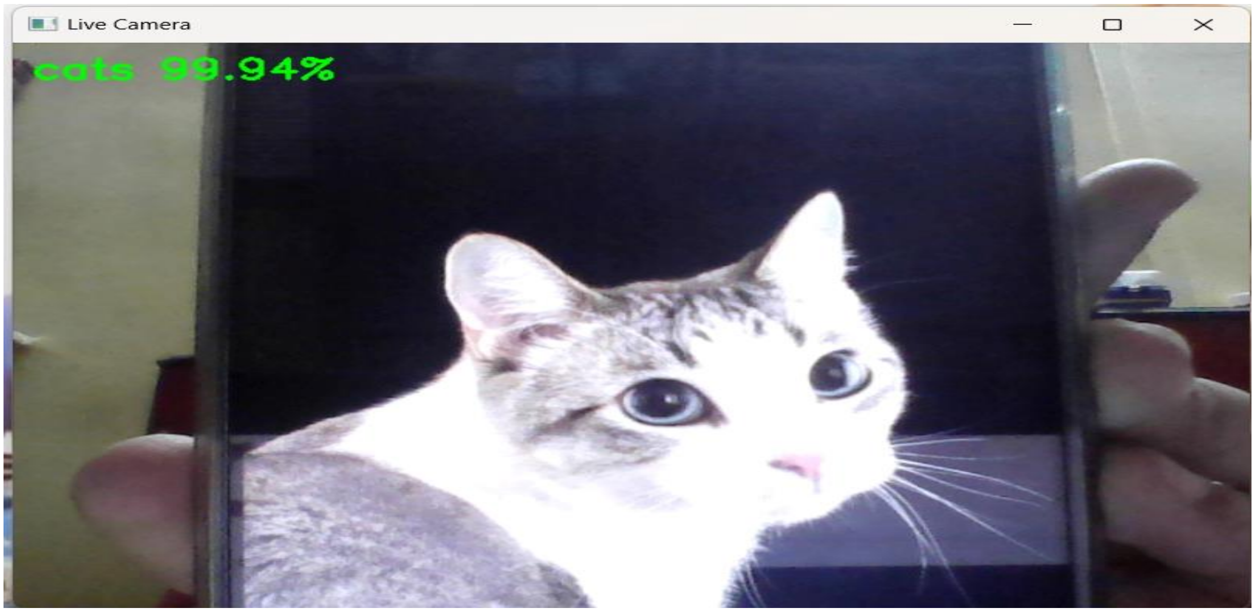


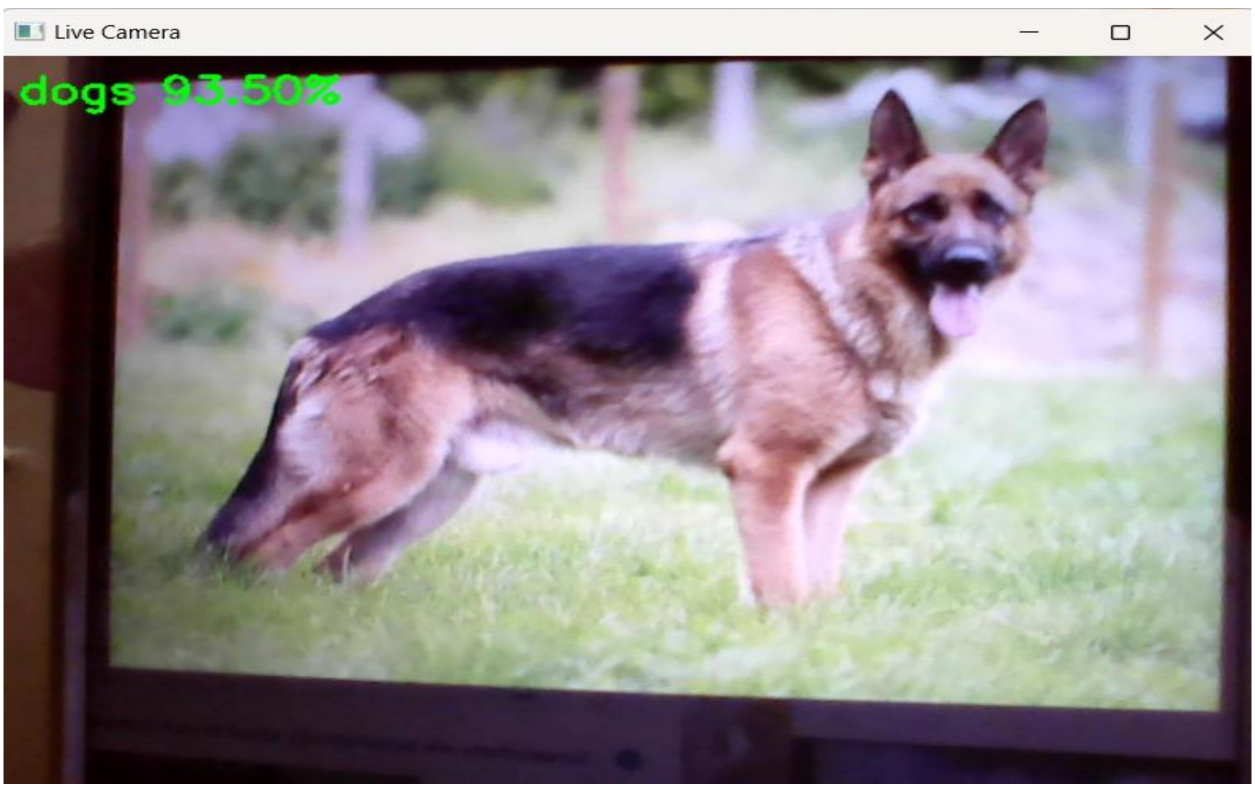

Live Camera
dogs 93.50%


Live Camera
horses 93.89%

Bharati Vidyapeeth (Deemed to be University)
Department of Engineering and Technology
Kharghar, Navi Mumbai

Department of Computer Science and Engineering (AIML)





Chapter 5

Conclusion

This mini-project successfully demonstrates the development and implementation of an Image Classification System using Convolutional Neural Networks (CNNs). By training the model on a labeled dataset, the system can accurately classify images into predefined categories, showcasing the power of deep learning in visual data analysis.

Throughout the project, key steps such as data preprocessing, model design, and evaluation were executed, leading to satisfactory results in terms of classification accuracy and other performance metrics. This system has potential applications in various industries, including healthcare, retail, and security, where automated image recognition is essential.

Future improvements could focus on enhancing accuracy with more complex models, leveraging larger datasets, and optimizing the system for real-world deployment. Overall, this project highlights the practical utility and effectiveness of machine learning techniques for image classification tasks.

Chapter 6

Reference

- GeeksforGeeks | A computer science portal for geeks
- Javatpoint Homepage
- Goodfellow, I., Bengio, Y., & Courville, A. (2016)
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012)
- Chollet, F. (2017)
- Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009)
- **Brownlee, J. (2019)**
- Russakovsky, O., Deng, J., Su, H., et al. (2015)