BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

AT

JODHPUR INSTITUTE OF ENGINEERING & TECHNOLOGY

NH-62, PALI ROAD, JODHPUR

SESSION: 2024-2025

A PROJECT REPORT

Image Classifier (Sad/Happy)

SUBMITTED BY:

Seetaram prajapat

CSE, III yr

SUBMITTED TO:

Harshita Khangarot

ASSISTANT PROFESSOR

Acknowledgement

We would like to acknowledge the contributions of the following people without whose help and guidance this project would not have been completed.

We are thankful to Ms. Mamta Garg, HOD(Mentor) and Anju Jangid, HOD(Academic) of Computer Science and Engineering Department, Jodhpur Institute of Engineering and Technology, for her constant encouragement, valuable suggestions and moral support and blessings.

We respectfully thank Mrs. Harshita Khangarot (Assistant Professor) of Computer Science and Engineering Department, Jodhpur Institute of Engineering and Technology, for providing me an opportunity to do this project work and giving me all support and guidance, which made me complete the project up to very extent.

Although it is not possible to name individually, we shall ever remain indebted to the faculty members of Jodhpur Institute of Engineering and Technology, for their persistent support and cooperation extended during his work.

This acknowledgement will remain in complete if we fail to express our deep sense of obligation to our parents and God for their consistent blessings and Encouragement

Table of Contents

Introduction

This project report details the development and evaluation of an image classifier specifically designed to distinguish between images of happy and sad human facial expressions. The primary objective of this project is to design and implement an image classification model capable of accurately performing this binary classification task. To achieve this, the project will explore the use of Convolutional Neural Networks (CNNs), a state-of-the-art technique in image classification, and evaluate its performance using relevant metrics such as accuracy, precision, recall.

The report will also delve into the potential applications of such a classifier in specific domains, including but not limited to human-computer interaction and well-being monitoring. Furthermore, it will address the inherent challenges and limitations associated with facial emotion recognition, acknowledging the complexities of human emotion expression and the ethical considerations surrounding this technology. The scope of this report encompasses the methodology employed for data collection and model development, the results obtained from the evaluation, a detailed analysis of these results, and the conclusions drawn regarding the feasibility and potential of the developed happy/sad facial expression classifier.

Technology Used in Project

Tools Used:

- **Jupyter Notebook / Google Colab:** Interactive environments for data exploration, analysis, and model development, allowing for code execution and visualization within a web browser.
- **Visual Studio:** A comprehensive integrated development environment (IDE) used for building and deploying applications, providing tools for coding, debugging, and project management.

Technologies Used:

- **Machine Learning:** A field of artificial intelligence that enables systems to learn from data without explicit programming, for tasks like prediction and classification.
- **Deep Learning:** A subfield of machine learning that uses artificial neural networks with multiple layers to learn complex patterns from large datasets.

Libraries Used:

- **NumPy:** A fundamental library for numerical computing in Python, providing support for arrays and mathematical operations.
- **Pandas:** A library for data manipulation and analysis, offering data structures like Data Frames for efficient data handling.
- **Scikit-learn:** [1] A machine learning library that provides tools for classification, regression, clustering, and model evaluation.
- **Matplotlib:** A plotting library for creating static, interactive, and animated visualizations in Python.
- **TensorFlow/Keras:** Deep learning frameworks for building and training neural networks, enabling complex model development.
- **Seaborn:** A data visualization library based on Matplotlib, providing high-level interfaces for creating informative statistical graphics.
- **Flask:** A lightweight web framework for building web applications in Python, used for deploying machine learning models as web services.

Requirements

- Python 3.x
- TensorFlow / Keras
- NumPy
- Pandas
- flask
- Matplotlib
- Scikit-learn
- IPL Score dataset

Methodology

- **Input:** Labeled face images (happy/sad).
- **Preprocessing:**
  - Face detection.
  - Resizing, normalization.
  - Data augmentation.

- **Feature Extraction:**
  - Convolutional Neural Network (CNN) learns features automatically.

- **Classification:**
  - CNN's final layers classify based on learned features.

- **Output:** Happy/Sad prediction.
- **Evaluation:** Accuracy, precision, recall, confusion matrix.
- **Key Features (Learned by CNN):**
  - Mouth shape (corners up/down).
  - Eye shape and wrinkles.
  - Eyebrow position

CODE

App.py

```
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model # type: ignore
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import cv2
import os

app = Flask(__name__)

model = load_model('models/imageclassifier.h5')

UPLOAD_FOLDER = "static/uploads"
if not os.path.exists(UPLOAD_FOLDER):
    os.makedirs(UPLOAD_FOLDER)

app.config["UPLOAD_FOLDER"] = UPLOAD_FOLDER

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
```

```
    if 'image-pred' not in request.files:
        return "No file uploaded!", 400

    file = request.files['image-pred']

    if file.filename == '':
        return "No selected file!", 400

    # Save uploaded file to 'static/uploads/'
    filepath = os.path.join(app.config["UPLOAD_FOLDER"], file.filename)
    file.save(filepath)

    # Read image using OpenCV
    img = cv2.imread(filepath)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (256, 256))
    img = img / 255.0
    img = np.expand_dims(img, axis=0)

    # Model Prediction
    pred = model.predict(img)
    prediction = "Sad" if pred > 0.5 else "Happy"

    return render_template('index.html', image_pred=file.filename, prediction_text=f"Prediction:
{prediction}")

if __name__ == "__main__":
    app.run(debug=True)
```

OUTPUT

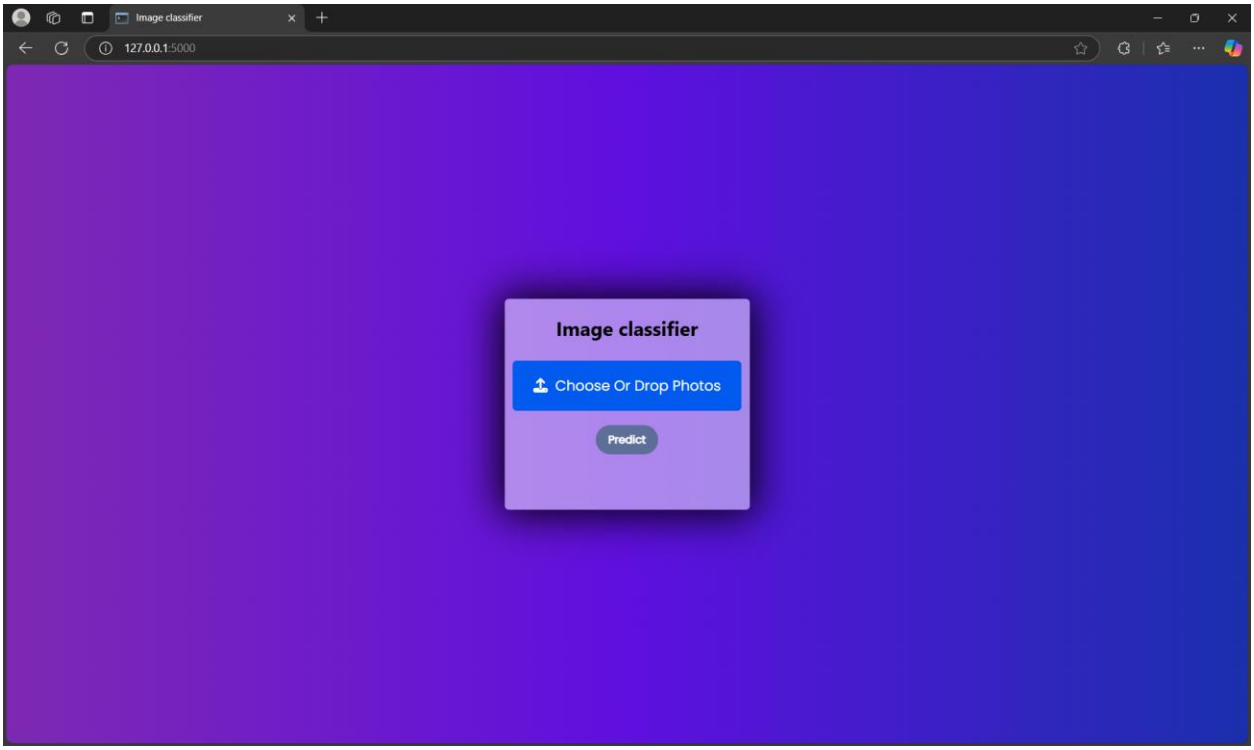
Image classifier
127.0.0.1:5000
Image classifier
Choose Or Drop Photos
Predict

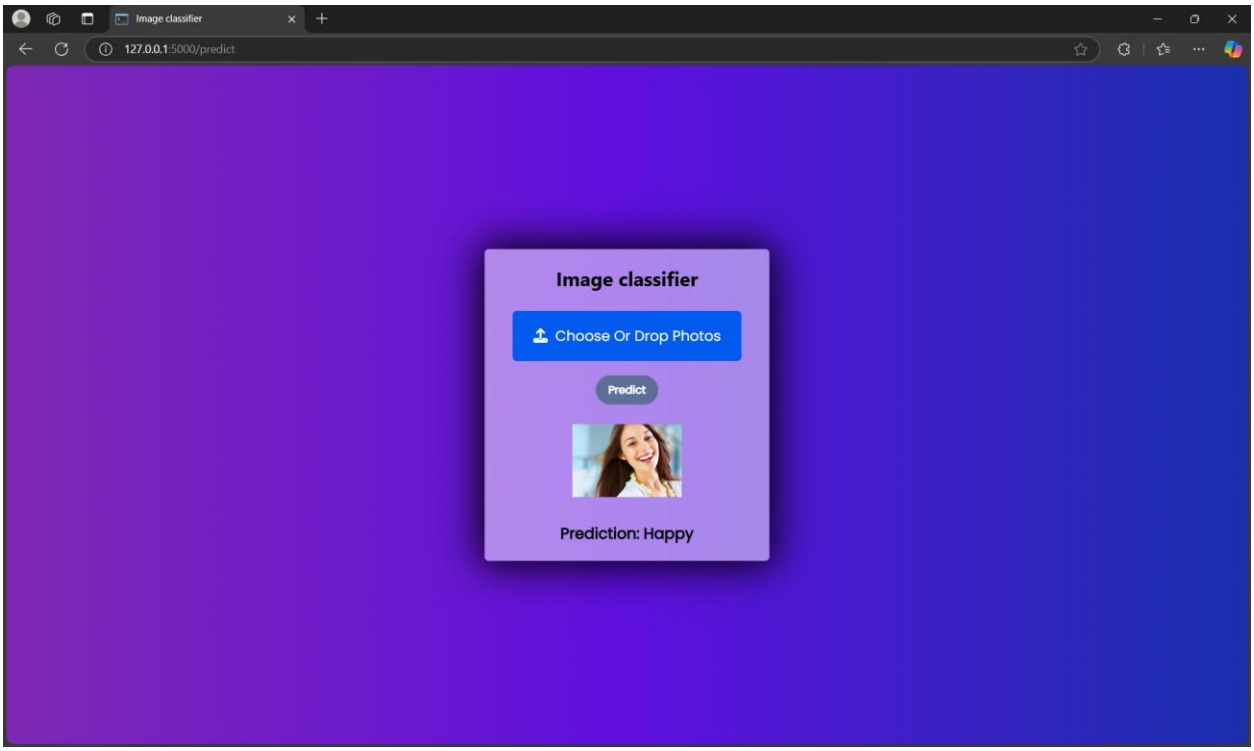
Image classifier
127.0.0.1:5000/predict
Image classifier
Choose Or Drop Photos
Predict
Prediction: Happy

Conclusion

In conclusion, this deep learning project successfully developed an image classifier capable of distinguishing between happy and sad facial expressions. By leveraging a Convolutional Neural Network (CNN), the model effectively learned relevant features from input images, achieving accurate classification. This demonstrates the power of deep learning for facial emotion recognition and its potential applications in various fields, including human-computer interaction, mental health analysis, and social robotics.

REFERENCES

- W3Schools - Python: https://www.w3schools.com/
- Python Documentation: https://docs.python.org: Comprehensive reference for Python libraries and functions.
- Matplotlib & Seaborn Documentation: https://matplotlib.org/stable/contents.html – Official references for data visualization libraries used in the UI.
- Scikit learn: https://scikit-learn.org/stable/modules/preprocessing.html
- TensorFlow: https://www.tensorflow.org/tutorials