Project Report Of AIDS-2 Lab

Topic: Bird Species Recognition System

Group Members

5.Yajat Bhasin

29. Kaushik Kotian

33. Mohit Mandhyani

67.Swapnil Mulani

Title: Bird Species Recognition System

DESCRIPTION:

This project simplifies bird species identification using machine learning, offering an efficient alternative to traditional methods. By leveraging convolutional neural networks (CNNs), the system accurately classifies bird species from images, overcoming challenges like subtle appearance differences and environmental factors. The model's performance is evaluated through metrics like accuracy, precision, recall, and F1-score. Users can also contribute new bird images, helping improve the model over time. This system not only enhances bird watching by automating identification but also evolves with user input, making it more accurate and reliable.

OBJECTIVES

1. **Implementing Data Augmentation:** Another objective might be to implement data augmentation techniques to increase the diversity and robustness of the training dataset, thereby enhancing the model's ability to generalize to unseen data.
2. **Fine-Tuning for Improved Performance:** The project could aim to fine-tune the pre-trained model to adapt it to the specific task of bird classification, with the objective of further improving performance metrics such as accuracy and precision.
3. **Provide a User-Friendly Interface**: Create a simple and interactive platform for bird watchers and researchers to easily upload images and receive species identification results.
4. **Support Conservation Efforts**: Assist in wildlife research and conservation by providing a quick, accurate tool for species identification, aiding in tracking and studying bird populations.

Convolutional Neural Networks For Bird Species Recognition:

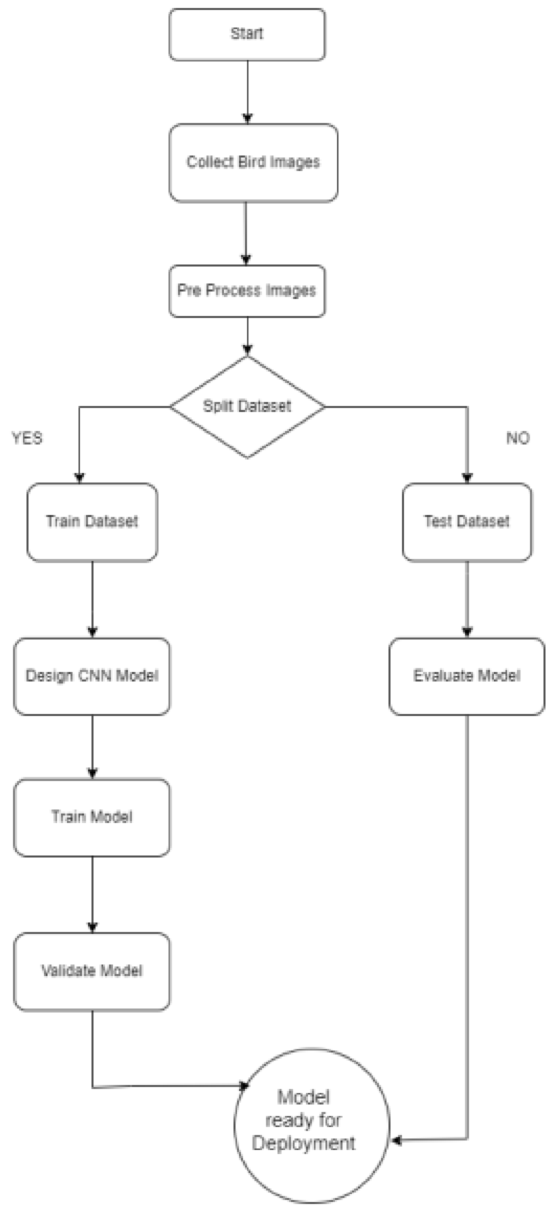
A Convolutional Neural Network (CNN) is used in this bird species classification project to automatically extract essential features like patterns, colors, and shapes from images, capturing complex spatial relationships that aid accurate identification across varying conditions. EfficientNetB4 is selected for its superior accuracy and efficiency, offering a balanced trade-off between model depth, width, and resolution through compound scaling. Compared to traditional CNNs like VGG16 and ResNet-18, EfficientNetB4 achieves higher performance with fewer parameters, reduced computational costs, and better scalability. Its robustness makes it well-suited for handling variations in bird postures, backgrounds, and lighting conditions, ensuring more reliable classification results.

How it works:

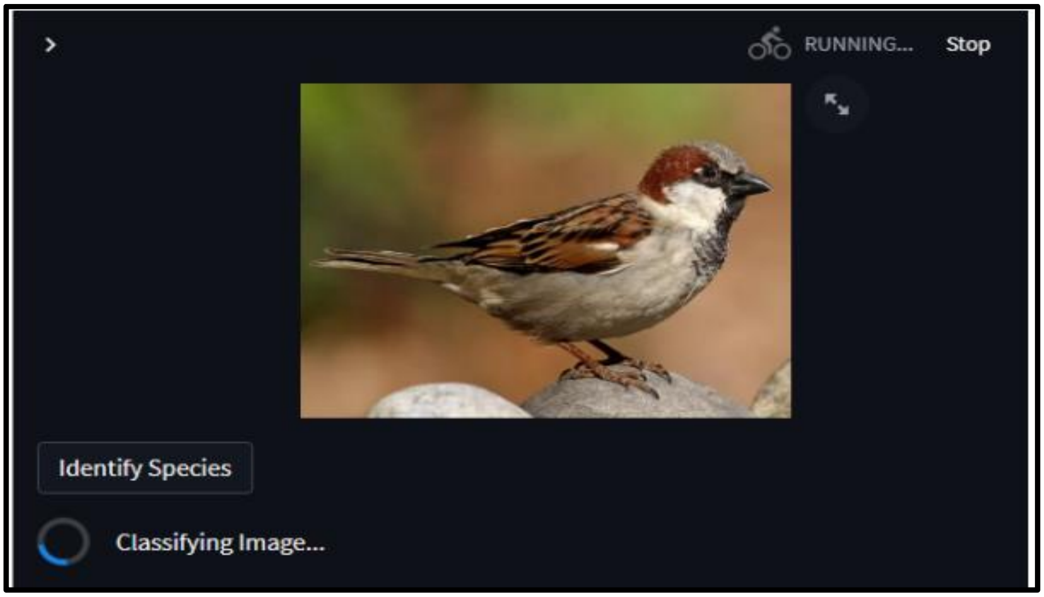
1. **Image Preprocessing**: The input bird images are resized and normalized to a standard format suitable for the model. This step ensures that all images have consistent dimensions and pixel values, preparing them for the CNN.
2. **Feature Extraction via CNN Layers**: EfficientNetB4 processes the images through multiple convolutional layers, extracting key features such as patterns, colors, textures, and shapes. These layers help the model learn important details that differentiate one bird species from another.

3. **Classification**: The extracted features are passed through fully connected (dense) layers, where the model classifies the bird based on the learned patterns. The model outputs a probability distribution across all possible bird species.
4. **Model Evaluation and Improvement**: The system evaluates its predictions using accuracy, precision, recall, and F1-score metrics. Users can also contribute new bird images, allowing the model to improve over time through continuous learning.

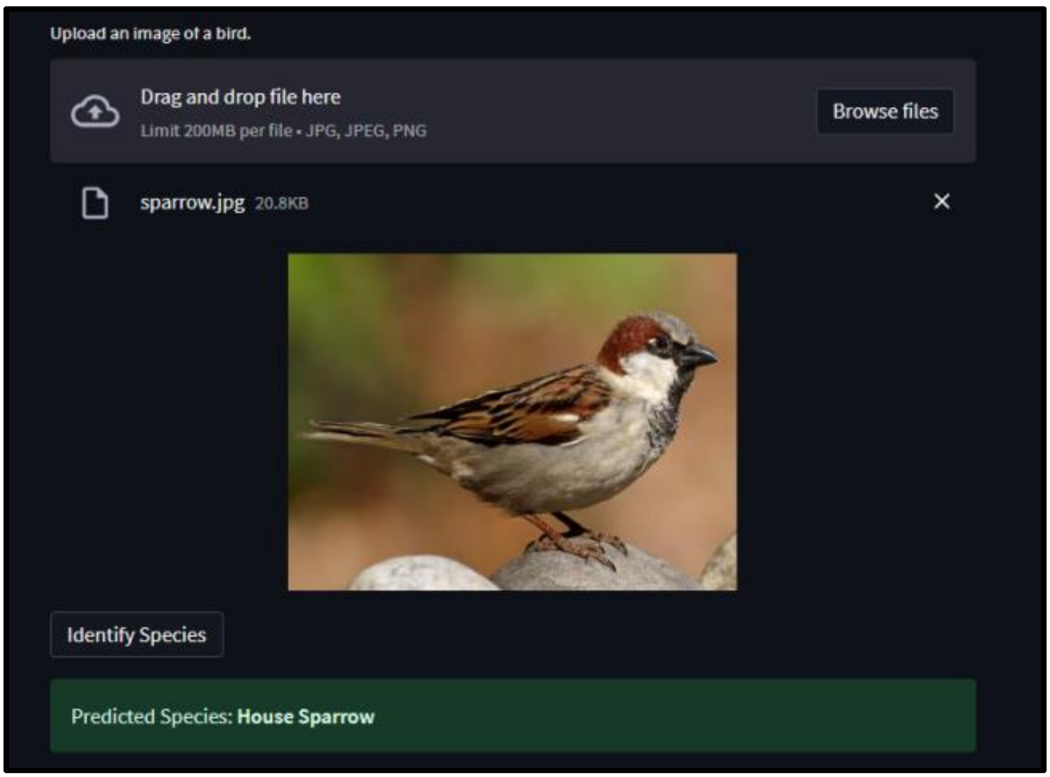
BLOCK DIAGRAM:



IMPLEMENTATION:



EXPECTED OUTPUT:



CODE:

```
import streamlit as st
import tensorflow as tf
import pandas as pd
import plotly.graph_objects as go
import json
import wikipedia
from PIL import Image
import io
import base64

# Run the app locally using: streamlit run app.py
st.set_page_config(page_title="What Bird is That?")  # Set tab title

# Hide hamburger menu and Streamlit watermark
hide_streamlit_style = """
<style>
#MainMenu {visibility: hidden;}
footer {visibility: hidden;}
</style>
"""
st.markdown(hide_streamlit_style, unsafe_allow_html=True)

# Functions
@st.cache_data
def load_label_to_scientific(json_filename: str = "common_to_scientific.json") -> dict:
    """Loads a JSON file containing the label to scientific name mapping."""
    with open(f"models_and_data/{json_filename}", "r") as json_file:
        label_to_scientific_dict = json.load(json_file)
    return label_to_scientific_dict

def display_bird_summary(best_guess_row: pd.Series) -> None:
    """Displays the bird summary, including the Wikipedia description and an image."""
    wiki_description = get_bird_description(best_guess_row["Scientific Name"])
    if wiki_description:
        other_image = Image.open(f"models_and_data/sample_photos/{best_guess_row['Common 
Name']}.jpg")
        image_width = 300
        image_bytes = io.BytesIO()
        other_image.save(image_bytes, format="JPEG")
```

```
        image_html = f'<img
src="data:image/jpeg;base64,{base64.b64encode(image_bytes.getvalue()).decode()}"' \
                     f'alt="Bird Image" style="float:right;width:{image_width}px;margin-left:20px;">'
        st.markdown(f'{image_html}{wiki_description}', unsafe_allow_html=True)
        st.info(f'Read more about the [{best_guess_row["Common Name"].title()} on
Wikipedia]({best_guess_row["Wikipedia Link"]})')

@st.cache_resource
def prep_image(img: bytes, shape: int = 260, scale: bool = False) -> tf.Tensor:
    """Preprocess the image data."""
    img = tf.image.decode_image(img, channels=3)
    img = tf.image.resize(img, size=([shape, shape]))
    if scale:
        img = img / 255.0
    return img

def classify_image(img: bytes, model: tf.keras.Model) -> pd.DataFrame:
    """Classifies the given image using the provided model and returns top 3 predictions."""
    img = prep_image(img)
    img = tf.cast(tf.expand_dims(img, axis=0), tf.int16)
    pred_probs = model.predict(img)
    top_3_indices = sorted(pred_probs.argsort()[0][-3:][::-1])
    values = pred_probs[0][top_3_indices] * 100
    labels = [class_labels[i] for i in top_3_indices]
    prediction_df = pd.DataFrame({"Common Name": labels, "Probability": values})
    return prediction_df.sort_values("Probability", ascending=False)


def add_wikipedia(input_df: pd.DataFrame) -> pd.DataFrame:
    """Adds Wikipedia links to the DataFrame."""
    input_df["Scientific Name"] = input_df["Common Name"].map(labels_to_sci)
    input_df["Wikipedia Link"] = input_df["Scientific Name"].apply(
        lambda species_name: 'https://en.wikipedia.org/wiki/' + species_name.lower().replace(' ',
'_')
    )
    return input_df

# Load label to scientific name mapping
labels_to_sci = load_label_to_scientific()
class_labels = sorted(list(labels_to_sci.keys()))
```

```
# Main Body content
st.title("What Bird is That?")
st.header("Identify what kind of bird you snapped a photo of!")
file = st.file_uploader(label="Upload an image of a bird.", type=["jpg", "jpeg", "png"])

if not file:
    st.button("Identify Species", disabled=True, help="Upload an image to make a prediction")
    st.stop()
else:
    image = file.read()
    col1, col2, col3 = st.columns([1, 2, 1])
    with col2:
        st.image(image, use_column_width="auto", width="50%")
    pred_button = st.button("Identify Species")

if pred_button:
    with st.spinner("Loading Image Classification Model..."):
        model = load_model()
    with st.spinner("Classifying Image..."):
        df = classify_image(image, model)
        df = add_wikipedia(df)
        top_prediction_row = df.iloc[0]
        st.success(f'Predicted Species: **{top_prediction_row["Common Name"].title()}**')
        display_bird_summary(top_prediction_row)
```

CONCLUSION:

This project involves building and training a deep learning model for bird species classification using transfer learning with the EfficientNetB4 architecture. The script loads a dataset of bird images, preprocesses them, and constructs a neural network model. The model is initially trained on the dataset, followed by fine-tuning for improved performance. Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to assess the model’s performance on validation and test sets. Finally, the trained model is saved for future use. Overall, this project aims to accurately identify bird species from images using deep learning techniques..