Multi Class Animal Recognition

VINNAKOTA NITISH RAJ
GITAM(Deemed to be)University
STU65edeb509fb961710091088

Learning Objectives

- Understood how Convolutional Neural Networks (CNNs) identify complex image patterns.
- Learned transfer learning using pre-trained models (MobileNetV2).
- Applied TensorFlow for building and training deep learning models.
- Gained hands-on experience with image preprocessing, augmentation, and classification.
- Deployed the trained model via a Streamlit-based web application.
- Explored model optimization, generalization, and practical deployment aspects.

edunet foundation

Tools and Technology used

- **Programming Language**: Python
- **Libraries/Frameworks**: TensorFlow, Keras, OpenCV, NumPy, Matplotlib, Streamlit
- **Model**: MobileNetV2 (Pretrained CNN via Transfer Learning)
- **Development Environment**: Google Colab (Training), Streamlit (Deployment)
- **Dataset Source**: Kaggle - Animal Image Dataset (90 animal species)
- **Other Tools**: PIL, cv2 for image handling, OS for class loading

Methodology

- **Step 1**: Collected and organized animal image dataset from Kaggle (90 classes).
- **Step 2**: Applied image preprocessing (resize, RGB conversion, normalization).
- **Step 3**: Performed image augmentation to enhance model robustness.
- **Step 4**: Chose MobileNetV2 for its efficiency in transfer learning.
- **Step 5**: Built and trained the CNN model on Colab using TensorFlow.
- **Step 6**: Evaluated model performance and exported .keras and class files.
- **Step 7**: Developed a web interface using Streamlit for real-time predictions.
- **Step 8**: Integrated the model with the frontend and tested on various species.

Problem Statement:

How can we build a real-time, accurate animal species recognition system that works reliably across diverse animal types and environmental conditions using deep learning?

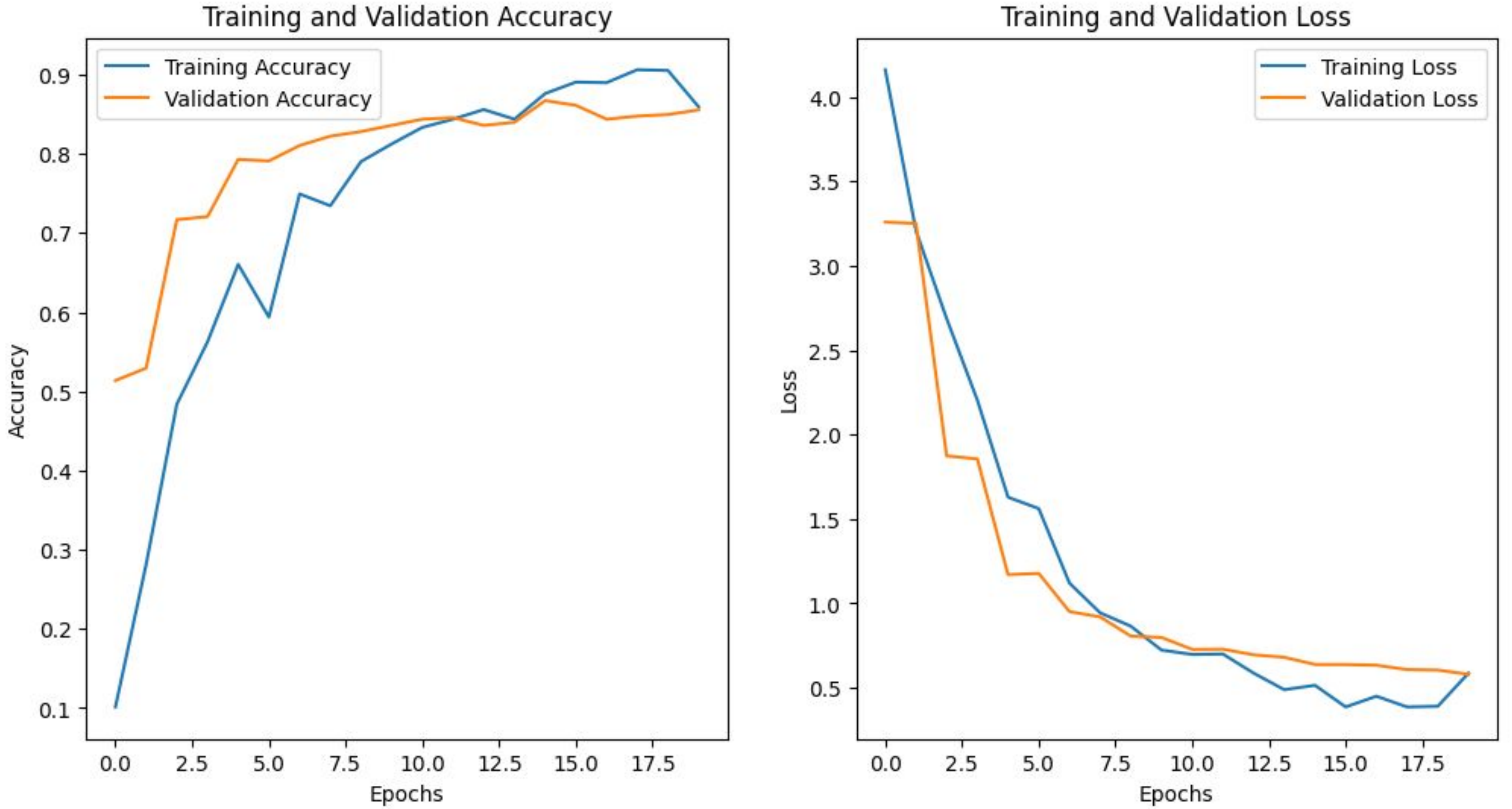
Solution:

Developed a deep learning-based animal classification system using Convolutional Neural Networks. Leveraged the **pretrained MobileNetV2 model** for transfer learning to extract robust features from animal images. The output layer was configured with **softmax activation** to classify **90 distinct animal categories**. The model was trained using TensorFlow on Google Colab and exported for deployment. A clean and responsive **Streamlit interface** was built to allow users to upload an image and receive instant, real-time predictions.

Screenshot of Output:

```
# Load MobileNetV2 as the base model
base_model = MobileNetV2(
    weights='imagenet',
    include_top=False,
    input_shape=(224, 224, 3)
)

# Freeze the base model weights, training
base_model.trainable = False

# Add custom layers on top of the base model
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(num_classes, activation='softmax')(x) # output layer

# Create the final model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.0001),
              loss = 'categorical_crossentropy',
              metrics = ['accuracy'])
# Model summary
model.summary()
```

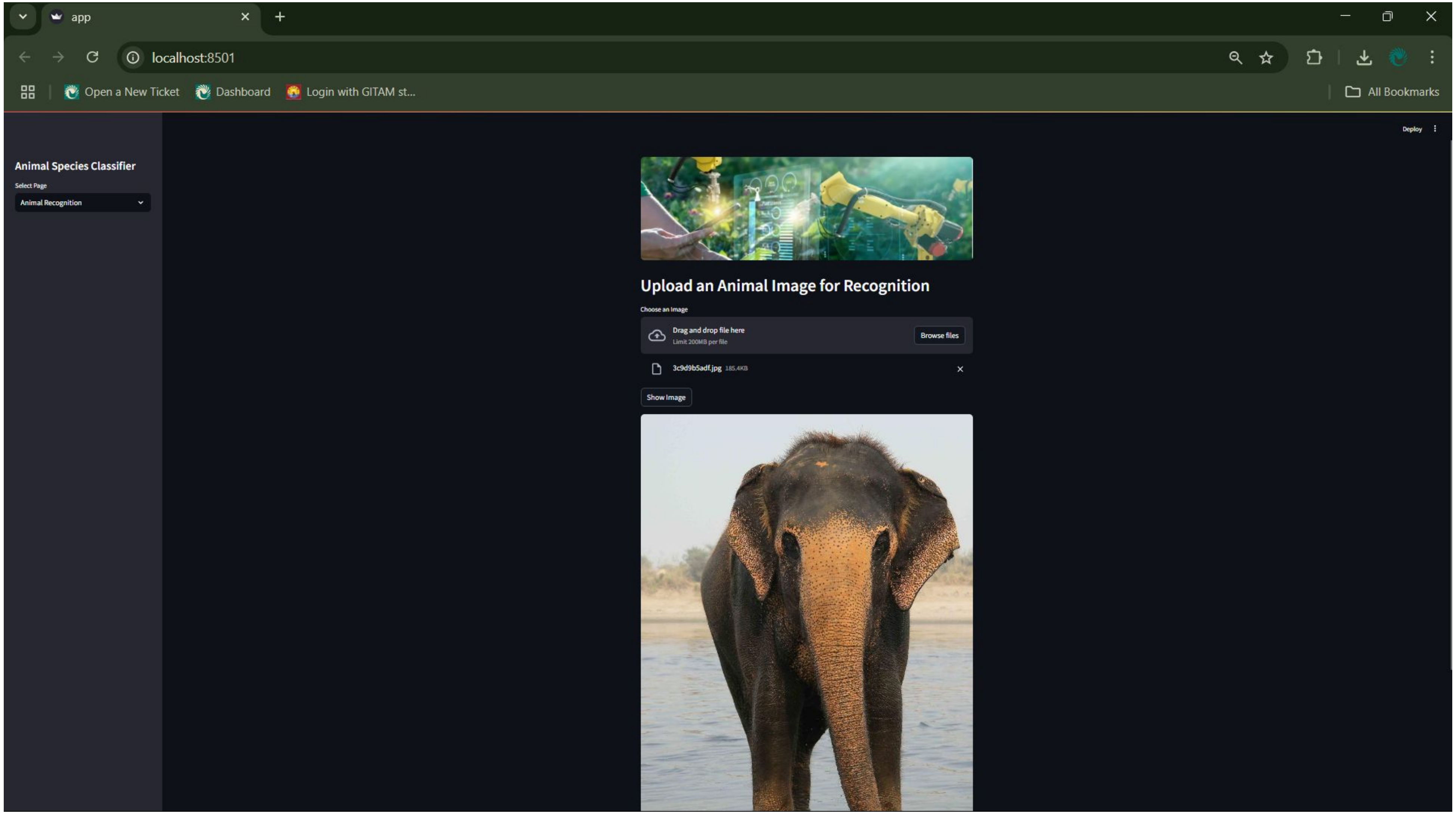
Screenshot of Output:



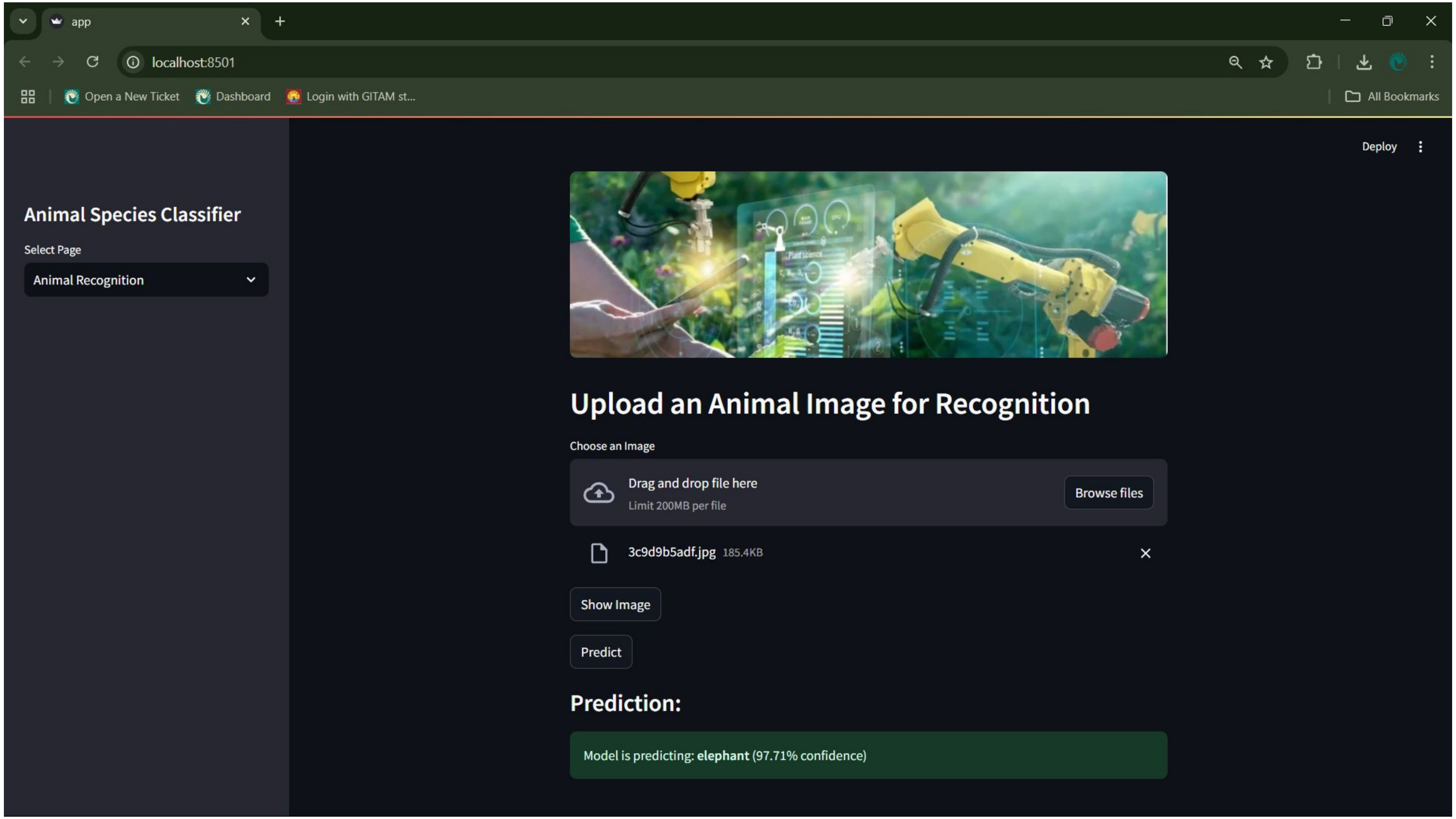
Screenshot of Output:

```
predict_animal('/kaggle/input/animal-image-dataset-90-different-animals/animals/animals/fox/27eaa1ccdd.jpg')
1/1 ━━━━━━━━━━━━━━━━━━━━ 4s 4s/step
```

Screenshot of Output:



Screenshot of Output:



Conclusion:

- Achieved an overall **accuracy of 84.8%** in classifying 90 different animal species.
- Utilized MobileNetV2 and transfer learning to reduce training time and improve performance.
- Applied effective preprocessing and augmentation to improve generalization on unseen data.
- Built and deployed a user-friendly frontend using Streamlit for real-time predictions.
- Verified consistent model performance across diverse animal classes and image types.
- **Future Scope**: Extend to video stream input, add endangered species detection, enable mobile deployment, and optimize model inference for faster response time.
- Github Repo Link: https://github.com/vnr-nitish/Animal-Species-Classifier-CNN-Streamlit.git