Automated Species Classification: A Transfer Learning Approach with EfficientNetB4

Subject: Naan Mudhalvan

Name: Vinothini R
Institute: Madras Institute of Technology
Department: Computer Science

OVERVIEW

1. PROBLEM STATEMENT
2. OBJECTIVES
3. PROPOSED SYSTEM / SOLUTION
4. SYSTEM APPROACH
5. ALGORITHM AND DEPLOYMENT
6. IMPLEMENTATION
7. RESULT
8. CONCLUSION
9. REFERENCES

PROBLEM STATEMENT

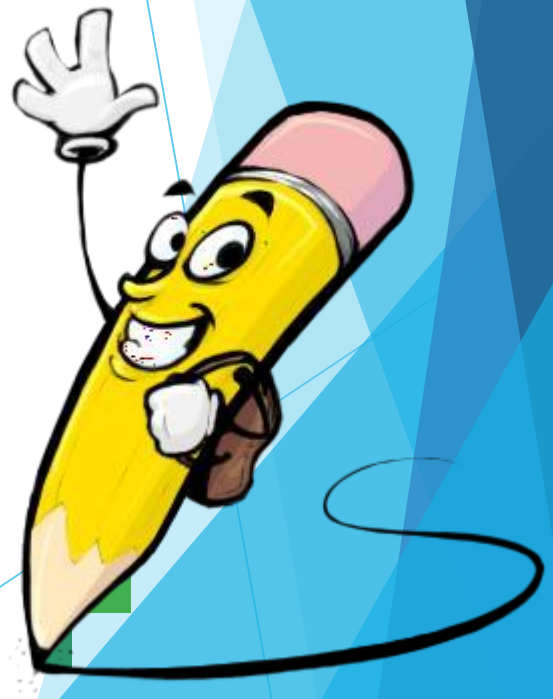
Maintaining biodiversity is vital for ecosystem stability, but species identification is challenging, with millions unidentified. Traditional methods are labor-intensive, prompting the adoption of machine learning and computer vision. Leveraging Convolutional Neural Networks (CNNs), particularly the EfficientNetB4 architecture, enables efficient species identification. The project implements a Python script for image classification using transfer learning with TensorFlow. It encompasses data preprocessing, model construction, training, fine-tuning, and evaluation, with visualization of training progress and metrics computation.

OBJECTIVES

Develop a Python script for species classification through image analysis using transfer learning with the EfficientNetB4 architecture.

Streamline the process of evaluating the model's performance and visualizing its training progress to facilitate model iteration and improvement.

Create a deployable system that can be integrated into various applications, enabling seamless use of the image classification model.



PROPOSED SYSTEM/ SOLUTION

The proposed solution utilizes transfer learning with TensorFlow and the EfficientNetB4 architecture to create an effective image classification system. It involves data preprocessing, model construction using EfficientNetB4 with additional layers for classification and dropout regularization, training on the dataset, fine-tuning, evaluation using metrics like accuracy, precision, recall, and F1-score, visualization of training progress, and deployment for real-world applications.

SYSTEM APPROACH

HARDWARE REQUIREMENTS:

1. Standard laptop / computer
2. Memory(RAM)
3. Internet Connection
4. Storage
5. GPU

SYSTEM APPROACH

SOFTWARE REQUIREMENTS:

1. Python
2. TensorFlow
3. Keras
4. Jupyter Notebook

ALGORITHM AND DEPLOYMENT

Data Preparation:

- Collect a dataset of images containing different species.
- Split the dataset into training, validation, and test sets.
- Apply data augmentation techniques to the training data to increase its diversity.

Model Construction:

- Choose a pre-trained base model (e.g., EfficientNetB4) suitable for transfer learning.
- Add custom layers on top of the base model to adapt it for species classification.
- Compile the model with appropriate loss function, optimizer, and metrics.

Model Training:

- Train the model on the training data:
- Initially, train with frozen base layers for a few epochs to allow the custom head layers to learn task-specific features.
- Fine-tune the model by unfreezing some layers of the base model and continuing training with a lower learning rate to fine-tune the learned features.
- Monitor training and validation metrics (loss and accuracy) to assess the model's performance and prevent overfitting.

ALGORITHM AND DEPLOYMENT

Evaluation and Validation:

- Evaluate the trained model on the validation and test datasets to assess its performance:
- Calculate metrics such as accuracy, precision, recall, and F1-score using the validation and test data.
- Visualize the model's performance using loss curves and classification results.

Deployment:

- Save the trained model in a deployable format (e.g., TensorFlow SavedModel or HDF5).
- Deploy the model to a production environment:
- Integrate the model with a deployment platform such as TensorFlow Serving, TensorFlow Lite, or TensorFlow.js.
- Expose an API endpoint for making predictions using the deployed model.
- Monitor the deployed model's performance and update it periodically with new data.

Iterative Improvement:

- Iterate on the model training and evaluation process to improve the model's performance:
- Experiment with different base models, architectures, and hyperparameters.
- Collect more data or fine-tune data augmentation techniques.
- Regularly evaluate the model's performance and make adjustments as needed.

IMPLEMENTATION

1. The project begins with data preparation, where a dataset containing images of various species is organized into training, validation, and testing sets.
2. Next, a pre-trained base model, such as EfficientNetB4, is customized with additional layers for species classification. The model is trained on the training data, initially with frozen base layers and then with fine-tuning.
3. Evaluation metrics like accuracy and F1-score are used to assess the model's performance on validation and test datasets. Finally, the trained model is deployed, and iterative improvements are made by experimenting with different architectures and techniques.

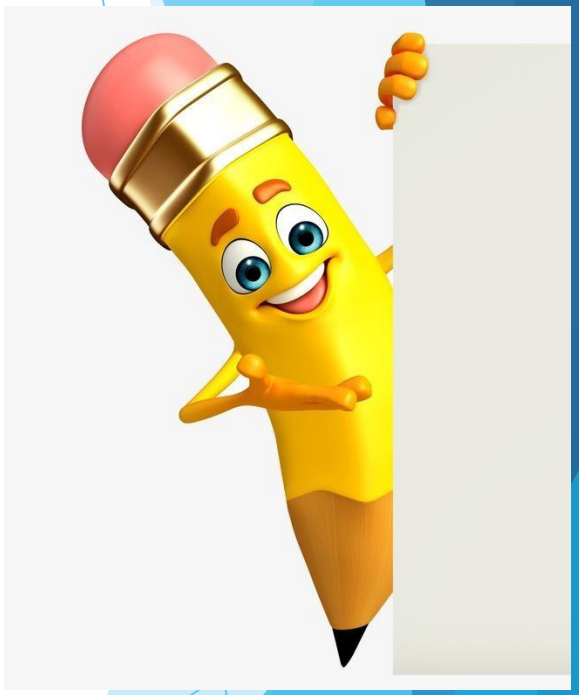
RESULT

Model Accuracy before fine tuning:

```
{'accuracy': 0.9462857142857143,
 'precision': 0.9587165532879818,
 'recall': 0.9462857142857143,
 'f1': 0.9454263856280662}
```

Model Accuracy after fine tuning:

```
{'accuracy': 0.9782857142857143,
 'precision': 0.9818594104308390,
 'recall': 0.9782857142857143,
 'f1': 0.9778991378991378}
```



https://github.com/Vinothini7904/NM---gen-ai-Automated-Species-Classification-.git

CONCLUSION

In conclusion, the species classification project involves a systematic approach to building, training, evaluating, and deploying a deep learning model for classifying images of various species. Beginning with data preparation and model construction, the project progresses through training and evaluation phases to assess the model's performance. The deployment phase ensures the model is integrated into a production environment for real-world use. Through continuous monitoring and iterative improvement, the model can be refined to achieve optimal performance and accuracy in species classification tasks. Overall, the project highlights the importance of thorough data preparation, model customization, and ongoing optimization to develop an effective and reliable species classification solution.

THANK YOU