A Comparative Analysis of Deep and Traditional Learning Models for Tomato Leaf Disease Classification

Project Overview

TomatoNet is a research project that compares the performance of traditional **Machine Learning (ML)** models and modern **Deep Learning (DL)** architectures in classifying tomato leaf diseases. With a focus on agricultural AI, this work explores how different models behave on real-world data, including the impact of background removal on performance.

Directory Structure

```
TOMATO-LEAF/
│
├── dataset/                        # Original dataset with disease folders
├── tomato\_bg\_removed/              # Background-removed version of dataset
├── models/                         # Saved DL models (.h5 files)
│   ├── best\_inceptionv3.h5
│   └── xception\_tomato\_leaf\_model.h5
├── metrics and visualizations/     # Evaluation visuals
│   ├── confusion\_matrix.png
│   ├── inception\_confusion\_matrix.png
│   ├── inception\_training\_metrics.png
│   └── training\_history.png
├── notebooks/                      # All model training notebooks
│   ├── bg\_removed\_inception.ipynb
│   ├── EfficientB1.ipynb
│   ├── inception.ipynb
│   ├── KNN+NB+XGBoost.ipynb
│   ├── Mobile\_shuffle\_efficient.ipynb
│   └── Xception.ipynb
└── readme.md                       # Project documentation
```

Dataset Details

- **Source:** Custom-collected dataset of tomato leaf images.
- **Classes:**
  - Tomato__Target_Spot
  - Tomato__Tomato_mosaic_virus
  - Tomato__Tomato_YellowLeaf__Curl_Virus
  - Tomato_Bacterial_spot
  - Tomato_Early_blight
  - Tomato_healthy
  - Tomato_Late_blight
  - Tomato_Leaf_Mold
  - Tomato_Septoria_leaf_spot
  - Tomato_Spider_mites_Two_spotted_spider_mite
- **Dataset Variants:**
  - **Original dataset**
  - **Background-removed dataset** (via preprocessing)

Models Compared

◇ Machine Learning Models

Implemented in `KNN+NB+XGBoost.ipynb`:

- K-Nearest Neighbors (KNN)
- Naive Bayes (NB)
- XGBoost

> Features were extracted using pre-trained CNNs or hand-crafted descriptors.

◇ Deep Learning Models

Model	Notebook	Accuracy (%)	Notes
Xception	`Xception.ipynb`	**97.00**	Best performing overall
EfficientNetB0	`EfficientB1.ipynb`	94.26	Excellent efficiency-accuracy
MobileNetV2	`Mobile_shuffle_efficient.ipynb`	92.79	Lightweight & fast
ShuffleNetV2	`Mobile_shuffle_efficient.ipynb`	90.54	Lightweight, less accurate
InceptionV2	`inception.ipynb`	83.00	Underperformed in this task

Evaluation Metrics

- **Accuracy**
- **Precision, Recall, F1-Score**
- **Confusion Matrix**
- **Training/Validation Curves**
- **Visuals** available in `metrics and visualizations/`

Key Findings

- **Xception** gave the highest accuracy (97%), making it ideal for deployment.
- **EfficientNetB0** offered a good trade-off between speed and performance.
- **MobileNetV2** and **ShuffleNetV2** are suitable for resource-constrained environments.
- **InceptionV2** was less effective despite its complexity.
- Traditional ML models are faster to train but were outperformed by DL models.
- **Removing background** boosted model performance by eliminating irrelevant noise.

Installation & Setup

1. Clone the repo:

```
git clone https://github.com/yourusername/tomato-leaf-disease-classification
cd tomato-leaf-disease-classification
```

```
2. Install dependencies:

   ```bash
   pip install -r requirements.txt
   ```

3. Run any Jupyter notebook in `notebooks/`.

```

```
## 🚀 How to Use

1. Use `KNN+NB+XGBoost.ipynb` for classical ML models.
2. Use the respective `.ipynb` files for DL model training.
3. Pre-trained models are available in the `models/` folder.
4. Evaluate performance using visuals from `metrics and visualizations/`.

---

## 🧠 Future Scope

* Include attention-based architectures (e.g., Vision Transformers)
* Fine-tune models with real-world data augmentation
* Deploy top-performing model into a **web/mobile plant health app**
* Test the model on other plant species and diseases

---

## 👨‍🔬 Author

**Obidur Rahman (Ashfin)**
🔗 [YouTube](https://www.youtube.com/@ObidurRahman) | 💬 Educator, ML Researcher & Content Creator
🧠 Passionate about AI, education, and solving real-world problems with tech.

---

## 📜 License

This project is open-source under the [MIT License](LICENSE).

---

*Made with 💚 and a lot of tomato leaves!*

```