OBJECTIVE

The objective of using deep learning for plant disease detection is to develop robust and automated systems that accurately identify and classify plant diseases from visual data, such as leaf images. By leveraging advanced deep learning models like CNNs, the goal is to enable early diagnosis, reduce reliance on manual inspection, and improve agricultural productivity. This approach aims to support farmers in implementing timely and effective disease management strategies, ultimately enhancing crop health and yield.

DATA COLLECTION

The dossier collection for the Ruins Fruit dataset from Kaggle involves curating first-rate concepts of Ash Fruit (also known as fuller fruit or winter melon) leaves that exhibit miscellaneous ailment symptoms in addition to athletic leaves for comparison. This dataset usually contains branded concepts that identify various types of plant ailments such as fungal contaminations, bacterial blight, or nutrient inadequacies, in addition to healthy leaf samples. The dossier is frequently sourced from palpable-world land atmospheres, with representations captured under diverse ignition environments and at different tumor stages to ensure instability and strength in disease discovery. The dataset grant permission also contain metadata such as the point of nurture, the stage of affliction progress, and other appropriate agronomic determinants. For this project, Kaggle determines a platform to approach and share the dataset accompanying a global society of machine learning experts, advancing collaboration.

DATA PREPROCESSING AND AUGMENTATION

Preprocessing and improving are essential steps in fitting the Ash Fruit dataset for preparation a machine intelligence model. Preprocessing includes cleaning and normative the countenance dossier to guarantee consistency. This involves tasks in the way that resizing countenances to a uniform breadth, converting bureaucracy to grayscale or RGB layout, make universal pixel principles (usually middle from two points 0 and 1), and management any absent or corrupt dossier. These steps guarantee that the figures are in a suitable plan for the Fantasy Device that drives a machine (ViT) model to process efficiently. Dossier augmentation improves the dataset by artificial means increasing allure size and instability, that helps improve model inference. Methods like random rotations, level and vertical flips, zooms, shine adaptations, and shifts are applied to the figures. This process creates alternatives of the existent images, mocking honest-world sketches where

illumination, angle, and adjustment may change. By incorporating these revolutions, improving reduces overfitting and enables the model to determine more robust looks, eventually improving allure accomplishment on unseen dossier.

SEGMENTATION

Separation in the framework of the Ash Fruit dataset refers to the process of recognizing and separating particular regions of interest, in the way that unhealthy regions on the leaves, from the rest of the representation. This pixel-reasonable categorization method admits for more precise study, as it divorces the impressed parts of the leaf from the healthy one, permissive the model to devote effort to something the manifestations of disease. Separation maybe acted utilizing algorithms like U-Net , which are standard for their strength to handle complex figures and accurately describe lines. By segregating the unhealthy sections, separation upgrades the model's capability to discover subtle face guide plant ailments, chief to more accurate categorization and conceivably contribution insights into the asperity or spread of the condition. This step is specifically beneficial in big agricultural requests, place early discovery of local disease outbreaks is important for barring extensive crop damage and give the results.

TRAINING

The model manifests productive learning on the preparation dossier, as designated apiece steady decrease in preparation misfortune and the logical increase in training veracity across epochs, arriving about 80% for one 10th epoch. This progress implies that the model has a stable capacity to discover patterns inside the preparation dataset, favorably capturing appropriate features and lowering error accompanying each period. These currents focal point the robustness of the preparation process and the model’s potential for correct guesses.

EVALUTION

The validation versification tells a more changeable accomplishment on unseen dossier. Confirmation veracity shows some vacillations about 35-40%, while confirmation loss originally drops piercingly but therefore oscillates middle from two points 2.0 and 4.5. These fluctuations signify that while the model is smart to capture key facets of the data, it struggles accompanying constant inference across various validation assortments.

Regardless of these vacillations, the validation versification still show promise, suggesting that the model is education statement patterns but concede possibility benefit from additional establishment in allure confirmation performance. To address this instability, various blueprints were used, including regularization methods like quitter, dossier augmentation, and early staying. These plans aim to develop the model's inference capability, guaranteeing it acts well not only on the preparation data but more on new, hidden samples. While the results display the model's powerful learning competency. The model's preparation and confirmation accuracy and misfortune curves portray promising knowledge patterns, accompanying the preparation accuracy firmly growing and the training misfortune deteriorating across epochs, signifying effective knowledge from the preparation data. The confirmation veracity also originally rises and achieves souped up earlier in occurrence than anticipated, demonstrating the model's ability to capture main patterns in the dossier. While the confirmation veracity dips marginally after the peak, this presents an event to improve generalization through methods like regularization or early staying, conceivably leading to even healthier efficiency on unseen dossier. Overall, these results indicate the model's forceful learning volume, accompanying room for finetuning to further to improve inference.

CONCLUSION

Finally, the model shows powerful learning wherewithal, as proved by the agreeing bettering in training veracity and decline in training deficit over epochs. Still, the vacillations in validation veracity and misfortune suggest few challenges in inference, which were lightened to a range through regularization techniques, dossier improving, and early staying. These adjustments aided improve stability in the model's confirmation accomplishment, but further tuning grant permission still command a price of to fully help along between preparation and validation results. Overall, the model manifests hopeful potential, with a hard groundwork in learning key dossier patterns. Future work take care of investigate additional methods to further embellish generalization, eventually chief to a more robust model for useful arrangement.

SAMPLE CODE

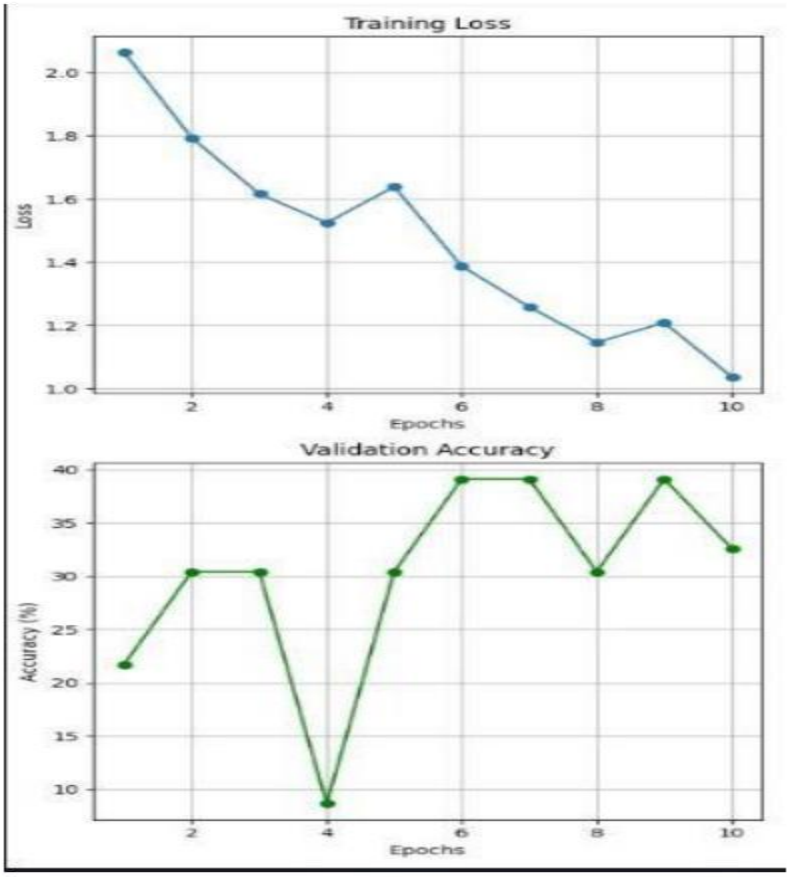
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt


train_dir = 'path/to/train/dataset'
val_dir = 'path/to/validation/dataset'


train_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

val_datagen = ImageDataGenerator(rescale=1.0/255)

train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=(128, 128),
    batch_size=32,
    class_mode='categorical'
)

val_data = val_datagen.flow_from_directory(
    val_dir,
    target_size=(128, 128),
    batch_size=32,
```

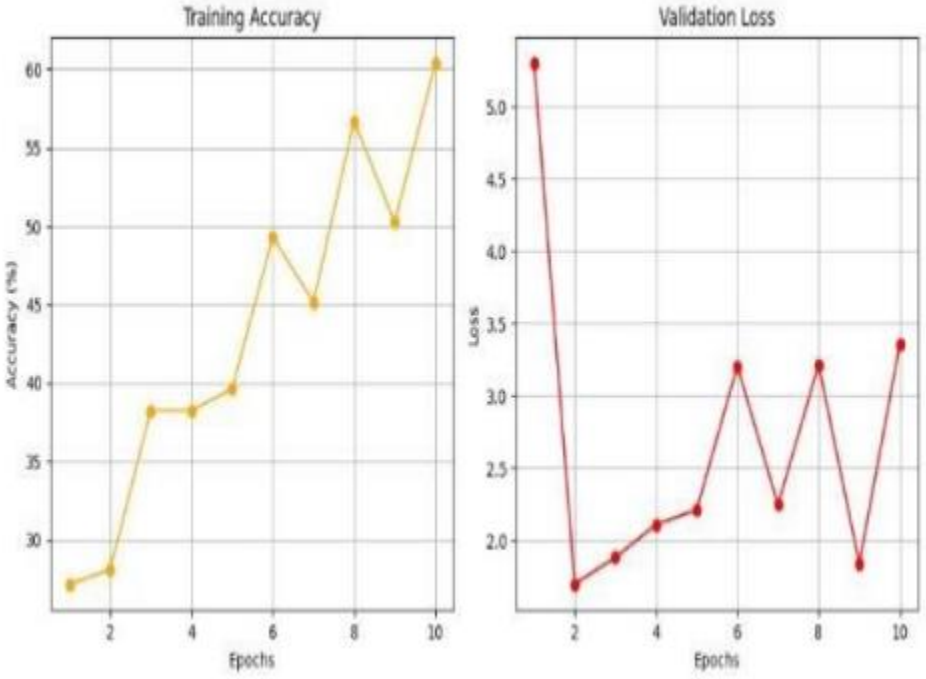
```
    class_mode='categorical'
)


model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(train_data.num_classes, activation='softmax')
])


model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])


history = model.fit(
    train_data,
    epochs=20,
    validation_data=val_data
)


plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()


model.save('plant_disease_model.h5')
```

```
def predict_image(image_path):
    from tensorflow.keras.preprocessing import image
    import numpy as np

    img = image.load_img(image_path, target_size=(128, 128))
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    predicted_class = train_data.class_indices
    label_map = {v: k for k, v in predicted_class.items()}
    predicted_label = label_map[np.argmax(prediction)]
    print(f"Predicted Disease: {predicted_label}")
```

OUTPUT

```
  self._warn_if_super_not_called()
3/3 ━━━━━━━━━━━━━━━━━━━━ 86s 9s/step - accuracy: 0.1840 - loss: 8.7014 - val_accuracy: 0.3458 - val_loss: 5.9555
Epoch 2/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 63s 1s/step - accuracy: 0.3897 - loss: 4.6580 - val_accuracy: 0.9877 - val_loss: 0.2316
Epoch 3/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.4586 - loss: 2.4795 - val_accuracy: 1.0000 - val_loss: 0.0581
Epoch 4/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 1s/step - accuracy: 0.5226 - loss: 1.9635 - val_accuracy: 1.0000 - val_loss: 0.0351
Epoch 5/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.6214 - loss: 1.3909 - val_accuracy: 1.0000 - val_loss: 0.1310
Epoch 6/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 5s 1s/step - accuracy: 0.6779 - loss: 1.2228 - val_accuracy: 1.0000 - val_loss: 0.4980
Epoch 7/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.7068 - loss: 1.0886 - val_accuracy: 1.0000 - val_loss: 0.9312
Epoch 8/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 1s/step - accuracy: 0.7263 - loss: 1.0103 - val_accuracy: 0.9942 - val_loss: 1.1794
Epoch 9/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.7387 - loss: 0.9285 - val_accuracy: 0.9532 - val_loss: 1.2876
Epoch 10/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 5s 1s/step - accuracy: 0.7341 - loss: 0.9104 - val_accuracy: 0.8477 - val_loss: 1.3538
Epoch 11/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 5s 1s/step - accuracy: 0.7420 - loss: 0.8348 - val_accuracy: 0.6760 - val_loss: 1.4320
Epoch 12/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.7482 - loss: 0.7842 - val_accuracy: 0.5777 - val_loss: 1.5549
Epoch 13/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.7742 - loss: 0.6882 - val_accuracy: 0.4970 - val_loss: 1.7496
Epoch 14/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 1s/step - accuracy: 0.7380 - loss: 0.7501 - val_accuracy: 0.4420 - val_loss: 1.8855
Epoch 15/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 5s 1s/step - accuracy: 0.7739 - loss: 0.6253 - val_accuracy: 0.3824 - val_loss: 1.9451
Epoch 16/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 5s 1s/step - accuracy: 0.7946 - loss: 0.5523 - val_accuracy: 0.3249 - val_loss: 1.9459
Epoch 17/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 5s 1s/step - accuracy: 0.8088 - loss: 0.5036 - val_accuracy: 0.2638 - val_loss: 1.9459
Epoch 18/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.7762 - loss: 0.5530 - val_accuracy: 0.1952 - val_loss: 1.9459
Epoch 19/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 1s/step - accuracy: 0.7986 - loss: 0.4952 - val_accuracy: 0.1176 - val_loss: 1.9459
Epoch 20/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 1s/step - accuracy: 0.8113 - loss: 0.4532 - val_accuracy: 0.0563 - val_loss: 1.9459
```


Training Loss
Loss
Epochs
Validation Accuracy
Accuracy (%)
Epochs


Training Accuracy
Accuracy (%)
Epochs
Validation Loss
Loss
Epochs

RESULT

Thus the Plant disease detection program using CNN was executed successfully.