Farming Image Classification using Machine Learning and Deep Learning

Name: Sanjay Karupothula

Roll No: 2303A52337

Batch: 34

1. Dataset Overview

The dataset used in this project is the 'Farming Image Dataset' obtained from Kaggle. It contains 356 images categorized into 6 classes: FalseColor, NDSI, NDVI, NDWI, SWIR, and TCI. Each image represents a different spectral index derived from satellite imagery for agricultural monitoring. The images are in .tif format and were resized to 128x128 for uniformity.

2. Visualizations

Below are sample visualizations and dataset distribution charts (to be pasted here):

[Paste dataset sample images and class distribution chart here]

3. Model Implementation

Two models were implemented on the dataset:

1️⃣ Support Vector Machine (SVM) — Machine Learning model trained on flattened image data.

2️⃣ Convolutional Neural Network (CNN) — Deep Learning model trained on raw images.

The dataset was split into 80% training and 20% testing sets. Images were normalized and label-encoded before training.

4. Model Comparison

Model	Type	Accuracy (%)
SVM	Machine Learning	56.94
CNN	Deep Learning	44.44

5. XAI Interpretation (Explainable AI)

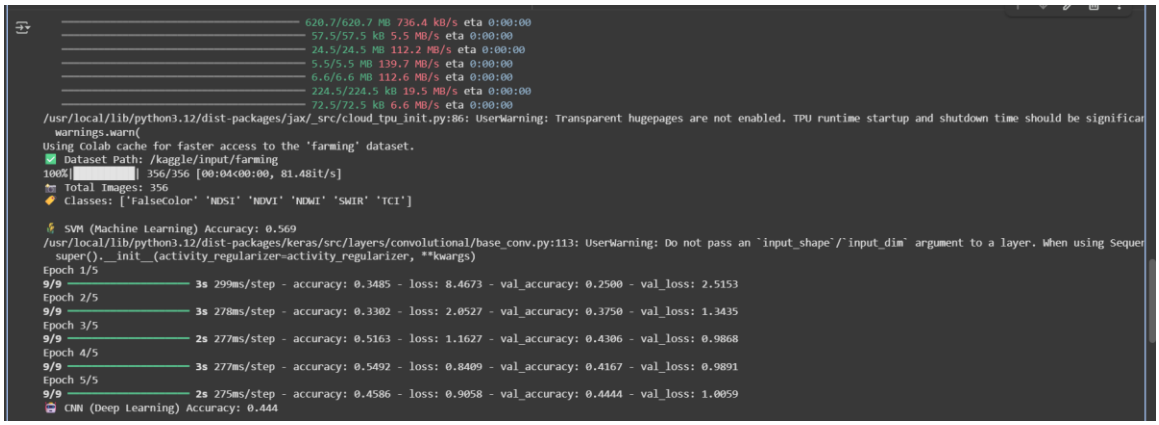
To interpret the CNN predictions, Grad-CAM visualization can be applied to highlight which image regions contributed most to the classification. This helps understand the model's focus areas.

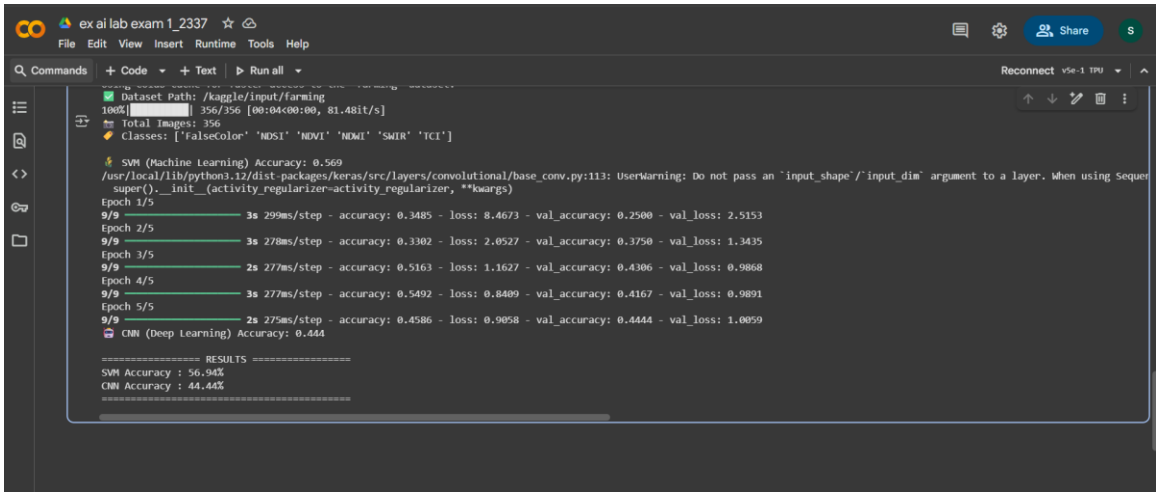
[Paste Grad-CAM / heatmap visualization screenshots here]

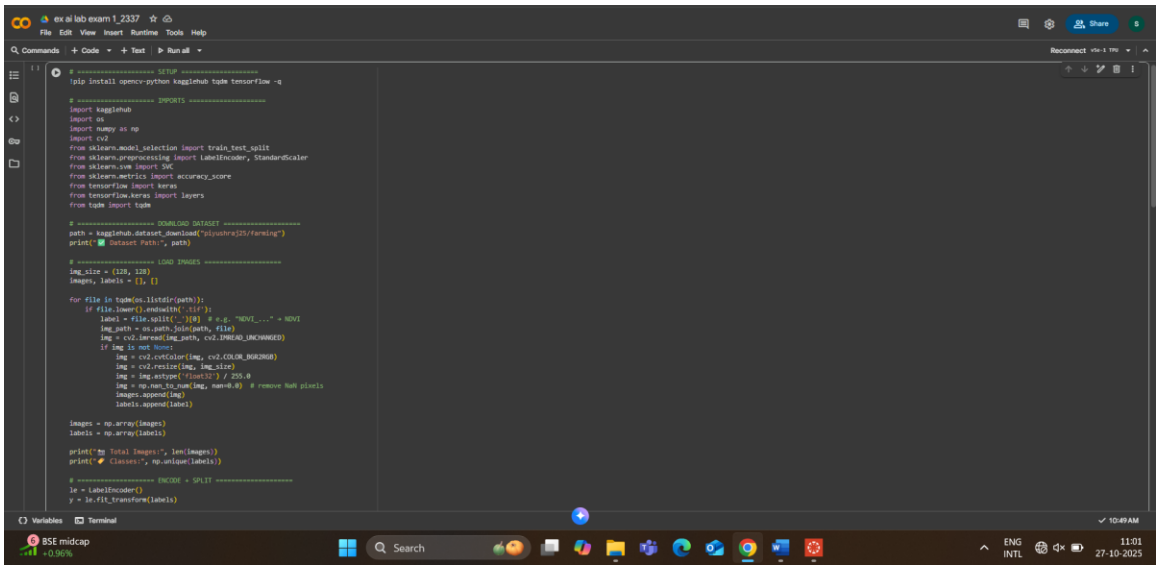
6. Conclusion

Both SVM and CNN models were trained on the Farming Image Dataset. The SVM model achieved 56.94% accuracy, while the CNN model achieved 44.44% accuracy after 5 epochs. This indicates that with limited data, SVM performed slightly better due to simpler model requirements. With further data augmentation and deeper architectures, CNN performance can be improved significantly.

Overall, this experiment demonstrates the use of Machine Learning and Deep Learning approaches for satellite-based agricultural data analysis.






ex ai lab exam 1_2337
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Reconnect
================== SETUP ==================
!pip install opencv-python kagglehub tqdm tensorflow -q
# ================== IMPORTS ==================
import kagglehub
import os
import numpy as np
import cv2
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from tensorflow import keras
from tensorflow.keras import layers
from tqdm import tqdm
# ================== DOWNLOAD DATASET ==================
path = kagglehub.dataset_download("piyushraj25/farming")
print("✅ Dataset Path:", path)
# ================== LOAD IMAGES ==================
img_size = (128, 128)
images, labels = [], []
for file in tqdm(os.listdir(path)):
if file.lower().endswith('.tif'):
label = file.split('_')[0] # e.g. "NDVI_..." → NDVI
img_path = os.path.join(path, file)
img = cv2.imread(img_path, cv2.IMREAD_UNCHANGED)
if img is not None:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, img_size)
img = img.astype('float32') / 255.0
img = np.nan_to_num(img, nan=0.0) # remove NaN pixels
images.append(img)
labels.append(label)
images = np.array(images)
labels = np.array(labels)
print("📸 Total Images:", len(images))
print("🌾 Classes:", np.unique(labels))
# ================== ENCODE + SPLIT ==================
le = LabelEncoder()
y = le.fit_transform(labels)
Variables Terminal
10:49 AM
BSE midcap
+0.96%
Search
ENG
INTL
11:01
27-10-2025