

day1-022

June 26, 2024

```
[ ]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
[ ]: import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
IMG_SIZE = 224
BATCH_SIZE = 32
```

```
[ ]: train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Agriculture_CNN/train',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)
val_generator = train_datagen.flow_from_directory(
    '//content/drive/MyDrive/Agriculture_CNN/train',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)
```

Found 125 images belonging to 4 classes.

Found 30 images belonging to 4 classes.

```
[ ]: model = keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_SIZE,
↪ IMG_SIZE, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
```

```

layers.MaxPooling2D((2, 2)),
layers.Flatten(),
layers.Dense(128, activation='relu'),
layers.Dense(1, activation='sigmoid')
])

```

```
[ ]: model.compile(optimizer='adam', loss='binary_crossentropy',
↳ metrics=['accuracy'])
```

```
[ ]: model.fit(train_generator, epochs=5, validation_data=val_generator)
```

```

Epoch 1/5
45/45 [=====] - 419s 9s/step - loss: -829751.2500 -
accuracy: 0.1532 - val_loss: -6144372.5000 - val_accuracy: 0.1525
Epoch 2/5
45/45 [=====] - 171s 4s/step - loss: -121109224.0000 -
accuracy: 0.1525 - val_loss: -507918400.0000 - val_accuracy: 0.1525
Epoch 3/5
45/45 [=====] - 179s 4s/step - loss: -2562991872.0000 -
accuracy: 0.1525 - val_loss: -7133280256.0000 - val_accuracy: 0.1525
Epoch 4/5
45/45 [=====] - 181s 4s/step - loss: -21155342336.0000
- accuracy: 0.1525 - val_loss: -46020231168.0000 - val_accuracy: 0.1525
Epoch 5/5
45/45 [=====] - 189s 4s/step - loss: -99907518464.0000
- accuracy: 0.1525 - val_loss: -183111860224.0000 - val_accuracy: 0.1525

```

```
[ ]: <keras.src.callbacks.History at 0x7b726c1020b0>
```

```
[ ]: model.save("model.h5", "label.txt")
```

```

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(

```

```

[ ]: from tensorflow.keras.models import load_model
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
# Load the saved model
model = load_model('/content/model.h5')
# Load and preprocessor the test image
test_image_path = '/content/drive/MyDrive/Agriculture_CNN/train/maize/maize001a.
↳ jpeg'
img = image.load_img(test_image_path, target_size=(224, 224))

```

```
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
# add batch demension
img_array = img_array / 255.0
# make prediction
predictions = model.predict(img_array)
#print the prediction
print(predictions)
```

```
1/1 [=====] - 0s 291ms/step
[[0.50504625]]
```

```
[ ]: if predictions < 0.25:
    print('It is a rice')
elif predictions < 0.5:
    print('It is a sugarcane')
elif predictions < 0.75:
    print("it is a maize")
else:
    print("it is a wheat")
```

```
it is a maize
```