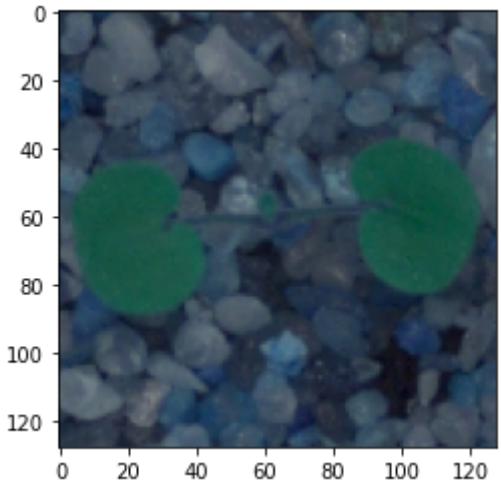
In [*]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn import metrics
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
```

In [ ]:
```
data = np.load("images.npy")
data.shape
```

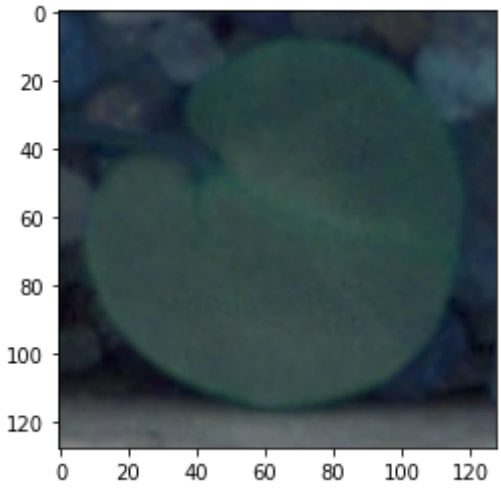
In [ ]:
```
labels = pd.read_csv("Labels.csv")
labels.head()
```

In [13]:
```
from matplotlib import pyplot as plt

plt.imshow(data[0])
plt.show()
```



In [12]:
```
plt.imshow(data[300])
plt.show()
```



Normalize the data

In [17]:
```
data = data.astype('float32')

data /= 255


print("data shape:", data.shape)
print("Images in data:", data.shape[0])
print("Max value in data:", data.max())
```

```
data shape: (4750, 128, 128, 3)
Images in data: 4750
Max value in data: 1.0
```

Gaussian Blurring

In [22]:
```
pip install google-colab
```

```
    Found existing installation: urllib3 1.25.10
    Uninstalling urllib3-1.25.10:
      Successfully uninstalled urllib3-1.25.10
  Attempting uninstall: idna
    Found existing installation: idna 2.10
    Uninstalling idna-2.10:
      Successfully uninstalled idna-2.10
  Attempting uninstall: requests
    Found existing installation: requests 2.24.0
    Uninstalling requests-2.24.0:
      Successfully uninstalled requests-2.24.0
Successfully installed google-auth-1.4.2 google-colab-1.0.0 idna-2.8 ipyk
ernel-4.6.1 ipython-5.5.0 notebook-5.2.2 pandas-0.24.2 portpicker-1.2.0 p
rompt-toolkit-1.0.18 requests-2.21.0 simplegeneric-0.8.1 six-1.12.0 torna
do-4.5.3 urllib3-1.24.3
WARNING: You are using pip version 20.0.2; however, version 20.2.4 is ava
ilable.
You should consider upgrading via the '/usr/local/Cellar/jupyterlab/2.0.
1/libexec/bin/python3.7 -m pip install --upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:
```
from google.colab import drive
from scipy.signal import convolve2d
import cv2
from google.colab.patches import cv2_imshow


Gaussian1 = data.GaussianBlur(image, (5, 5), 0)
Gaussian2 = data.GaussianBlur(image, (15, 15), 0)
print('Original Image:\n')
cv2_imshow(image)
print('\n Output after first gaussian blurring: \n')
cv2_imshow(Gaussian1)
print('\n Output after second gaussian blurring: \n')
cv2_imshow(Gaussian2)
```

Split the data

In [14]:
```
from sklearn.model_selection import train_test_split

X = data
Y = labels
X_train, X_test, ytrain, ytest = train_test_split( X, Y, test_size = 0.1

```

One hot encode the labels

In [ ]:
```
from tensorflow.keras.utils import to_categorical

y_train = to_categorical(y_train, num_classes=12)
y_test = to_categorical(y_test, num_classes=12)

print("Shape of y_train:", y_train.shape)
print("One value of y_train:", y_train[0])
```

This does not need to be reshaped in order to work for Keras since its already in the format (n_e, n_h, n_w, n_c)

n_e= number of examples, n_h = height, n_w = width, n_c = number of channels

Build model

In [ ]:
```
from tensorflow.keras.layers import Conv2D

model = Sequential()
model.add(Conv2D(filters=32, kernel_size=3, activation="relu", input_sha
model.add(Conv2D(filters=32, kernel_size=3, activation="relu"))
model.add(Flatten())
model.add(Dense(128, activation="relu"))
model.add(Dense(12, activation="softmax"))
```

Split for validation dataset below in the fit

In [ ]:
```
# Compile the model
model.compile(loss="categorical_crossentropy", metrics=["accuracy"], opt

# Fit the model
model.fit( x=X_train, y=ytrain, batch_size=32, epochs=10, validation_spl
```

In [ ]:
```
model.evaluate(X_test, ytest)
```

Visualize the data

x_test[2], x_test[3], x_test[33], x_test[36], x_test[59]

In [ ]:
```
plt.imshow(X_test[2])
plt.show()
```

In [ ]:
```
plt.imshow(X_test[3])
plt.show()
```

In [ ]:
```
plt.imshow(X_test[33])
plt.show()
```

In [ ]:
```
plt.imshow(X_test[36])
plt.show()
```

In [ ]:
```
plt.imshow(X_test[59])
plt.show()
```