In [13]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers
```

In [2]:

```
batch_size = 100
img_height = 250
img_width = 250
```

In [3]:

```
training_ds = tf.keras.preprocessing.image_dataset_from_directory(
    'New Plant Diseases Dataset(Augmented)/train',
    seed=42,
    image_size= (img_height, img_width),
    batch_size=batch_size)
```

```
Found 70295 files belonging to 38 classes.
```

In [4]:

```
validation_ds =  tf.keras.preprocessing.image_dataset_from_directory(
    'New Plant Diseases Dataset(Augmented)/valid',
    seed=42,
    image_size= (img_height, img_width),
    batch_size=batch_size)
```

```
Found 17572 files belonging to 38 classes.
```

In [5]:

```
class_names = training_ds.class_names
```

In [6]:

```
MyCnn = tf.keras.models.Sequential([
  layers.BatchNormalization(),
  layers.Conv2D(32, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(128, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(256, activation='relu'),
  layers.Dense(len(class_names), activation= 'softmax')
])
```

In [7]:

```
MyCnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy', metrics=['accur
```

In [8]:

```
retVal = MyCnn.fit(training_ds,validation_data= validation_ds,epochs = 2)
```

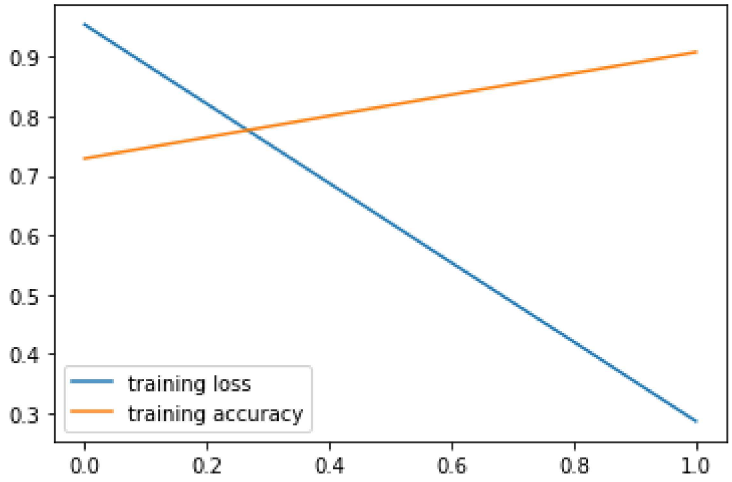
```
Epoch 1/2
703/703 [==============================] - 2620s 4s/step - loss: 0.9547 - accuracy:
0.7291 - val_loss: 0.4364 - val_accuracy: 0.8646
Epoch 2/2
703/703 [==============================] - 3109s 4s/step - loss: 0.2869 - accuracy:
0.9083 - val_loss: 0.2936 - val_accuracy: 0.9046
```

In [9]:

```
plt.plot(retVal.history['loss'], label = 'training loss')
plt.plot(retVal.history['accuracy'], label = 'training accuracy')
plt.legend()
```

Out[9]:

```
<matplotlib.legend.Legend at 0x1b73e5dbaf0>
```



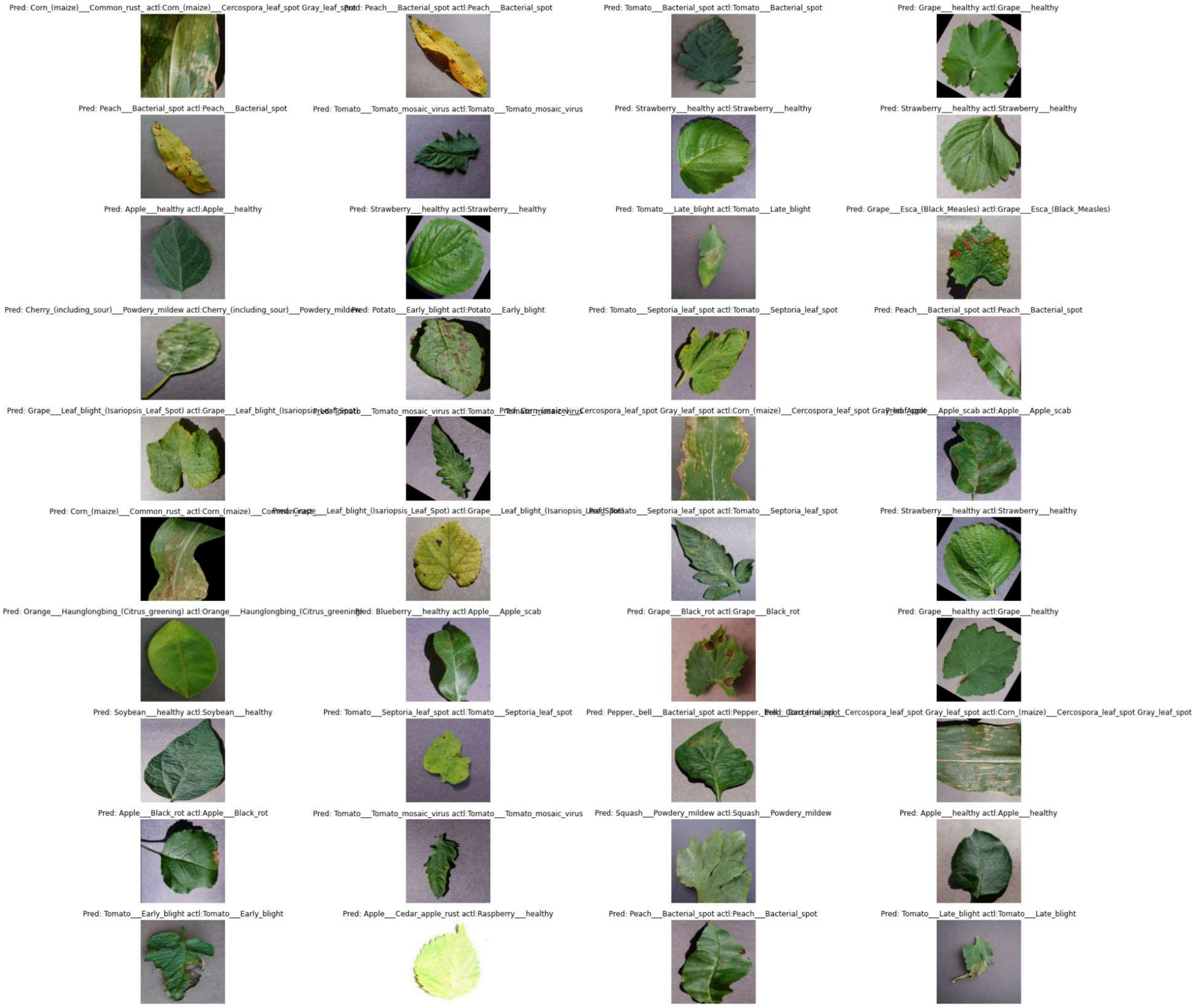
In [10]:

```
AccuracyVector = []
plt.figure(figsize=(30, 30))
for images, labels in validation_ds.take(1):
    predictions = MyCnn.predict(images)
    predlabel = []
    prdlbl = []

    for mem in predictions:
        predlabel.append(class_names[np.argmax(mem)])
        prdlbl.append(np.argmax(mem))

    AccuracyVector = np.array(prdlbl) == labels
    for i in range(40):
        ax = plt.subplot(10, 4, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title('Pred: '+ predlabel[i]+' actl:'+class_names[labels[i]] )
        plt.axis('off')
        plt.grid(True)
```

```
4/4 [==============================] - 1s 186ms/step
```

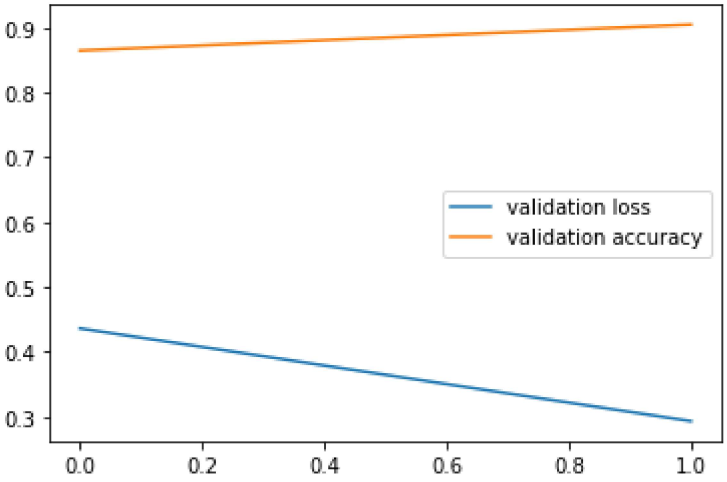

Pred: Corn_(maize)___Common_rust_ actl:Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
Pred: Peach___Bacterial_spot actl:Peach___Bacterial_spot
Pred: Tomato___Bacterial_spot actl:Tomato___Bacterial_spot
Pred: Grape___healthy actl:Grape___healthy
Pred: Peach___Bacterial_spot actl:Peach___Bacterial_spot
Pred: Tomato___Tomato_mosaic_virus actl:Tomato___Tomato_mosaic_virus
Pred: Strawberry___healthy actl:Strawberry___healthy
Pred: Strawberry___healthy actl:Strawberry___healthy
Pred: Apple___healthy actl:Apple___healthy
Pred: Strawberry___healthy actl:Strawberry___healthy
Pred: Tomato___Late_blight actl:Tomato___Late_blight
Pred: Grape___Esca_(Black_Measles) actl:Grape___Esca_(Black_Measles)
Pred: Cherry_(including_sour)___Powdery_mildew actl:Cherry_(including_sour)___Powdery_mildew
Pred: Potato___Early_blight actl:Potato___Early_blight
Pred: Tomato___Septoria_leaf_spot actl:Tomato___Septoria_leaf_spot
Pred: Peach___Bacterial_spot actl:Peach___Bacterial_spot
Pred: Grape___Leaf_blight_(Isariopsis_Leaf_Spot) actl:Grape___Leaf_blight_(Isariopsis_Leaf_Spot)
Pred: Tomato___Tomato_mosaic_virus actl:Tomato___Tomato_mosaic_virus
Pred: Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot actl:Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
Pred: Apple___Apple_scab actl:Apple___Apple_scab
Pred: Corn_(maize)___Common_rust_ actl:Corn_(maize)___Common_rust_
Pred: Grape___Leaf_blight_(Isariopsis_Leaf_Spot) actl:Grape___Leaf_blight_(Isariopsis_Leaf_Spot)
Pred: Tomato___Septoria_leaf_spot actl:Tomato___Septoria_leaf_spot
Pred: Strawberry___healthy actl:Strawberry___healthy
Pred: Orange___Haunglongbing_(Citrus_greening) actl:Orange___Haunglongbing_(Citrus_greening)
Pred: Blueberry___healthy actl:Apple___Apple_scab
Pred: Grape___Black_rot actl:Grape___Black_rot
Pred: Grape___healthy actl:Grape___healthy
Pred: Soybean___healthy actl:Soybean___healthy
Pred: Tomato___Septoria_leaf_spot actl:Tomato___Septoria_leaf_spot
Pred: Pepper,_bell___Bacterial_spot actl:Pepper,_bell___Bacterial_spot
Pred: Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot actl:Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
Pred: Apple___Black_rot actl:Apple___Black_rot
Pred: Tomato___Tomato_mosaic_virus actl:Tomato___Tomato_mosaic_virus
Pred: Squash___Powdery_mildew actl:Squash___Powdery_mildew
Pred: Apple___healthy actl:Apple___healthy
Pred: Tomato___Early_blight actl:Tomato___Early_blight
Pred: Apple___Cedar_apple_rust actl:Raspberry___healthy
Pred: Peach___Bacterial_spot actl:Peach___Bacterial_spot
Pred: Tomato___Late_blight actl:Tomato___Late_blight

In [11]:

```
plt.plot(retVal.history['val_loss'], label = 'validation loss')
plt.plot(retVal.history['val_accuracy'], label = 'validation accuracy')
plt.legend()
```

Out[11]:

```
<matplotlib.legend.Legend at 0x1b788e66cd0>
```



In [ ]: