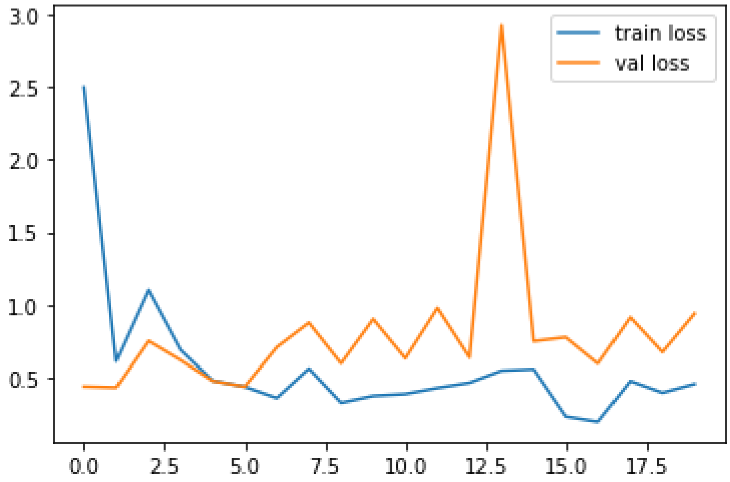
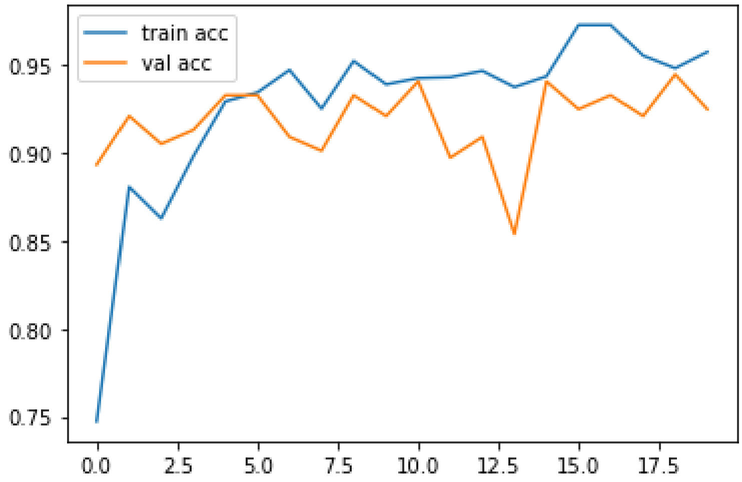
In [17]:

```
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.5
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)

from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.inception_v3 import preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt

# Define the image size and paths to the train, validation, and test directories
IMAGE_SIZE = [224, 224]
train_path = r'C:\Users\Abhishek\Downloads\archive (23)\Cotton Disease\train'
valid_path =r'C:\Users\Abhishek\Downloads\archive (23)\Cotton Disease\val'
test_path = r'C:\Users\Abhishek\Downloads\archive (23)\Cotton Disease\test'

# Use the InceptionV3 pre-trained model with the weights from ImageNet
inception = InceptionV3(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_to

# Freeze all the layers in the pre-trained model
for layer in inception.layers:
    layer.trainable = False

# Get the number of output classes
folders = glob(train_path + '/*')

# Flatten the output from the pre-trained model
x = Flatten()(inception.output)

# Add a fully connected layer with a softmax activation function for multiclass clas
prediction = Dense(len(folders), activation='softmax')(x)

# Create a model object using the input and output layers
model = Model(inputs=inception.input, outputs=prediction)

# Compile the model with categorical cross-entropy loss, Adam optimizer, and accurac
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)

# Use ImageDataGenerator to import images and perform data augmentation
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2,
valid_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_set = train_datagen.flow_from_directory(train_path, target_size=(224, 224), ba
valid_set = valid_datagen.flow_from_directory(valid_path, target_size=(224, 224), ba
test_set = test_datagen.flow_from_directory(test_path, target_size=(224, 224), batch

# Train the model using the train and validation datasets
r = model.fit_generator(
  train_set,
  validation_data=valid_set,
  epochs=20,
  steps_per_epoch=len(train_set),
  validation_steps=len(valid_set)
)

# Plot the training and validation loss and accuracy
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()

plt.plot(r.history['accuracy'], label='train acc')
plt.plot(r.history['val_accuracy'], label='val acc')
plt.legend()
plt.show()

# Evaluate the model using the test dataset
test_loss, test_acc = model.evaluate_generator(test_set)
print("Test Loss:", test_loss)
print("Test Accuracy:", test_acc)
```

```
C:\Users\Abhishek\anaconda3\lib\site-packages\tensorflow\python\client\session.py:17
69: UserWarning: An interactive session is already active. This can cause out-of-mem
ory errors in some cases. You must explicitly call `InteractiveSession.close()` to r
elease resources held by the other session(s).
  warnings.warn('An interactive session is already active. This can '
Found 1951 images belonging to 4 classes.
Found 253 images belonging to 4 classes.
Found 106 images belonging to 4 classes.
<ipython-input-17-f305f83a4b85>:59: UserWarning: `Model.fit_generator` is deprecated
and will be removed in a future version. Please use `Model.fit`, which supports gene
rators.
  r = model.fit_generator(
Epoch 1/20
61/61 [==============================] - 158s 2s/step - loss: 2.4977 - accuracy: 0.7
473 - val_loss: 0.4390 - val_accuracy: 0.8933
Epoch 2/20
61/61 [==============================] - 146s 2s/step - loss: 0.6146 - accuracy: 0.8
806 - val_loss: 0.4325 - val_accuracy: 0.9209
Epoch 3/20
61/61 [==============================] - 143s 2s/step - loss: 1.1019 - accuracy: 0.8
626 - val_loss: 0.7524 - val_accuracy: 0.9051
Epoch 4/20
61/61 [==============================] - 145s 2s/step - loss: 0.6908 - accuracy: 0.8
980 - val_loss: 0.6209 - val_accuracy: 0.9130
Epoch 5/20
61/61 [==============================] - 145s 2s/step - loss: 0.4794 - accuracy: 0.9
293 - val_loss: 0.4770 - val_accuracy: 0.9328
Epoch 6/20
61/61 [==============================] - 147s 2s/step - loss: 0.4378 - accuracy: 0.9
344 - val_loss: 0.4379 - val_accuracy: 0.9328
Epoch 7/20
61/61 [==============================] - 146s 2s/step - loss: 0.3602 - accuracy: 0.9
472 - val_loss: 0.7094 - val_accuracy: 0.9091
Epoch 8/20
61/61 [==============================] - 148s 2s/step - loss: 0.5620 - accuracy: 0.9
252 - val_loss: 0.8771 - val_accuracy: 0.9012
Epoch 9/20
61/61 [==============================] - 140s 2s/step - loss: 0.3277 - accuracy: 0.9
523 - val_loss: 0.5987 - val_accuracy: 0.9328
Epoch 10/20
61/61 [==============================] - 143s 2s/step - loss: 0.3752 - accuracy: 0.9
390 - val_loss: 0.9019 - val_accuracy: 0.9209
Epoch 11/20
61/61 [==============================] - 149s 2s/step - loss: 0.3884 - accuracy: 0.9
426 - val_loss: 0.6334 - val_accuracy: 0.9407
Epoch 12/20
61/61 [==============================] - 145s 2s/step - loss: 0.4296 - accuracy: 0.9
```

```
431 - val_loss: 0.9783 - val_accuracy: 0.8972
Epoch 13/20
61/61 [==============================] - 315s 5s/step - loss: 0.4651 - accuracy: 0.9
467 - val_loss: 0.6391 - val_accuracy: 0.9091
Epoch 14/20
61/61 [==============================] - 156s 3s/step - loss: 0.5477 - accuracy: 0.9
375 - val_loss: 2.9277 - val_accuracy: 0.8538
Epoch 15/20
61/61 [==============================] - 156s 3s/step - loss: 0.5578 - accuracy: 0.9
436 - val_loss: 0.7503 - val_accuracy: 0.9407
Epoch 16/20
61/61 [==============================] - 158s 3s/step - loss: 0.2338 - accuracy: 0.9
728 - val_loss: 0.7775 - val_accuracy: 0.9249
Epoch 17/20
61/61 [==============================] - 160s 3s/step - loss: 0.1974 - accuracy: 0.9
728 - val_loss: 0.5975 - val_accuracy: 0.9328
Epoch 18/20
61/61 [==============================] - 175s 3s/step - loss: 0.4763 - accuracy: 0.9
554 - val_loss: 0.9136 - val_accuracy: 0.9209
Epoch 19/20
61/61 [==============================] - 178s 3s/step - loss: 0.3968 - accuracy: 0.9
482 - val_loss: 0.6761 - val_accuracy: 0.9447
Epoch 20/20
61/61 [==============================] - 175s 3s/step - loss: 0.4567 - accuracy: 0.9
575 - val_loss: 0.9398 - val_accuracy: 0.9249
```





```
<ipython-input-17-f305f83a4b85>:79: UserWarning: `Model.evaluate_generator` is depre
cated and will be removed in a future version. Please use `Model.evaluate`, which su
pports generators.
  test_loss, test_acc = model.evaluate_generator(test_set)
Test Loss: 0.49902674555778503
Test Accuracy: 0.9528301954269409
```

In [ ]: