In [ ]:
```
import os
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.applications import MobileNet
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.models import load_model
```

In [ ]:
```
# File Directory for both the train and test
train_path = "chest-ctscan-images/Data/train"
val_path = "chest-ctscan-images/Data/valid"
test_path = "chest-ctscan-images/Data/test"
```

In [ ]:
```
def GetDatasetSize(path):
    num_of_image = {}
    for folder in os.listdir(path):
        # Counting the Number of Files in the Folder
        num_of_image[folder] = len(os.listdir(os.path.join(path, folder)))
    return num_of_image

train_set = GetDatasetSize(train_path)
val_set = GetDatasetSize(val_path)
test_set = GetDatasetSize(test_path)
print(train_set,"\n\n",val_set,"\n\n",test_set)
```

```
{'adenocarcinoma_left.lower.lobe_T2_N0_M0_Ib': 195, 'large.cell.carcinoma_left.hilum_
T2_N2_M0_IIIa': 115, 'normal': 148, 'squamous.cell.carcinoma_left.hilum_T1_N2_M0_III
a': 155}

 {'adenocarcinoma_left.lower.lobe_T2_N0_M0_Ib': 23, 'large.cell.carcinoma_left.hilum_
T2_N2_M0_IIIa': 21, 'normal': 13, 'squamous.cell.carcinoma_left.hilum_T1_N2_M0_IIIa':
15}

 {'adenocarcinoma': 120, 'large.cell.carcinoma': 51, 'normal': 54, 'squamous.cell.car
cinoma': 90}
```

In [ ]:
```
train_datagen = image.ImageDataGenerator(
    rescale=1.0/255.0,
    horizontal_flip=True,
    fill_mode='nearest',
    zoom_range=0.2,
    shear_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rotation_range=0.4
)
train_data = train_datagen.flow_from_directory(
    train_path,
    batch_size=5,
    target_size=(224, 224),
    class_mode='categorical'
)
```

```
Found 613 images belonging to 4 classes.
```

In [ ]:
```
val_datagen = image.ImageDataGenerator(rescale=1.0/255.0)
val_data = val_datagen.flow_from_directory(
    val_path,
    batch_size=5,
    target_size=(224, 224),
    class_mode='categorical'
)
```

```
Found 72 images belonging to 4 classes.
```

In [ ]:
```
test_datagen = image.ImageDataGenerator(rescale=1.0/255.0)
test_data = test_datagen.flow_from_directory(
    test_path,
    batch_size=5,
    target_size=(224, 224),
    class_mode='categorical'
)
```

```
Found 315 images belonging to 4 classes.
```

In [ ]:
```
# MobileNet Model
mob_model = MobileNet(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet'
)
```

In [ ]:
```
for layer in mob_model.layers:
    layer.trainable = False

x = GlobalAveragePooling2D()(mob_model.output)
x = Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)

# Add a final sigmoid layer with 4 nodes for classification output
output = Dense(4, activation='sigmoid')(x)
```

In [ ]:
```
model_mobnet = Model(mob_model.input, output)

model_mobnet.compile(
    optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

In [ ]:
```
mc = ModelCheckpoint(
    filepath="./mobnet_model.hdf5",
    monitor='val_accuracy',
    verbose=1,
    save_best_only=True,
    mode='auto'
)
```

In [ ]:
```
# Fitting the Model
history = model_mobnet.fit(
    train_data,
    steps_per_epoch=train_data.samples // train_data.batch_size,
    epochs=35,
```

```
    validation_data=val_data,
    validation_steps=val_data.samples // val_data.batch_size,
    callbacks=[mc]
)
```

```
Epoch 1/35
122/122 [==============================] - ETA: 0s - loss: 1.1402 - accuracy: 0.4918
Epoch 1: val_accuracy improved from -inf to 0.51429, saving model to .\mobnet_model.h
df5
122/122 [==============================] - 70s 416ms/step - loss: 1.1402 - accuracy:
0.4918 - val_loss: 0.9128 - val_accuracy: 0.5143
Epoch 2/35
122/122 [==============================] - ETA: 0s - loss: 0.8069 - accuracy: 0.6349
Epoch 2: val_accuracy improved from 0.51429 to 0.58571, saving model to .\mobnet_mode
l.hdf5
122/122 [==============================] - 48s 391ms/step - loss: 0.8069 - accuracy:
0.6349 - val_loss: 0.8205 - val_accuracy: 0.5857
Epoch 3/35
122/122 [==============================] - ETA: 0s - loss: 0.6981 - accuracy: 0.7007
Epoch 3: val_accuracy improved from 0.58571 to 0.62857, saving model to .\mobnet_mode
l.hdf5
122/122 [==============================] - 48s 393ms/step - loss: 0.6981 - accuracy:
0.7007 - val_loss: 0.8644 - val_accuracy: 0.6286
Epoch 4/35
122/122 [==============================] - ETA: 0s - loss: 0.6527 - accuracy: 0.7303
Epoch 4: val_accuracy improved from 0.62857 to 0.67143, saving model to .\mobnet_mode
l.hdf5
122/122 [==============================] - 49s 402ms/step - loss: 0.6527 - accuracy:
0.7303 - val_loss: 0.8299 - val_accuracy: 0.6714
Epoch 5/35
122/122 [==============================] - ETA: 0s - loss: 0.5630 - accuracy: 0.7632
Epoch 5: val_accuracy did not improve from 0.67143
122/122 [==============================] - 46s 376ms/step - loss: 0.5630 - accuracy:
0.7632 - val_loss: 0.6899 - val_accuracy: 0.6571
Epoch 6/35
122/122 [==============================] - ETA: 0s - loss: 0.5179 - accuracy: 0.7928
Epoch 6: val_accuracy improved from 0.67143 to 0.75714, saving model to .\mobnet_mode
l.hdf5
122/122 [==============================] - 48s 395ms/step - loss: 0.5179 - accuracy:
0.7928 - val_loss: 0.6658 - val_accuracy: 0.7571
Epoch 7/35
122/122 [==============================] - ETA: 0s - loss: 0.4955 - accuracy: 0.7977
Epoch 7: val_accuracy did not improve from 0.75714
122/122 [==============================] - 45s 371ms/step - loss: 0.4955 - accuracy:
0.7977 - val_loss: 0.6643 - val_accuracy: 0.7429
Epoch 8/35
122/122 [==============================] - ETA: 0s - loss: 0.4908 - accuracy: 0.8026
Epoch 8: val_accuracy did not improve from 0.75714
122/122 [==============================] - 46s 372ms/step - loss: 0.4908 - accuracy:
0.8026 - val_loss: 0.7295 - val_accuracy: 0.6857
Epoch 9/35
122/122 [==============================] - ETA: 0s - loss: 0.4419 - accuracy: 0.8158
Epoch 9: val_accuracy did not improve from 0.75714
122/122 [==============================] - 46s 374ms/step - loss: 0.4419 - accuracy:
0.8158 - val_loss: 0.8464 - val_accuracy: 0.6714
Epoch 10/35
122/122 [==============================] - ETA: 0s - loss: 0.3738 - accuracy: 0.8520
Epoch 10: val_accuracy did not improve from 0.75714
122/122 [==============================] - 47s 388ms/step - loss: 0.3738 - accuracy:
0.8520 - val_loss: 0.8292 - val_accuracy: 0.6714
Epoch 11/35
122/122 [==============================] - ETA: 0s - loss: 0.4030 - accuracy: 0.8240
Epoch 11: val_accuracy did not improve from 0.75714
122/122 [==============================] - 46s 377ms/step - loss: 0.4030 - accuracy:
0.8240 - val_loss: 0.7387 - val_accuracy: 0.6714
```

```
Epoch 12/35
122/122 [==============================] - ETA: 0s - loss: 0.3924 - accuracy: 0.8273
Epoch 12: val_accuracy did not improve from 0.75714
122/122 [==============================] - 45s 371ms/step - loss: 0.3924 - accuracy:
0.8273 - val_loss: 0.7019 - val_accuracy: 0.7286
Epoch 13/35
122/122 [==============================] - ETA: 0s - loss: 0.3721 - accuracy: 0.8470
Epoch 13: val_accuracy did not improve from 0.75714
122/122 [==============================] - 46s 372ms/step - loss: 0.3721 - accuracy:
0.8470 - val_loss: 0.6663 - val_accuracy: 0.7429
Epoch 14/35
122/122 [==============================] - ETA: 0s - loss: 0.3011 - accuracy: 0.8865
Epoch 14: val_accuracy improved from 0.75714 to 0.80000, saving model to .\mobnet_mod
el.hdf5
122/122 [==============================] - 49s 401ms/step - loss: 0.3011 - accuracy:
0.8865 - val_loss: 0.5523 - val_accuracy: 0.8000
Epoch 15/35
122/122 [==============================] - ETA: 0s - loss: 0.3178 - accuracy: 0.8766
Epoch 15: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 373ms/step - loss: 0.3178 - accuracy:
0.8766 - val_loss: 0.5559 - val_accuracy: 0.7857
Epoch 16/35
122/122 [==============================] - ETA: 0s - loss: 0.2799 - accuracy: 0.8947
Epoch 16: val_accuracy did not improve from 0.80000
122/122 [==============================] - 45s 371ms/step - loss: 0.2799 - accuracy:
0.8947 - val_loss: 0.9202 - val_accuracy: 0.7000
Epoch 17/35
122/122 [==============================] - ETA: 0s - loss: 0.2983 - accuracy: 0.8766
Epoch 17: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 373ms/step - loss: 0.2983 - accuracy:
0.8766 - val_loss: 0.6666 - val_accuracy: 0.7429
Epoch 18/35
122/122 [==============================] - ETA: 0s - loss: 0.3041 - accuracy: 0.8783
Epoch 18: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 376ms/step - loss: 0.3041 - accuracy:
0.8783 - val_loss: 0.7299 - val_accuracy: 0.7429
Epoch 19/35
122/122 [==============================] - ETA: 0s - loss: 0.2703 - accuracy: 0.9030
Epoch 19: val_accuracy did not improve from 0.80000
122/122 [==============================] - 45s 371ms/step - loss: 0.2703 - accuracy:
0.9030 - val_loss: 0.7079 - val_accuracy: 0.7571
Epoch 20/35
122/122 [==============================] - ETA: 0s - loss: 0.2852 - accuracy: 0.8898
Epoch 20: val_accuracy did not improve from 0.80000
122/122 [==============================] - 45s 371ms/step - loss: 0.2852 - accuracy:
0.8898 - val_loss: 0.7498 - val_accuracy: 0.7000
Epoch 21/35
122/122 [==============================] - ETA: 0s - loss: 0.2226 - accuracy: 0.9243
Epoch 21: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 374ms/step - loss: 0.2226 - accuracy:
0.9243 - val_loss: 0.8325 - val_accuracy: 0.6857
Epoch 22/35
122/122 [==============================] - ETA: 0s - loss: 0.2147 - accuracy: 0.9293
Epoch 22: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 376ms/step - loss: 0.2147 - accuracy:
0.9293 - val_loss: 0.6188 - val_accuracy: 0.7714
Epoch 23/35
122/122 [==============================] - ETA: 0s - loss: 0.2591 - accuracy: 0.9046
Epoch 23: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 374ms/step - loss: 0.2591 - accuracy:
```

```
0.9046 - val_loss: 0.6512 - val_accuracy: 0.7429
Epoch 24/35
122/122 [==============================] - ETA: 0s - loss: 0.2221 - accuracy: 0.9178
Epoch 24: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 375ms/step - loss: 0.2221 - accuracy:
0.9178 - val_loss: 0.7707 - val_accuracy: 0.7714
Epoch 25/35
122/122 [==============================] - ETA: 0s - loss: 0.2371 - accuracy: 0.9095
Epoch 25: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 374ms/step - loss: 0.2371 - accuracy:
0.9095 - val_loss: 0.6792 - val_accuracy: 0.7571
Epoch 26/35
122/122 [==============================] - ETA: 0s - loss: 0.2301 - accuracy: 0.9145
Epoch 26: val_accuracy did not improve from 0.80000
122/122 [==============================] - 46s 378ms/step - loss: 0.2301 - accuracy:
0.9145 - val_loss: 0.6061 - val_accuracy: 0.7714
Epoch 27/35
122/122 [==============================] - ETA: 0s - loss: 0.2064 - accuracy: 0.9211
Epoch 27: val_accuracy improved from 0.80000 to 0.81429, saving model to .\mobnet_mod
el.hdf5
122/122 [==============================] - 48s 396ms/step - loss: 0.2064 - accuracy:
0.9211 - val_loss: 0.5797 - val_accuracy: 0.8143
Epoch 28/35
122/122 [==============================] - ETA: 0s - loss: 0.1759 - accuracy: 0.9457
Epoch 28: val_accuracy did not improve from 0.81429
122/122 [==============================] - 46s 379ms/step - loss: 0.1759 - accuracy:
0.9457 - val_loss: 0.8934 - val_accuracy: 0.7429
Epoch 29/35
122/122 [==============================] - ETA: 0s - loss: 0.1932 - accuracy: 0.9276
Epoch 29: val_accuracy did not improve from 0.81429
122/122 [==============================] - 46s 373ms/step - loss: 0.1932 - accuracy:
0.9276 - val_loss: 0.6883 - val_accuracy: 0.7429
Epoch 30/35
122/122 [==============================] - ETA: 0s - loss: 0.1968 - accuracy: 0.9260
Epoch 30: val_accuracy did not improve from 0.81429
122/122 [==============================] - 42s 343ms/step - loss: 0.1968 - accuracy:
0.9260 - val_loss: 0.6156 - val_accuracy: 0.7714
Epoch 31/35
122/122 [==============================] - ETA: 0s - loss: 0.1911 - accuracy: 0.9260
Epoch 31: val_accuracy did not improve from 0.81429
122/122 [==============================] - 48s 392ms/step - loss: 0.1911 - accuracy:
0.9260 - val_loss: 0.5657 - val_accuracy: 0.7857
Epoch 32/35
122/122 [==============================] - ETA: 0s - loss: 0.1764 - accuracy: 0.9359
Epoch 32: val_accuracy did not improve from 0.81429
122/122 [==============================] - 47s 382ms/step - loss: 0.1764 - accuracy:
0.9359 - val_loss: 0.6726 - val_accuracy: 0.7857
Epoch 33/35
122/122 [==============================] - ETA: 0s - loss: 0.1919 - accuracy: 0.9375
Epoch 33: val_accuracy did not improve from 0.81429
122/122 [==============================] - 47s 384ms/step - loss: 0.1919 - accuracy:
0.9375 - val_loss: 0.6913 - val_accuracy: 0.7714
Epoch 34/35
122/122 [==============================] - ETA: 0s - loss: 0.1837 - accuracy: 0.9326
Epoch 34: val_accuracy did not improve from 0.81429
122/122 [==============================] - 47s 382ms/step - loss: 0.1837 - accuracy:
0.9326 - val_loss: 0.6841 - val_accuracy: 0.7714
Epoch 35/35
122/122 [==============================] - ETA: 0s - loss: 0.1672 - accuracy: 0.9375
Epoch 35: val_accuracy improved from 0.81429 to 0.85714, saving model to .\mobnet_mod
```

```
el.hdf5
122/122 [==============================] - 50s 406ms/step - loss: 0.1672 - accuracy:
0.9375 - val_loss: 0.5719 - val_accuracy: 0.8571
```

In [ ]:
```
model_mobnet = load_model("mobnet_model.hdf5")
```

In [ ]:
```
# Evaluate the Model
test_score = model_mobnet.evaluate(test_data)
accuracy = test_score[1] * 100
print(f"Test Accuracy Mobile Net: {accuracy:.2f}%")
loss = test_score[0] * 100
print(f"Model Loss: {loss:.2f}%")
```

```
63/63 [==============================] - 20s 296ms/step - loss: 0.5084 - accuracy: 0.
8444
Test Accuracy Mobile Net: 84.44%
Model Loss: 50.84%
```

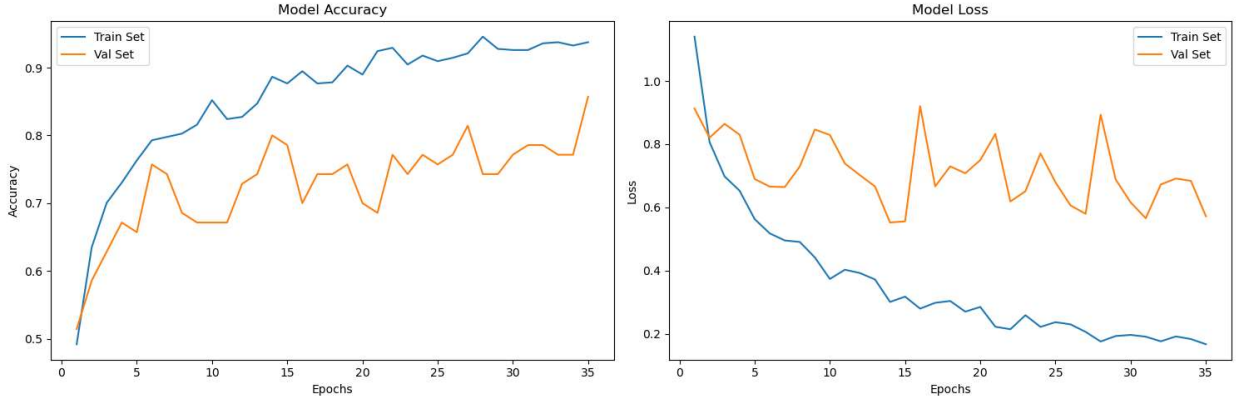
In [ ]:
```
# Plot model performance
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs_range = range(1, len(history.epoch) + 1)

plt.figure(figsize=(15,5))

plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Train Set')
plt.plot(epochs_range, val_acc, label='Val Set')
plt.legend(loc="best")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Model Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Train Set')
plt.plot(epochs_range, val_loss, label='Val Set')
plt.legend(loc="best")
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Model Loss')

plt.tight_layout()
plt.show()
```



In [ ]:
```python
def chestScanPrediction(path, model):
    classes_dir = ["Adenocarcinoma", "Large cell carcinoma", "Normal", "Squamous cell

    # Loading Image
    img = image.load_img(path, target_size=(224, 224))
    # Converting Image to Numpy Array
    input_arr = image.img_to_array(img)
    # Preprocess the input image
    input_arr = tf.keras.applications.mobilenet.preprocess_input(input_arr)
    # Adding extra dimension to match the model input shape
    input_arr = np.expand_dims(input_arr, axis=0)

    # Getting Predictions
    pred = model.predict(input_arr)
    # Converting probability to percentage
    prob = round(np.max(pred) * 100, 2)
    # Getting predicted class index
    pred_index = np.argmax(pred)
    # Getting predicted class
    pred_class = classes_dir[pred_index]

    # Returning prediction result
    return {"class": pred_class, "probability": prob}
```

In [ ]:
```python
# path = "chest-ctscan-images/Data/test/large.cell.carcinoma/000110.png"
# path = "chest-ctscan-images/Data/test/adenocarcinoma/000114.png"
path = "chest-ctscan-images/Data/test/squamous.cell.carcinoma/000119.png"
# path = "chest-ctscan-images/Data/test/normal/7.png"
chestScanPrediction(path,model_mobnet)
```

Out[ ]:
```
1/1 [==============================] - 1s 1s/step
{'class': 'Squamous cell carcinoma', 'probability': 97.83}
```

In [ ]:
```python

```