

In [1]:

```
from tensorflow.keras.callbacks import LearningRateScheduler
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib.image as mpimg
import tensorflow as tf
import os

ACC=0.1
try_num = 1

while (ACC<0.88 and try_num<50):
    # DOE factors:
    learning_rate = 0.001
    dropout_value = 0.1
    # n-conv_layers = 3
    n_units_last_layer = 2048
    n_filters_l1 = 16
    n_filters_l2 = 32

    # other factors:
    img_size = 130
    batch_size = 32
    validation_split = 0.1 # 10% for validation
    test_split = 0.00 # 0% for testing
    shuffle_buffer_size = 1000
    seed_num = 101
    desired_accuracy = 0.99 # it should be active if EarlyStoppingCallback is a
    loss = 'binary_crossentropy'
    #optimizer = tf.keras.optimizers.RMSprop(Learning_rate=learning_rate)
    optimizer = tf.keras.optimizers.Adam(learning_rate=learning_rate)
    metrics = ['accuracy']
    epochs = 18
    f_mode = 'nearest' # fill_mode in image augmentation

    #DATA_DIR = "D:\\CS online courses\\Free DataSets\\Free Images\\Easier portr
    DATA_DIR = "/Users/hossein/Downloads/Easier portrait images_GPU_03"

    # Subdirectories for each class
    data_dir_woman = os.path.join(DATA_DIR, 'woman')
    data_dir_man = os.path.join(DATA_DIR, 'man')
    image_size = (img_size, img_size) # Resize images to this size
    # Load train dataset (excluding validation & test set):
    train_dataset = tf.keras.utils.image_dataset_from_directory(
        directory = DATA_DIR,
        image_size = image_size,
        batch_size = batch_size,
        label_mode='binary',
        validation_split = validation_split + test_split, # Total split for val
        subset = "training",
        seed = seed_num
    )
    # Load validation dataset
    val_dataset = tf.keras.utils.image_dataset_from_directory(
        directory = DATA_DIR,
```

```

        image_size = image_size,
        batch_size = batch_size,
        label_mode='binary',
        validation_split = validation_split + test_split,
        subset = "validation",
        seed = seed_num
    )
# Further manually split validation dataset to extract test dataset
val_batches = tf.data.experimental.cardinality(val_dataset)
# Compute test dataset size (number of batches)
test_size = round(val_batches.numpy() * (test_split / (validation_split + te
# Split validation dataset into validation and test subsets
test_dataset = val_dataset.take(test_size)
val_dataset = val_dataset.skip(test_size)
# Optimize for performance
AUTOTUNE = tf.data.AUTOTUNE
training_dataset = train_dataset.cache().shuffle(shuffle_buffer_size).prefet
validation_dataset = val_dataset.cache().prefetch(buffer_size = AUTOTUNE)
test_dataset = test_dataset.cache().prefetch(buffer_size = AUTOTUNE)

# Get the first batch of images and labels
for images, labels in training_dataset.take(1):
    example_batch_images = images
    example_batch_labels = labels
max_pixel = np.max(example_batch_images)

# Reduce LR every 10 epochs (Learning rate decay factor)
def scheduler(epoch, lr):
    if epoch < 10:
        if epoch % 5 == 0 and epoch > 0:
            return lr / 1.1
        return lr
    elif epoch < 15:
        if epoch % 5 == 0 and epoch > 0:
            return lr / 5
        return lr
    elif epoch < 30:
        if epoch % 5 == 0 and epoch > 0:
            return lr / 1
        return lr
    else:
        return lr
lr_callback = LearningRateScheduler(scheduler)

# augmentation_model
def augment_model():
    augmentation_model = tf.keras.Sequential([
        # Specify the input shape.
        tf.keras.Input(shape = (img_size, img_size, 3)),

        tf.keras.layers.RandomFlip("horizontal"),
        tf.keras.layers.RandomRotation(0.1, fill_mode = f_mode),
        #tf.keras.layers.RandomTranslation(0.1, 0.1, fill_mode = f_mode),
        #tf.keras.layers.RandomZoom(0.1, fill_mode=f_mode)
    ])
    return augmentation_model

def create_and_compile_model():
    augmentation_layers = augment_model()
    model = tf.keras.Sequential([

```

```

# Note: the input shape is the desired size of the image: 150x150 wi
tf.keras.layers.InputLayer(shape = (img_size, img_size, 3)),
augmentation_layers,
tf.keras.layers.Rescaling(1./255),
##### CONV_LAYER_1: #####
tf.keras.layers.Conv2D(n_filters_l1, (4, 4), activation = 'linear'),
tf.keras.layers.MaxPooling2D(2, 2),
##### CONV_LAYER_2: #####
tf.keras.layers.Conv2D(n_filters_l2, (3, 3), activation = 'relu'),
tf.keras.layers.MaxPooling2D(2, 2),
##### CONV_LAYER_3: #####
tf.keras.layers.Conv2D(64, (3, 3), activation = 'relu'),
tf.keras.layers.MaxPooling2D(2, 2),
tf.keras.layers.Flatten(),
tf.keras.layers.Dropout(dropout_value),
##### BEFORE_LAST_LAYER: #####
tf.keras.layers.Dense(n_units_last_layer, activation = 'relu'),
# It will contain a value from 0-1 where 0 for the class 'female' an
tf.keras.layers.Dense(1, activation = 'sigmoid')))

model.compile(
    loss = loss,
    optimizer = optimizer,
    metrics = metrics
)
return model

# Create the compiled but untrained model
def reset_weights(model):
    for layer in model.layers:
        if hasattr(layer, 'kernel_initializer'):
            layer.kernel.assign(layer.kernel_initializer(layer.kernel.shape))
        if hasattr(layer, 'bias_initializer'):
            layer.bias.assign(layer.bias_initializer(layer.bias.shape))

model = create_and_compile_model()
reset_weights(model) # Reset all layer weights
training_history = model.fit(training_dataset,
                             epochs=epochs,
                             validation_data=validation_dataset,
                             callbacks=[lr_callback],
                             verbose=2)
result_history = pd.DataFrame(model.history.history)
ACC = result_history['val_accuracy'].iloc[-1]
print(f"Current validation accuracy: {ACC}")
model.save('trained_model_run37_advanced_control.h5')
# Restart script
print("Reseting all weights...")
print(f'Current number of trials: {try_num}')
try_num += 1

result_history.head(15)
result_history[['loss', 'val_loss']].plot(figsize=(5, 3))
result_history[['accuracy', 'val_accuracy']].plot(figsize=(5, 3))
plt.show()
print(model.metrics_names)
print(model.evaluate(validation_dataset))
y_true = np.concatenate([y.numpy() for _, y in validation_dataset])
y_pred_prob = model.predict(validation_dataset)
# Convert probabilities to class labels (0:Female or 1:Male)

```

```
y_pred = (y_pred_prob > 0.5).astype(int).flatten()
print("Classification Report:\n", classification_report(y_true, y_pred, target_n
```

Found 943 files belonging to 2 classes.  
Using 849 files for training.  
Found 943 files belonging to 2 classes.  
Using 94 files for validation.  
Epoch 1/18

2025-05-05 09:03:54.623203: I tensorflow/core/framework/local\_rendezvous.cc:405] Local rendezvous is aborting with status: OUT\_OF\_RANGE: End of sequence

```
27/27 - 3s - 99ms/step - accuracy: 0.5842 - loss: 0.8365 - val_accuracy: 0.7021 -  
val_loss: 0.5900 - learning_rate: 1.0000e-03  
Epoch 2/18  
27/27 - 2s - 76ms/step - accuracy: 0.7303 - loss: 0.5312 - val_accuracy: 0.7447 -  
val_loss: 0.5771 - learning_rate: 1.0000e-03  
Epoch 3/18  
27/27 - 2s - 75ms/step - accuracy: 0.7892 - loss: 0.4598 - val_accuracy: 0.8617 -  
val_loss: 0.3828 - learning_rate: 1.0000e-03  
Epoch 4/18  
27/27 - 2s - 75ms/step - accuracy: 0.7856 - loss: 0.4565 - val_accuracy: 0.8723 -  
val_loss: 0.3858 - learning_rate: 1.0000e-03  
Epoch 5/18  
27/27 - 2s - 75ms/step - accuracy: 0.8233 - loss: 0.4250 - val_accuracy: 0.8511 -  
val_loss: 0.3681 - learning_rate: 1.0000e-03  
Epoch 6/18  
27/27 - 2s - 75ms/step - accuracy: 0.8363 - loss: 0.3870 - val_accuracy: 0.7979 -  
val_loss: 0.4342 - learning_rate: 9.0909e-04  
Epoch 7/18  
27/27 - 2s - 75ms/step - accuracy: 0.8598 - loss: 0.3216 - val_accuracy: 0.8191 -  
val_loss: 0.5233 - learning_rate: 9.0909e-04  
Epoch 8/18  
27/27 - 2s - 75ms/step - accuracy: 0.8751 - loss: 0.3151 - val_accuracy: 0.8404 -  
val_loss: 0.4190 - learning_rate: 9.0909e-04  
Epoch 9/18  
27/27 - 2s - 77ms/step - accuracy: 0.8693 - loss: 0.3002 - val_accuracy: 0.7979 -  
val_loss: 0.5181 - learning_rate: 9.0909e-04  
Epoch 10/18  
27/27 - 2s - 75ms/step - accuracy: 0.8551 - loss: 0.3123 - val_accuracy: 0.8404 -  
val_loss: 0.3822 - learning_rate: 9.0909e-04  
Epoch 11/18  
27/27 - 2s - 75ms/step - accuracy: 0.8822 - loss: 0.2719 - val_accuracy: 0.8191 -  
val_loss: 0.3558 - learning_rate: 1.8182e-04  
Epoch 12/18  
27/27 - 2s - 75ms/step - accuracy: 0.9058 - loss: 0.2469 - val_accuracy: 0.8404 -  
val_loss: 0.3580 - learning_rate: 1.8182e-04  
Epoch 13/18  
27/27 - 2s - 76ms/step - accuracy: 0.9117 - loss: 0.2257 - val_accuracy: 0.8298 -  
val_loss: 0.3922 - learning_rate: 1.8182e-04  
Epoch 14/18  
27/27 - 2s - 75ms/step - accuracy: 0.9234 - loss: 0.2277 - val_accuracy: 0.8298 -  
val_loss: 0.3646 - learning_rate: 1.8182e-04  
Epoch 15/18  
27/27 - 2s - 75ms/step - accuracy: 0.9140 - loss: 0.2167 - val_accuracy: 0.8511 -  
val_loss: 0.3934 - learning_rate: 1.8182e-04  
Epoch 16/18  
27/27 - 2s - 76ms/step - accuracy: 0.9176 - loss: 0.2302 - val_accuracy: 0.8511 -  
val_loss: 0.3786 - learning_rate: 1.8182e-04  
Epoch 17/18  
27/27 - 2s - 77ms/step - accuracy: 0.9199 - loss: 0.2135 - val_accuracy: 0.8404 -  
val_loss: 0.4177 - learning_rate: 1.8182e-04  
Epoch 18/18  
27/27 - 2s - 75ms/step - accuracy: 0.9246 - loss: 0.1978 - val_accuracy: 0.8191 -  
val_loss: 0.3870 - learning_rate: 1.8182e-04
```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or `keras.saving.save\_model(model, 'my\_model.keras')`.

```
Current validation accuracy: 0.8191489577293396
Reseting all weights...
Current number of trials: 1
Found 943 files belonging to 2 classes.
Using 849 files for training.
Found 943 files belonging to 2 classes.
Using 94 files for validation.
Epoch 1/18
```

```
2025-05-05 09:04:32.331632: I tensorflow/core/framework/local_rendezvous.cc:405]
Local rendezvous is aborting with status: OUT_OF_RANGE: End of sequence
```

```
27/27 - 3s - 99ms/step - accuracy: 0.5736 - loss: 0.8871 - val_accuracy: 0.6489 -  
val_loss: 0.6211 - learning_rate: 1.0000e-03  
Epoch 2/18  
27/27 - 2s - 76ms/step - accuracy: 0.7185 - loss: 0.5594 - val_accuracy: 0.7766 -  
val_loss: 0.4961 - learning_rate: 1.0000e-03  
Epoch 3/18  
27/27 - 2s - 76ms/step - accuracy: 0.7821 - loss: 0.4638 - val_accuracy: 0.8298 -  
val_loss: 0.4118 - learning_rate: 1.0000e-03  
Epoch 4/18  
27/27 - 2s - 76ms/step - accuracy: 0.7927 - loss: 0.4230 - val_accuracy: 0.8617 -  
val_loss: 0.4378 - learning_rate: 1.0000e-03  
Epoch 5/18  
27/27 - 2s - 75ms/step - accuracy: 0.8292 - loss: 0.3985 - val_accuracy: 0.8511 -  
val_loss: 0.3285 - learning_rate: 1.0000e-03  
Epoch 6/18  
27/27 - 2s - 77ms/step - accuracy: 0.8245 - loss: 0.3879 - val_accuracy: 0.8511 -  
val_loss: 0.4075 - learning_rate: 9.0909e-04  
Epoch 7/18  
27/27 - 2s - 75ms/step - accuracy: 0.8457 - loss: 0.3479 - val_accuracy: 0.8404 -  
val_loss: 0.3480 - learning_rate: 9.0909e-04  
Epoch 8/18  
27/27 - 2s - 75ms/step - accuracy: 0.8410 - loss: 0.3686 - val_accuracy: 0.8617 -  
val_loss: 0.3315 - learning_rate: 9.0909e-04  
Epoch 9/18  
27/27 - 2s - 76ms/step - accuracy: 0.8634 - loss: 0.3182 - val_accuracy: 0.8511 -  
val_loss: 0.4516 - learning_rate: 9.0909e-04  
Epoch 10/18  
27/27 - 2s - 75ms/step - accuracy: 0.8645 - loss: 0.3354 - val_accuracy: 0.8511 -  
val_loss: 0.3740 - learning_rate: 9.0909e-04  
Epoch 11/18  
27/27 - 2s - 75ms/step - accuracy: 0.8869 - loss: 0.2903 - val_accuracy: 0.8404 -  
val_loss: 0.3080 - learning_rate: 1.8182e-04  
Epoch 12/18  
27/27 - 2s - 76ms/step - accuracy: 0.8940 - loss: 0.2558 - val_accuracy: 0.8298 -  
val_loss: 0.3368 - learning_rate: 1.8182e-04  
Epoch 13/18  
27/27 - 2s - 76ms/step - accuracy: 0.8975 - loss: 0.2485 - val_accuracy: 0.8404 -  
val_loss: 0.3370 - learning_rate: 1.8182e-04  
Epoch 14/18  
27/27 - 2s - 77ms/step - accuracy: 0.9081 - loss: 0.2292 - val_accuracy: 0.8298 -  
val_loss: 0.3500 - learning_rate: 1.8182e-04  
Epoch 15/18  
27/27 - 2s - 75ms/step - accuracy: 0.9117 - loss: 0.2180 - val_accuracy: 0.8298 -  
val_loss: 0.3458 - learning_rate: 1.8182e-04  
Epoch 16/18  
27/27 - 2s - 75ms/step - accuracy: 0.9011 - loss: 0.2274 - val_accuracy: 0.8511 -  
val_loss: 0.3981 - learning_rate: 1.8182e-04  
Epoch 17/18  
27/27 - 2s - 76ms/step - accuracy: 0.9223 - loss: 0.2095 - val_accuracy: 0.8404 -  
val_loss: 0.3417 - learning_rate: 1.8182e-04  
Epoch 18/18  
27/27 - 2s - 75ms/step - accuracy: 0.9105 - loss: 0.2132 - val_accuracy: 0.8511 -  
val_loss: 0.3807 - learning_rate: 1.8182e-04
```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or `keras.saving.save\_model(model, 'my\_model.keras')`.

```
Current validation accuracy: 0.8510638475418091
Reseting all weights...
Current number of trials: 2
Found 943 files belonging to 2 classes.
Using 849 files for training.
Found 943 files belonging to 2 classes.
Using 94 files for validation.
Epoch 1/18
27/27 - 3s - 96ms/step - accuracy: 0.6196 - loss: 0.7737 - val_accuracy: 0.6809 -
val_loss: 0.6570 - learning_rate: 1.0000e-03
Epoch 2/18
27/27 - 2s - 77ms/step - accuracy: 0.7126 - loss: 0.5492 - val_accuracy: 0.7553 -
val_loss: 0.5327 - learning_rate: 1.0000e-03
Epoch 3/18
27/27 - 2s - 77ms/step - accuracy: 0.7515 - loss: 0.5017 - val_accuracy: 0.7447 -
val_loss: 0.6311 - learning_rate: 1.0000e-03
Epoch 4/18
27/27 - 2s - 76ms/step - accuracy: 0.7585 - loss: 0.4983 - val_accuracy: 0.8085 -
val_loss: 0.4348 - learning_rate: 1.0000e-03
Epoch 5/18
27/27 - 2s - 75ms/step - accuracy: 0.8092 - loss: 0.4216 - val_accuracy: 0.8191 -
val_loss: 0.3837 - learning_rate: 1.0000e-03
Epoch 6/18
27/27 - 2s - 76ms/step - accuracy: 0.8210 - loss: 0.4175 - val_accuracy: 0.8617 -
val_loss: 0.4047 - learning_rate: 9.0909e-04
Epoch 7/18
27/27 - 2s - 76ms/step - accuracy: 0.8327 - loss: 0.3813 - val_accuracy: 0.8404 -
val_loss: 0.4233 - learning_rate: 9.0909e-04
Epoch 8/18
27/27 - 2s - 76ms/step - accuracy: 0.8481 - loss: 0.3652 - val_accuracy: 0.8830 -
val_loss: 0.3422 - learning_rate: 9.0909e-04
Epoch 9/18
27/27 - 2s - 76ms/step - accuracy: 0.8704 - loss: 0.3121 - val_accuracy: 0.8723 -
val_loss: 0.3719 - learning_rate: 9.0909e-04
Epoch 10/18
27/27 - 2s - 76ms/step - accuracy: 0.8669 - loss: 0.3217 - val_accuracy: 0.8404 -
val_loss: 0.3518 - learning_rate: 9.0909e-04
Epoch 11/18
27/27 - 2s - 78ms/step - accuracy: 0.8728 - loss: 0.2987 - val_accuracy: 0.8511 -
val_loss: 0.3787 - learning_rate: 1.8182e-04
Epoch 12/18
27/27 - 2s - 76ms/step - accuracy: 0.8869 - loss: 0.2620 - val_accuracy: 0.8511 -
val_loss: 0.4006 - learning_rate: 1.8182e-04
Epoch 13/18
27/27 - 2s - 76ms/step - accuracy: 0.9058 - loss: 0.2460 - val_accuracy: 0.8617 -
val_loss: 0.3594 - learning_rate: 1.8182e-04
Epoch 14/18
27/27 - 2s - 76ms/step - accuracy: 0.8881 - loss: 0.2768 - val_accuracy: 0.8617 -
val_loss: 0.3260 - learning_rate: 1.8182e-04
Epoch 15/18
27/27 - 2s - 76ms/step - accuracy: 0.8893 - loss: 0.2539 - val_accuracy: 0.8830 -
val_loss: 0.3460 - learning_rate: 1.8182e-04
Epoch 16/18
27/27 - 2s - 76ms/step - accuracy: 0.9034 - loss: 0.2400 - val_accuracy: 0.8723 -
val_loss: 0.3321 - learning_rate: 1.8182e-04
Epoch 17/18
27/27 - 2s - 76ms/step - accuracy: 0.9022 - loss: 0.2414 - val_accuracy: 0.8830 -
val_loss: 0.3544 - learning_rate: 1.8182e-04
Epoch 18/18
```

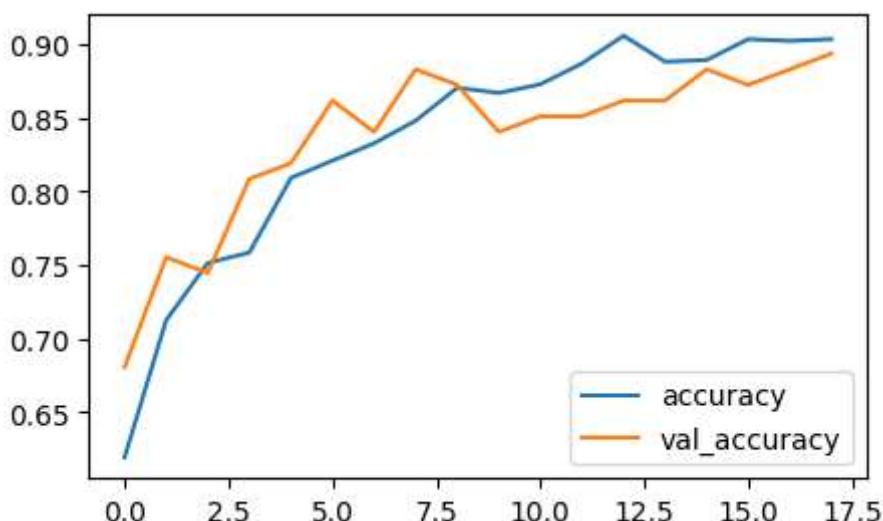
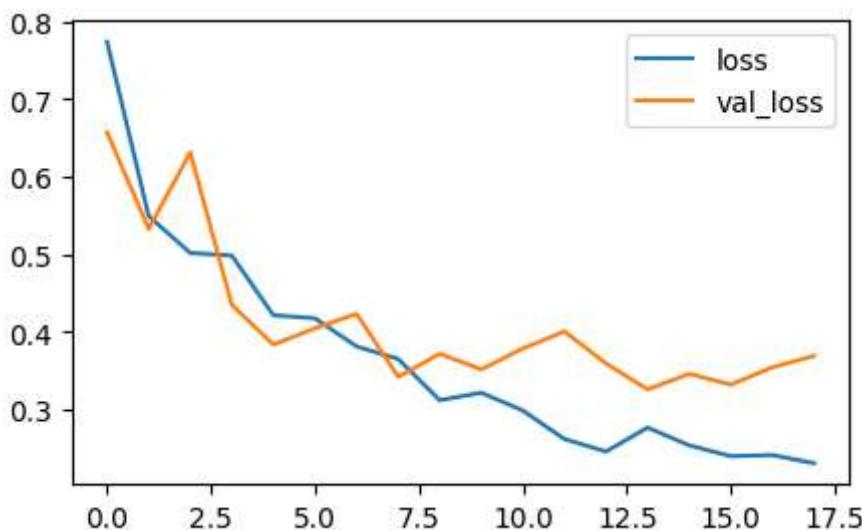
```
27/27 - 2s - 76ms/step - accuracy: 0.9034 - loss: 0.2309 - val_accuracy: 0.8936 - val_loss: 0.3691 - learning_rate: 1.8182e-04
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
```

```
Current validation accuracy: 0.8936170339584351
```

```
Reseting all weights...
```

```
Current number of trials: 3
```



```
['loss', 'compile_metrics']
3/3 0s 20ms/step - accuracy: 0.8921 - loss: 0.4016
[0.36914893984794617, 0.8936170339584351]
3/3 0s 31ms/step
```

```
Classification Report:
```

	precision	recall	f1-score	support
Female	0.82	0.98	0.89	41
Male	0.98	0.83	0.90	53
accuracy			0.89	94
macro avg	0.90	0.90	0.89	94
weighted avg	0.91	0.89	0.89	94

```
2025-05-05 09:05:48.068042: I tensorflow/core/framework/local_rendezvous.cc:405] Local rendezvous is aborting with status: OUT_OF_RANGE: End of sequence
```

In [ ]:

In [ ]: