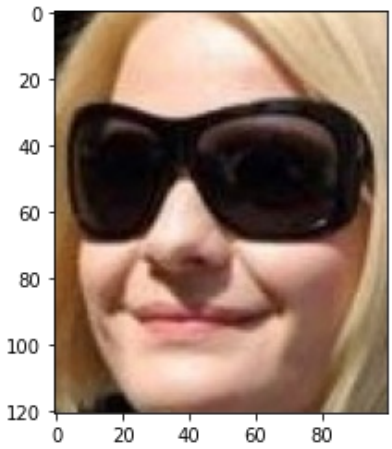
In [10]:

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.preprocessing import image
import keras
import cv2
from PIL import ImageFile
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
%matplotlib inline
from keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import pandas as pd
```

In [11]:

```
ImageFile.LOAD_TRUNCATED_IMAGES = True
img = image.load_img("/Users/mehme/Datasets/Gender Classification Dataset/gender/train/female/0005.jpg")
plt.imshow(img)

cv2.imread("/Users/mehme/Datasets/Gender Classification Dataset/gender/train/female/0005.jpg").shape

train = ImageDataGenerator(rescale = 1/255)
test = ImageDataGenerator(rescale = 1/255)
```



In [12]:

```
train_dataset = train.flow_from_directory("C:/Users/mehme/Datasets/Gender Classification Dataset/gender/train",
                                          target_size = (32,32),
                                             batch_size =3,
                                           class_mode =  "binary"
                                          )

test_dataset = test.flow_from_directory("C:/Users/mehme/Datasets/Gender Classification Dataset/gender/test",
                                          target_size = (32,32),
                                             batch_size =3,
                                           class_mode =  "binary"
                                          )
```

```
Found 3491 images belonging to 2 classes.
Found 200 images belonging to 2 classes.
```

In [13]:

```
train_dataset.class_indices
train_dataset.classes
```

Out[13]:

```
array([0, 0, 0, ..., 1, 1, 1])
```

In [14]:

```
test_dataset.class_indices
test_dataset.classes
```

Out[14]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1])
```

In [15]:

```
cnn = models.Sequential([
    tf.keras.layers.Conv2D(32,(3,3),activation="relu",input_shape=(32,32,3)),
    tf.keras.layers.MaxPool2D(2,2),


    tf.keras.layers.Conv2D(64,(3,3),activation="relu"),
    tf.keras.layers.MaxPool2D(2,2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64,activation="relu"),
    tf.keras.layers.Dense(2,activation="softmax")
])
```

In [16]:

```
cnn.compile(optimizer="adam",
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

In [17]:

```
history1 =cnn.fit(train_dataset,
       steps_per_epoch = 50,
       batch_size = 3,
       epochs = 100,
       validation_data = test_dataset

)
```

```
Epoch 1/100
50/50 [==============================] - 1s 10ms/step - loss: 0.7153 - accuracy: 0.5133 -
val_loss: 0.6970 - val_accuracy: 0.5000
Epoch 2/100
50/50 [==============================] - 0s 7ms/step - loss: 0.6951 - accuracy: 0.5333 -
val_loss: 0.7073 - val_accuracy: 0.5000
Epoch 3/100
50/50 [==============================] - 0s 6ms/step - loss: 0.6927 - accuracy: 0.5400 -
val_loss: 0.6812 - val_accuracy: 0.6500
Epoch 4/100
50/50 [==============================] - 0s 6ms/step - loss: 0.6919 - accuracy: 0.5667 -
val_loss: 0.7343 - val_accuracy: 0.5000
Epoch 5/100
```

```
50/50 [==============================] - 0s 6ms/step - loss: 0.6849 - accuracy: 0.5533 -
val_loss: 0.6439 - val_accuracy: 0.5750
Epoch 6/100
50/50 [==============================] - 0s 7ms/step - loss: 0.6497 - accuracy: 0.6267 -
val_loss: 0.6187 - val_accuracy: 0.7600
Epoch 7/100
50/50 [==============================] - 0s 6ms/step - loss: 0.6079 - accuracy: 0.7067 -
val_loss: 0.7411 - val_accuracy: 0.5400
Epoch 8/100
50/50 [==============================] - 0s 6ms/step - loss: 0.5472 - accuracy: 0.7200 -
val_loss: 0.6007 - val_accuracy: 0.6850
Epoch 9/100
50/50 [==============================] - 0s 6ms/step - loss: 0.5589 - accuracy: 0.7533 -
val_loss: 0.4405 - val_accuracy: 0.8500
Epoch 10/100
50/50 [==============================] - 0s 6ms/step - loss: 0.4956 - accuracy: 0.7733 -
val_loss: 0.4397 - val_accuracy: 0.8150
Epoch 11/100
50/50 [==============================] - 0s 6ms/step - loss: 0.5706 - accuracy: 0.6667 -
val_loss: 0.4932 - val_accuracy: 0.8600
Epoch 12/100
50/50 [==============================] - 0s 7ms/step - loss: 0.4822 - accuracy: 0.7800 -
val_loss: 0.3929 - val_accuracy: 0.8700
Epoch 13/100
50/50 [==============================] - 0s 6ms/step - loss: 0.5062 - accuracy: 0.7933 -
val_loss: 0.3859 - val_accuracy: 0.8300
Epoch 14/100
50/50 [==============================] - 0s 6ms/step - loss: 0.4473 - accuracy: 0.7919 -
val_loss: 0.3584 - val_accuracy: 0.8850
Epoch 15/100
50/50 [==============================] - 0s 6ms/step - loss: 0.4544 - accuracy: 0.7867 -
val_loss: 0.3470 - val_accuracy: 0.8700
Epoch 16/100
50/50 [==============================] - 0s 6ms/step - loss: 0.4420 - accuracy: 0.8133 -
val_loss: 0.4083 - val_accuracy: 0.8200
Epoch 17/100
50/50 [==============================] - 0s 6ms/step - loss: 0.3840 - accuracy: 0.8400 -
val_loss: 0.3320 - val_accuracy: 0.8750
Epoch 18/100
50/50 [==============================] - 0s 6ms/step - loss: 0.4097 - accuracy: 0.8267 -
val_loss: 0.3913 - val_accuracy: 0.8250
Epoch 19/100
50/50 [==============================] - 0s 8ms/step - loss: 0.4035 - accuracy: 0.8067 -
val_loss: 0.2884 - val_accuracy: 0.9100
Epoch 20/100
50/50 [==============================] - 0s 7ms/step - loss: 0.3175 - accuracy: 0.8733 -
val_loss: 0.2826 - val_accuracy: 0.9000
Epoch 21/100
50/50 [==============================] - 0s 7ms/step - loss: 0.4012 - accuracy: 0.8333 -
val_loss: 0.3022 - val_accuracy: 0.8850
Epoch 22/100
50/50 [==============================] - 0s 7ms/step - loss: 0.3183 - accuracy: 0.8667 -
val_loss: 0.3230 - val_accuracy: 0.8800
Epoch 23/100
50/50 [==============================] - 0s 6ms/step - loss: 0.3332 - accuracy: 0.8600 -
val_loss: 0.2501 - val_accuracy: 0.9100
Epoch 24/100
50/50 [==============================] - 0s 7ms/step - loss: 0.4103 - accuracy: 0.8467 -
val_loss: 0.3216 - val_accuracy: 0.8900
Epoch 25/100
50/50 [==============================] - 0s 6ms/step - loss: 0.3235 - accuracy: 0.8533 -
val_loss: 0.2387 - val_accuracy: 0.9200
Epoch 26/100
50/50 [==============================] - 0s 7ms/step - loss: 0.3498 - accuracy: 0.8600 -
val_loss: 0.2405 - val_accuracy: 0.9300
Epoch 27/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2518 - accuracy: 0.9333 -
val_loss: 0.2090 - val_accuracy: 0.9250
Epoch 28/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2414 - accuracy: 0.9067 -
val_loss: 0.2292 - val_accuracy: 0.9200
Epoch 29/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2670 - accuracy: 0.8867
```

50/50 [==============================] - 0s 6ms/step - loss: 0.2670 - accuracy: 0.8867 -
val_loss: 0.2840 - val_accuracy: 0.8950
Epoch 30/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2429 - accuracy: 0.9195 -
val_loss: 0.2295 - val_accuracy: 0.9250
Epoch 31/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2850 - accuracy: 0.8800 -
val_loss: 0.2096 - val_accuracy: 0.9400
Epoch 32/100
50/50 [==============================] - 0s 7ms/step - loss: 0.3073 - accuracy: 0.8800 -
val_loss: 0.2474 - val_accuracy: 0.9050
Epoch 33/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2277 - accuracy: 0.9067 -
val_loss: 0.2350 - val_accuracy: 0.9100
Epoch 34/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2231 - accuracy: 0.8933 -
val_loss: 0.1931 - val_accuracy: 0.9200
Epoch 35/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2076 - accuracy: 0.9200 -
val_loss: 0.1976 - val_accuracy: 0.9300
Epoch 36/100
50/50 [==============================] - 0s 6ms/step - loss: 0.3028 - accuracy: 0.8867 -
val_loss: 0.3438 - val_accuracy: 0.8400
Epoch 37/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2887 - accuracy: 0.9133 -
val_loss: 0.1940 - val_accuracy: 0.9350
Epoch 38/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2498 - accuracy: 0.9000 -
val_loss: 0.2179 - val_accuracy: 0.9150
Epoch 39/100
50/50 [==============================] - 0s 7ms/step - loss: 0.3536 - accuracy: 0.8133 -
val_loss: 0.2119 - val_accuracy: 0.9150
Epoch 40/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2504 - accuracy: 0.8867 -
val_loss: 0.1917 - val_accuracy: 0.9200
Epoch 41/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2837 - accuracy: 0.8800 -
val_loss: 0.1679 - val_accuracy: 0.9500
Epoch 42/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2645 - accuracy: 0.8800 -
val_loss: 0.1860 - val_accuracy: 0.9300
Epoch 43/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2575 - accuracy: 0.9000 -
val_loss: 0.1954 - val_accuracy: 0.9250
Epoch 44/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2825 - accuracy: 0.9060 -
val_loss: 0.1750 - val_accuracy: 0.9350
Epoch 45/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1723 - accuracy: 0.9133 -
val_loss: 0.1641 - val_accuracy: 0.9450
Epoch 46/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2371 - accuracy: 0.8933 -
val_loss: 0.2403 - val_accuracy: 0.9150
Epoch 47/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2575 - accuracy: 0.9400 -
val_loss: 0.1743 - val_accuracy: 0.9450
Epoch 48/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1010 - accuracy: 0.9733 -
val_loss: 0.2060 - val_accuracy: 0.9250
Epoch 49/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2715 - accuracy: 0.8667 -
val_loss: 0.1689 - val_accuracy: 0.9350
Epoch 50/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1691 - accuracy: 0.9333 -
val_loss: 0.1915 - val_accuracy: 0.9250
Epoch 51/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2218 - accuracy: 0.8933 -
val_loss: 0.1820 - val_accuracy: 0.9300
Epoch 52/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1829 - accuracy: 0.9133 -
val_loss: 0.2784 - val_accuracy: 0.8750
Epoch 53/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1766 - accuracy: 0.9133 -

50/50 [==============================] - 0s 7ms/step - loss: 0.1766 - accuracy: 0.9133 -
val_loss: 0.2344 - val_accuracy: 0.9200
Epoch 54/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2426 - accuracy: 0.9067 -
val_loss: 0.1969 - val_accuracy: 0.9250
Epoch 55/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1622 - accuracy: 0.9400 -
val_loss: 0.1848 - val_accuracy: 0.9350
Epoch 56/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1854 - accuracy: 0.9333 -
val_loss: 0.1595 - val_accuracy: 0.9400
Epoch 57/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2047 - accuracy: 0.9133 -
val_loss: 0.1798 - val_accuracy: 0.9350
Epoch 58/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1651 - accuracy: 0.9533 -
val_loss: 0.1661 - val_accuracy: 0.9400
Epoch 59/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1880 - accuracy: 0.9267 -
val_loss: 0.2065 - val_accuracy: 0.9250
Epoch 60/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2250 - accuracy: 0.9000 -
val_loss: 0.1875 - val_accuracy: 0.9250
Epoch 61/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2569 - accuracy: 0.8867 -
val_loss: 0.1843 - val_accuracy: 0.9400
Epoch 62/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1293 - accuracy: 0.9400 -
val_loss: 0.1652 - val_accuracy: 0.9450
Epoch 63/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2493 - accuracy: 0.9000 -
val_loss: 0.3212 - val_accuracy: 0.8650
Epoch 64/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2203 - accuracy: 0.9267 -
val_loss: 0.1764 - val_accuracy: 0.9250
Epoch 65/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1403 - accuracy: 0.9333 -
val_loss: 0.1615 - val_accuracy: 0.9450
Epoch 66/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1491 - accuracy: 0.9400 -
val_loss: 0.1620 - val_accuracy: 0.9450
Epoch 67/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1914 - accuracy: 0.9200 -
val_loss: 0.1980 - val_accuracy: 0.9200
Epoch 68/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1827 - accuracy: 0.9200 -
val_loss: 0.1541 - val_accuracy: 0.9400
Epoch 69/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1744 - accuracy: 0.9333 -
val_loss: 0.1850 - val_accuracy: 0.9250
Epoch 70/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1089 - accuracy: 0.9400 -
val_loss: 0.1835 - val_accuracy: 0.9350
Epoch 71/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1210 - accuracy: 0.9333 -
val_loss: 0.1888 - val_accuracy: 0.9250
Epoch 72/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2308 - accuracy: 0.9128 -
val_loss: 0.1999 - val_accuracy: 0.9450
Epoch 73/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2062 - accuracy: 0.9333 -
val_loss: 0.2174 - val_accuracy: 0.9100
Epoch 74/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1012 - accuracy: 0.9800 -
val_loss: 0.1806 - val_accuracy: 0.9400
Epoch 75/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1185 - accuracy: 0.9467 -
val_loss: 0.1777 - val_accuracy: 0.9350
Epoch 76/100
50/50 [==============================] - 0s 8ms/step - loss: 0.2299 - accuracy: 0.9133 -
val_loss: 0.2198 - val_accuracy: 0.9150
Epoch 77/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1553 - accuracy: 0.9267 -

```
50/50 [==============================] - 0s 7ms/step - loss: 0.1553 - accuracy: 0.9267 -
val_loss: 0.1683 - val_accuracy: 0.9200
Epoch 78/100
50/50 [==============================] - ETA: 0s - loss: 0.1609 - accuracy: 0.93 - 0s 7ms
/step - loss: 0.1561 - accuracy: 0.9333 - val_loss: 0.1658 - val_accuracy: 0.9350
Epoch 79/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1184 - accuracy: 0.9533 -
val_loss: 0.1818 - val_accuracy: 0.9400
Epoch 80/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1654 - accuracy: 0.9400 -
val_loss: 0.1666 - val_accuracy: 0.9450
Epoch 81/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1767 - accuracy: 0.9267 -
val_loss: 0.1476 - val_accuracy: 0.9450
Epoch 82/100
50/50 [==============================] - 0s 8ms/step - loss: 0.1404 - accuracy: 0.9530 -
val_loss: 0.1732 - val_accuracy: 0.9450
Epoch 83/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1988 - accuracy: 0.9267 -
val_loss: 0.1779 - val_accuracy: 0.9400
Epoch 84/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2146 - accuracy: 0.9267 -
val_loss: 0.1934 - val_accuracy: 0.9300
Epoch 85/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1270 - accuracy: 0.9467 -
val_loss: 0.1909 - val_accuracy: 0.9100
Epoch 86/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1683 - accuracy: 0.9267 -
val_loss: 0.1652 - val_accuracy: 0.9350
Epoch 87/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1559 - accuracy: 0.9200 -
val_loss: 0.1694 - val_accuracy: 0.9400
Epoch 88/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1550 - accuracy: 0.9396 -
val_loss: 0.1604 - val_accuracy: 0.9450
Epoch 89/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1113 - accuracy: 0.9667 -
val_loss: 0.1885 - val_accuracy: 0.9350
Epoch 90/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1412 - accuracy: 0.9333 -
val_loss: 0.1644 - val_accuracy: 0.9500
Epoch 91/100
50/50 [==============================] - 0s 7ms/step - loss: 0.2082 - accuracy: 0.9400 -
val_loss: 0.2017 - val_accuracy: 0.9100
Epoch 92/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1795 - accuracy: 0.9400 -
val_loss: 0.1922 - val_accuracy: 0.9450
Epoch 93/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1306 - accuracy: 0.9333 -
val_loss: 0.1862 - val_accuracy: 0.9350
Epoch 94/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1430 - accuracy: 0.9467 -
val_loss: 0.1745 - val_accuracy: 0.9400
Epoch 95/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1101 - accuracy: 0.9600 -
val_loss: 0.1847 - val_accuracy: 0.9200
Epoch 96/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1277 - accuracy: 0.9533 -
val_loss: 0.1766 - val_accuracy: 0.9400
Epoch 97/100
50/50 [==============================] - 0s 6ms/step - loss: 0.2098 - accuracy: 0.9200 -
val_loss: 0.1640 - val_accuracy: 0.9500
Epoch 98/100
50/50 [==============================] - 0s 6ms/step - loss: 0.0891 - accuracy: 0.9800 -
val_loss: 0.1725 - val_accuracy: 0.9550
Epoch 99/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1767 - accuracy: 0.9467 -
val_loss: 0.1745 - val_accuracy: 0.9450
Epoch 100/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1387 - accuracy: 0.9467 -
val_loss: 0.1508 - val_accuracy: 0.9550
```

In [18]:

```
cnn.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                147520
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 130
=================================================================
Total params: 167,042
Trainable params: 167,042
Non-trainable params: 0
_________________________________________________________________
```

In [19]:

```
dosya = open("C:/Users/mehme/Datasets/Gender Classification Dataset/gender/deneme.txt","r
",encoding="utf-8")
dizi = dosya.read().split('\n')

#ConfusionMatrix degerleri
tp = 0
tn = 0
fp = 0
fn = 0


for i in range(1, 101):

    path = 'C:/Users/mehme/Datasets/Gender Classification Dataset/gender/deneme/' + '('+
str(i) + ')' + '.jpg'
    test_image = image.load_img(path , target_size=(32, 32))
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis = 0)
    result = cnn.predict(test_image)
    train_dataset.class_indices
    if result[0][0] == 1:
        prediction = 'Kadın'
    else:
        prediction = 'Erkek'
    print(str(i)+ ".Deger : " + prediction + " - Gercek Deger : " + dizi[i] )
    if str(prediction) == 'Kadın' and str(dizi[i]) == 'Kadın':
        tp = tp + 1
    if str(prediction) == 'Erkek' and str(dizi[i]) == 'Erkek':
        tn = tn + 1
    if str(prediction) == 'Kadın' and str(dizi[i]) == 'Kadın':
        fp = fp + 1
    if str(prediction) == 'Kadın' and str(dizi[i]) == 'Kadın':
        fn = fn + 1
print('----------------------------------------------------------------------------
-')
print("Confusion Matrix Sonuclari :")
print("")
print("TP : " + str(tp) + " TN : " + str(tn) + " FP : " + str(fp) + " FN : " + str(fn) )
print("Recall : " + str(tp/(tp+fn)))
print("Precision : " + str(tp/(tp+fp)))
print("Accuracy : " + str((tp+tn)/100))
print("F1 - Score : " + str((2*tp)/(2*tp+fp+fn)))
#tp-kadın kadın
```

```
#tn-erkek erkek
#fp-kadın erkek
#fn-erkek kadın
```

```
1.Deger : Erkek - Gercek Deger : Kadın
2.Deger : Kadın - Gercek Deger : Kadın
3.Deger : Erkek - Gercek Deger : Kadın
4.Deger : Erkek - Gercek Deger : Kadın
5.Deger : Kadın - Gercek Deger : Kadın
6.Deger : Kadın - Gercek Deger : Kadın
7.Deger : Kadın - Gercek Deger : Kadın
8.Deger : Erkek - Gercek Deger : Kadın
9.Deger : Kadın - Gercek Deger : Kadın
10.Deger : Kadın - Gercek Deger : Kadın
11.Deger : Kadın - Gercek Deger : Kadın
12.Deger : Kadın - Gercek Deger : Kadın
13.Deger : Kadın - Gercek Deger : Kadın
14.Deger : Kadın - Gercek Deger : Kadın
15.Deger : Erkek - Gercek Deger : Kadın
16.Deger : Kadın - Gercek Deger : Kadın
17.Deger : Erkek - Gercek Deger : Kadın
18.Deger : Erkek - Gercek Deger : Kadın
19.Deger : Erkek - Gercek Deger : Kadın
20.Deger : Erkek - Gercek Deger : Kadın
21.Deger : Kadın - Gercek Deger : Kadın
22.Deger : Kadın - Gercek Deger : Kadın
23.Deger : Kadın - Gercek Deger : Kadın
24.Deger : Erkek - Gercek Deger : Kadın
25.Deger : Erkek - Gercek Deger : Kadın
26.Deger : Kadın - Gercek Deger : Kadın
27.Deger : Kadın - Gercek Deger : Kadın
28.Deger : Kadın - Gercek Deger : Kadın
29.Deger : Kadın - Gercek Deger : Kadın
30.Deger : Kadın - Gercek Deger : Kadın
31.Deger : Erkek - Gercek Deger : Kadın
32.Deger : Erkek - Gercek Deger : Kadın
33.Deger : Kadın - Gercek Deger : Kadın
34.Deger : Erkek - Gercek Deger : Kadın
35.Deger : Erkek - Gercek Deger : Kadın
36.Deger : Kadın - Gercek Deger : Kadın
37.Deger : Kadın - Gercek Deger : Kadın
38.Deger : Kadın - Gercek Deger : Kadın
39.Deger : Erkek - Gercek Deger : Kadın
40.Deger : Kadın - Gercek Deger : Kadın
41.Deger : Kadın - Gercek Deger : Kadın
42.Deger : Kadın - Gercek Deger : Kadın
43.Deger : Kadın - Gercek Deger : Kadın
44.Deger : Erkek - Gercek Deger : Kadın
45.Deger : Kadın - Gercek Deger : Kadın
46.Deger : Erkek - Gercek Deger : Kadın
47.Deger : Kadın - Gercek Deger : Kadın
48.Deger : Kadın - Gercek Deger : Kadın
49.Deger : Kadın - Gercek Deger : Kadın
50.Deger : Kadın - Gercek Deger : Kadın
51.Deger : Erkek - Gercek Deger : Erkek
52.Deger : Erkek - Gercek Deger : Erkek
53.Deger : Erkek - Gercek Deger : Erkek
54.Deger : Erkek - Gercek Deger : Erkek
55.Deger : Erkek - Gercek Deger : Erkek
56.Deger : Erkek - Gercek Deger : Erkek
57.Deger : Erkek - Gercek Deger : Erkek
58.Deger : Erkek - Gercek Deger : Erkek
59.Deger : Erkek - Gercek Deger : Erkek
60.Deger : Erkek - Gercek Deger : Erkek
61.Deger : Erkek - Gercek Deger : Erkek
62.Deger : Erkek - Gercek Deger : Erkek
63.Deger : Erkek - Gercek Deger : Erkek
64.Deger : Erkek - Gercek Deger : Erkek
65.Deger : Erkek - Gercek Deger : Erkek
66.Deger : Erkek - Gercek Deger : Erkek
67.Deger : Erkek - Gercek Deger : Erkek
68.Deger : Erkek - Gercek Deger : Erkek
```

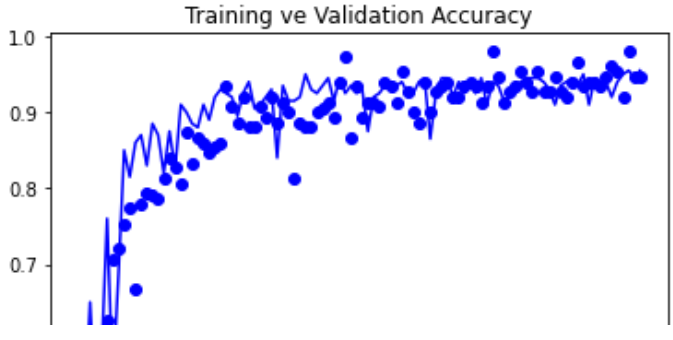
```
69.Deger : Erkek - Gercek Deger : Erkek
70.Deger : Erkek - Gercek Deger : Erkek
71.Deger : Erkek - Gercek Deger : Erkek
72.Deger : Erkek - Gercek Deger : Erkek
73.Deger : Erkek - Gercek Deger : Erkek
74.Deger : Erkek - Gercek Deger : Erkek
75.Deger : Erkek - Gercek Deger : Erkek
76.Deger : Erkek - Gercek Deger : Erkek
77.Deger : Erkek - Gercek Deger : Erkek
78.Deger : Erkek - Gercek Deger : Erkek
79.Deger : Kadın - Gercek Deger : Erkek
80.Deger : Erkek - Gercek Deger : Erkek
81.Deger : Erkek - Gercek Deger : Erkek
82.Deger : Erkek - Gercek Deger : Erkek
83.Deger : Erkek - Gercek Deger : Erkek
84.Deger : Erkek - Gercek Deger : Erkek
85.Deger : Erkek - Gercek Deger : Erkek
86.Deger : Erkek - Gercek Deger : Erkek
87.Deger : Erkek - Gercek Deger : Erkek
88.Deger : Erkek - Gercek Deger : Erkek
89.Deger : Erkek - Gercek Deger : Erkek
90.Deger : Erkek - Gercek Deger : Erkek
91.Deger : Erkek - Gercek Deger : Erkek
92.Deger : Erkek - Gercek Deger : Erkek
93.Deger : Erkek - Gercek Deger : Erkek
94.Deger : Erkek - Gercek Deger : Erkek
95.Deger : Erkek - Gercek Deger : Erkek
96.Deger : Erkek - Gercek Deger : Erkek
97.Deger : Erkek - Gercek Deger : Erkek
98.Deger : Erkek - Gercek Deger : Erkek
99.Deger : Erkek - Gercek Deger : Erkek
100.Deger : Erkek - Gercek Deger : Erkek
--------------------------------------------------------------------------------
Confusion Matrix Sonuclari :

TP : 32 TN : 49 FP : 32 FN : 32
Recall : 0.5
Precision : 0.5
Accuracy : 0.81
F1 - Score : 0.5
```

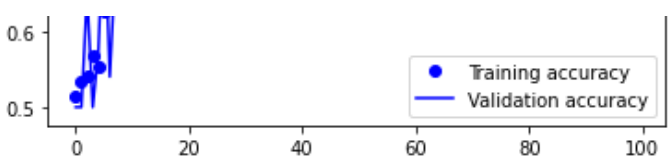
In [21]:

```
import matplotlib.pyplot as plt
%matplotlib inline

accuracy = history1.history['accuracy']
val_accuracy = history1.history['val_accuracy']
loss = history1.history['loss']
val_loss = history1.history['val_loss']
epochs = range(len(accuracy))

plt.plot(epochs, accuracy, 'bo', label='Training accuracy')
plt.plot(epochs, val_accuracy, 'b', label='Validation accuracy')
plt.title('Training ve Validation Accuracy')
plt.legend()
plt.figure()
```

Out[21]:

```
<Figure size 432x288 with 0 Axes>
```






```
<Figure size 432x288 with 0 Axes>
```

In [ ]: