

In [15]:

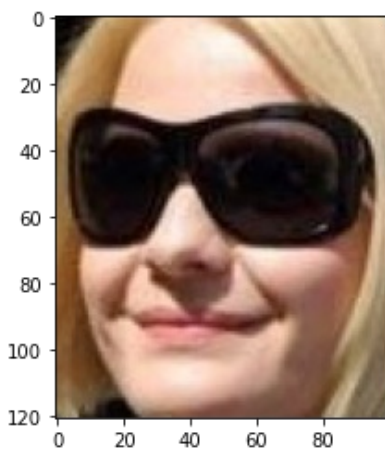
```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers 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 [16]:

```
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 [17]:

```
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 [18]:

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

Out[18]:

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

In [19]:

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

Out[19]:

```
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, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1])
```

## Train ve validation yapısında bulunan sınıf(class) yapıları

In [21]:

```
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(32, (3,3), activation="relu"),
    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 [22]:

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

In [23]:

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

)
```

Epoch 1/100

50/50 [=====] - 0s 9ms/step - loss: 0.7071 - accuracy: 0.4733 - val\_loss: 0.6927 - val\_accuracy: 0.5050

Epoch 2/100

50/50 [=====] - 0s 6ms/step - loss: 0.6931 - accuracy: 0.5733 - val\_loss: 0.6927 - val\_accuracy: 0.5000

Epoch 3/100

50/50 [=====] - 0s 6ms/step - loss: 0.6933 - accuracy: 0.4933 - val loss: 0.6887 - val accuracv: 0.5000

```
Epoch 4/100
50/50 [=====] - 0s 6ms/step - loss: 0.6955 - accuracy: 0.5267 -
val_loss: 0.6876 - val_accuracy: 0.5500
Epoch 5/100
50/50 [=====] - 0s 6ms/step - loss: 0.6891 - accuracy: 0.5733 -
val_loss: 0.6812 - val_accuracy: 0.5500
Epoch 6/100
50/50 [=====] - 0s 6ms/step - loss: 0.6706 - accuracy: 0.6267 -
val_loss: 0.6424 - val_accuracy: 0.6050
Epoch 7/100
50/50 [=====] - 0s 6ms/step - loss: 0.6488 - accuracy: 0.6667 -
val_loss: 0.6143 - val_accuracy: 0.7050
Epoch 8/100
50/50 [=====] - 0s 6ms/step - loss: 0.6216 - accuracy: 0.6400 -
val_loss: 0.6404 - val_accuracy: 0.6200
Epoch 9/100
50/50 [=====] - 0s 6ms/step - loss: 0.6328 - accuracy: 0.6333 -
val_loss: 0.5852 - val_accuracy: 0.7000
Epoch 10/100
50/50 [=====] - 0s 6ms/step - loss: 0.6324 - accuracy: 0.6467 -
val_loss: 0.5947 - val_accuracy: 0.7050
Epoch 11/100
50/50 [=====] - 0s 6ms/step - loss: 0.6015 - accuracy: 0.6733 -
val_loss: 0.5694 - val_accuracy: 0.7400
Epoch 12/100
50/50 [=====] - 0s 6ms/step - loss: 0.6202 - accuracy: 0.6800 -
val_loss: 0.5411 - val_accuracy: 0.7150
Epoch 13/100
50/50 [=====] - 0s 6ms/step - loss: 0.5887 - accuracy: 0.7067 -
val_loss: 0.5037 - val_accuracy: 0.8050
Epoch 14/100
50/50 [=====] - 0s 6ms/step - loss: 0.4810 - accuracy: 0.8133 -
val_loss: 0.4937 - val_accuracy: 0.7900
Epoch 15/100
50/50 [=====] - 0s 7ms/step - loss: 0.5270 - accuracy: 0.7400 -
val_loss: 0.4007 - val_accuracy: 0.8500
Epoch 16/100
50/50 [=====] - 0s 6ms/step - loss: 0.4577 - accuracy: 0.7933 -
val_loss: 0.3434 - val_accuracy: 0.8900
Epoch 17/100
50/50 [=====] - 0s 6ms/step - loss: 0.4381 - accuracy: 0.8267 -
val_loss: 0.4032 - val_accuracy: 0.8750
Epoch 18/100
50/50 [=====] - 0s 6ms/step - loss: 0.4180 - accuracy: 0.8000 -
val_loss: 0.3084 - val_accuracy: 0.8650
Epoch 19/100
50/50 [=====] - 0s 6ms/step - loss: 0.3941 - accuracy: 0.8400 -
val_loss: 0.3485 - val_accuracy: 0.8450
Epoch 20/100
50/50 [=====] - 0s 6ms/step - loss: 0.3202 - accuracy: 0.8667 -
val_loss: 0.2914 - val_accuracy: 0.8650
Epoch 21/100
50/50 [=====] - 0s 6ms/step - loss: 0.3541 - accuracy: 0.8467 -
val_loss: 0.3097 - val_accuracy: 0.8750
Epoch 22/100
50/50 [=====] - 0s 6ms/step - loss: 0.4231 - accuracy: 0.7987 -
val_loss: 0.3456 - val_accuracy: 0.8700
Epoch 23/100
50/50 [=====] - 0s 7ms/step - loss: 0.3573 - accuracy: 0.8733 -
val_loss: 0.2978 - val_accuracy: 0.8750
Epoch 24/100
50/50 [=====] - 0s 7ms/step - loss: 0.2990 - accuracy: 0.8933 -
val_loss: 0.2772 - val_accuracy: 0.9050
Epoch 25/100
50/50 [=====] - 0s 7ms/step - loss: 0.3348 - accuracy: 0.8400 -
val_loss: 0.2852 - val_accuracy: 0.9200
Epoch 26/100
50/50 [=====] - 0s 6ms/step - loss: 0.1918 - accuracy: 0.9267 -
val_loss: 0.2982 - val_accuracy: 0.8850
Epoch 27/100
50/50 [=====] - 0s 6ms/step - loss: 0.4389 - accuracy: 0.8333 -
val_loss: 0.2825 - val_accuracy: 0.9050
```

Epoch 28/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2622 - accuracy: 0.9000 -  
val\_loss: 0.2539 - val\_accuracy: 0.9150  
Epoch 29/100  
50/50 [=====] - 0s 6ms/step - loss: 0.3157 - accuracy: 0.8600 -  
val\_loss: 0.2320 - val\_accuracy: 0.9200  
Epoch 30/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2993 - accuracy: 0.8933 -  
val\_loss: 0.3192 - val\_accuracy: 0.8700  
Epoch 31/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2641 - accuracy: 0.9067 -  
val\_loss: 0.2331 - val\_accuracy: 0.9250  
Epoch 32/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2378 - accuracy: 0.9267 -  
val\_loss: 0.3416 - val\_accuracy: 0.8950  
Epoch 33/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2837 - accuracy: 0.9000 -  
val\_loss: 0.2804 - val\_accuracy: 0.8750  
Epoch 34/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2018 - accuracy: 0.9133 -  
val\_loss: 0.2556 - val\_accuracy: 0.9050  
Epoch 35/100  
50/50 [=====] - 0s 7ms/step - loss: 0.3097 - accuracy: 0.8533 -  
val\_loss: 0.2667 - val\_accuracy: 0.9050  
Epoch 36/100  
50/50 [=====] - 0s 7ms/step - loss: 0.3561 - accuracy: 0.8600 -  
val\_loss: 0.2329 - val\_accuracy: 0.9200  
Epoch 37/100  
50/50 [=====] - 0s 8ms/step - loss: 0.3215 - accuracy: 0.8667 -  
val\_loss: 0.2159 - val\_accuracy: 0.9250  
Epoch 38/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2398 - accuracy: 0.9267 -  
val\_loss: 0.1936 - val\_accuracy: 0.9100  
Epoch 39/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2630 - accuracy: 0.8933 -  
val\_loss: 0.1935 - val\_accuracy: 0.9400  
Epoch 40/100  
50/50 [=====] - 0s 6ms/step - loss: 0.3578 - accuracy: 0.8867 -  
val\_loss: 0.2559 - val\_accuracy: 0.9100  
Epoch 41/100  
50/50 [=====] - 0s 6ms/step - loss: 0.3499 - accuracy: 0.8400 -  
val\_loss: 0.2068 - val\_accuracy: 0.9350  
Epoch 42/100  
50/50 [=====] - 0s 6ms/step - loss: 0.3423 - accuracy: 0.8600 -  
val\_loss: 0.2124 - val\_accuracy: 0.9350  
Epoch 43/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2969 - accuracy: 0.8667 -  
val\_loss: 0.2047 - val\_accuracy: 0.9150  
Epoch 44/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2706 - accuracy: 0.8933 -  
val\_loss: 0.2162 - val\_accuracy: 0.9200  
Epoch 45/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2147 - accuracy: 0.9067 -  
val\_loss: 0.2167 - val\_accuracy: 0.9250  
Epoch 46/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2520 - accuracy: 0.8867 -  
val\_loss: 0.1924 - val\_accuracy: 0.9300  
Epoch 47/100  
50/50 [=====] - 0s 8ms/step - loss: 0.1691 - accuracy: 0.9467 -  
val\_loss: 0.2152 - val\_accuracy: 0.9050  
Epoch 48/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2136 - accuracy: 0.9067 -  
val\_loss: 0.2854 - val\_accuracy: 0.8850  
Epoch 49/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2768 - accuracy: 0.8933 -  
val\_loss: 0.1729 - val\_accuracy: 0.9350  
Epoch 50/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2674 - accuracy: 0.8933 -  
val\_loss: 0.2466 - val\_accuracy: 0.9150  
Epoch 51/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2248 - accuracy: 0.9133 -  
val\_loss: 0.3672 - val\_accuracy: 0.8200

Epoch 52/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1646 - accuracy: 0.9400 -  
val\_loss: 0.2780 - val\_accuracy: 0.9050  
Epoch 53/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1971 - accuracy: 0.8933 -  
val\_loss: 0.2138 - val\_accuracy: 0.9150  
Epoch 54/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2508 - accuracy: 0.9000 -  
val\_loss: 0.1724 - val\_accuracy: 0.9350  
Epoch 55/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2028 - accuracy: 0.9200 -  
val\_loss: 0.1989 - val\_accuracy: 0.9200  
Epoch 56/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2503 - accuracy: 0.8867 -  
val\_loss: 0.2488 - val\_accuracy: 0.8900  
Epoch 57/100  
50/50 [=====] - 0s 6ms/step - loss: 0.3246 - accuracy: 0.8667 -  
val\_loss: 0.2119 - val\_accuracy: 0.9150  
Epoch 58/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2037 - accuracy: 0.9200 -  
val\_loss: 0.2719 - val\_accuracy: 0.8800  
Epoch 59/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2279 - accuracy: 0.9200 -  
val\_loss: 0.1987 - val\_accuracy: 0.9300  
Epoch 60/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2891 - accuracy: 0.8933 -  
val\_loss: 0.2220 - val\_accuracy: 0.9300  
Epoch 61/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2466 - accuracy: 0.8867 -  
val\_loss: 0.1544 - val\_accuracy: 0.9400  
Epoch 62/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2336 - accuracy: 0.9128 -  
val\_loss: 0.2037 - val\_accuracy: 0.9100  
Epoch 63/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1372 - accuracy: 0.9600 -  
val\_loss: 0.1853 - val\_accuracy: 0.9400  
Epoch 64/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2707 - accuracy: 0.9000 -  
val\_loss: 0.2747 - val\_accuracy: 0.8800  
Epoch 65/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2166 - accuracy: 0.9200 -  
val\_loss: 0.1706 - val\_accuracy: 0.9400  
Epoch 66/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2125 - accuracy: 0.9200 -  
val\_loss: 0.1804 - val\_accuracy: 0.9400  
Epoch 67/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2207 - accuracy: 0.9267 -  
val\_loss: 0.1810 - val\_accuracy: 0.9450  
Epoch 68/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2243 - accuracy: 0.9333 -  
val\_loss: 0.1702 - val\_accuracy: 0.9400  
Epoch 69/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2676 - accuracy: 0.8933 -  
val\_loss: 0.3015 - val\_accuracy: 0.8900  
Epoch 70/100  
50/50 [=====] - 0s 6ms/step - loss: 0.3188 - accuracy: 0.8533 -  
val\_loss: 0.2053 - val\_accuracy: 0.9350  
Epoch 71/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2326 - accuracy: 0.9200 -  
val\_loss: 0.2230 - val\_accuracy: 0.9350  
Epoch 72/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2012 - accuracy: 0.9267 -  
val\_loss: 0.2337 - val\_accuracy: 0.9100  
Epoch 73/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2440 - accuracy: 0.9067 -  
val\_loss: 0.1974 - val\_accuracy: 0.9250  
Epoch 74/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2934 - accuracy: 0.8800 -  
val\_loss: 0.1997 - val\_accuracy: 0.9350  
Epoch 75/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2148 - accuracy: 0.8933 -  
val\_loss: 0.1958 - val\_accuracy: 0.9300

Epoch 76/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1991 - accuracy: 0.9262 -  
val\_loss: 0.1941 - val\_accuracy: 0.9400  
Epoch 77/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2768 - accuracy: 0.9133 -  
val\_loss: 0.1881 - val\_accuracy: 0.9250  
Epoch 78/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1896 - accuracy: 0.9267 -  
val\_loss: 0.1783 - val\_accuracy: 0.9350  
Epoch 79/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2532 - accuracy: 0.8933 -  
val\_loss: 0.1926 - val\_accuracy: 0.9200  
Epoch 80/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1639 - accuracy: 0.9333 -  
val\_loss: 0.1948 - val\_accuracy: 0.9500  
Epoch 81/100  
50/50 [=====] - 0s 8ms/step - loss: 0.2021 - accuracy: 0.8933 -  
val\_loss: 0.1901 - val\_accuracy: 0.9450  
Epoch 82/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1932 - accuracy: 0.9333 -  
val\_loss: 0.1875 - val\_accuracy: 0.9350  
Epoch 83/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1992 - accuracy: 0.9133 -  
val\_loss: 0.1944 - val\_accuracy: 0.9300  
Epoch 84/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2213 - accuracy: 0.9200 -  
val\_loss: 0.1741 - val\_accuracy: 0.9400  
Epoch 85/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1868 - accuracy: 0.9400 -  
val\_loss: 0.1754 - val\_accuracy: 0.9450  
Epoch 86/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1431 - accuracy: 0.9600 -  
val\_loss: 0.2359 - val\_accuracy: 0.9350  
Epoch 87/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2463 - accuracy: 0.9000 -  
val\_loss: 0.1826 - val\_accuracy: 0.9250  
Epoch 88/100  
50/50 [=====] - 0s 8ms/step - loss: 0.2374 - accuracy: 0.8933 -  
val\_loss: 0.1931 - val\_accuracy: 0.9200  
Epoch 89/100  
50/50 [=====] - 0s 7ms/step - loss: 0.1829 - accuracy: 0.9128 -  
val\_loss: 0.2130 - val\_accuracy: 0.9350  
Epoch 90/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1276 - accuracy: 0.9600 -  
val\_loss: 0.2110 - val\_accuracy: 0.9450  
Epoch 91/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1910 - accuracy: 0.9133 -  
val\_loss: 0.3173 - val\_accuracy: 0.8550  
Epoch 92/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1899 - accuracy: 0.9267 -  
val\_loss: 0.1518 - val\_accuracy: 0.9650  
Epoch 93/100  
50/50 [=====] - 0s 6ms/step - loss: 0.2011 - accuracy: 0.9400 -  
val\_loss: 0.1836 - val\_accuracy: 0.9400  
Epoch 94/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1550 - accuracy: 0.9467 -  
val\_loss: 0.1923 - val\_accuracy: 0.9200  
Epoch 95/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1833 - accuracy: 0.9267 -  
val\_loss: 0.1662 - val\_accuracy: 0.9300  
Epoch 96/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2043 - accuracy: 0.9333 -  
val\_loss: 0.1767 - val\_accuracy: 0.9450  
Epoch 97/100  
50/50 [=====] - 0s 8ms/step - loss: 0.1305 - accuracy: 0.9533 -  
val\_loss: 0.2410 - val\_accuracy: 0.9150  
Epoch 98/100  
50/50 [=====] - 0s 6ms/step - loss: 0.1909 - accuracy: 0.9267 -  
val\_loss: 0.1496 - val\_accuracy: 0.9500  
Epoch 99/100  
50/50 [=====] - 0s 7ms/step - loss: 0.2506 - accuracy: 0.9267 -  
val\_loss: 0.1586 - val\_accuracy: 0.9350

```
Epoch 100/100
50/50 [=====] - 0s 8ms/step - loss: 0.2227 - accuracy: 0.8993 -
val_loss: 0.2134 - val_accuracy: 0.9150
```

In [20]:

```
cnn.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 6, 6, 32)	0
conv2d_2 (Conv2D)	(None, 4, 4, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 64)	16448
dense_1 (Dense)	(None, 2)	130
=====		
Total params: 45,218		
Trainable params: 45,218		
Non-trainable params: 0		

In [56]:

```
y_pred = cnn.predict(test_dataset)
y_pred[:5]
```

Out[56]:

```
array([[0.04510095, 0.9548991 ],
       [0.05023478, 0.94976526],
       [0.85365826, 0.14634176],
       [0.9680272 , 0.03197284],
       [0.992386 , 0.007614  ]], dtype=float32)
```

In [98]:

```
y_classes = [np.argmax(i) for i in y_pred]
y_classes[:10]
```

Out[98]:

```
[1, 1, 0, 0, 0, 0, 0, 1, 0, 0]
```

In [159]:

```
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 : Erkek - Gercek Deger : Kadın
10.Deger : Erkek - Gercek Deger : Kadın
11.Deger : Kadın - Gercek Deger : Kadın
12.Deger : Erkek - 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 : Kadın - 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 : Erkek - Gercek Deger : Kadın
27.Deger : Erkek - 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 : Erkek - Gercek Deger : Kadın
37.Deger : Kadın - Gercek Deger : Kadın
38.Deger : Kadın - Gercek Deger : Kadın

```



39.Deger : Kadın - Gercek Deger : Kadın  
40.Deger : Erkek - Gercek Deger : Kadın  
41.Deger : Erkek - Gercek Deger : Kadın  
42.Deger : Kadın - Gercek Deger : Kadın  
43.Deger : Erkek - 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 : Erkek - Gercek Deger : Kadın  
51.Deger : Erkek - Gercek Deger : Erkek  
52.Deger : Erkek - Gercek Deger : Erkek  
53.Deger : Kadın - 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 : Erkek - 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 : 24 TN : 49 FP : 24 FN : 24

Recall : 0.5

Precision : 0.5

Accuracy : 0.73

F1 - Score : 0.5

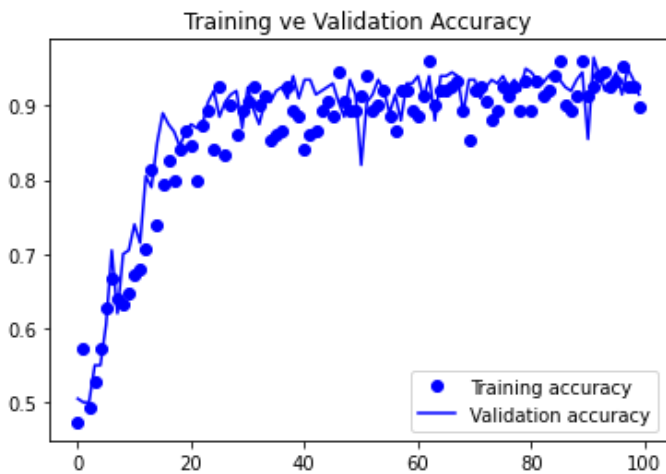
```
import matplotlib.pyplot as plt
%matplotlib inline

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[162]:

<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>

In [ ]: