In [ ]:
```
### Famous Iconic Women
```

Disclamer These images were collected using icrawler. If you find any damaged/bad images, please start a discussion with a list of the images' names so we can clean the dataset.

Context I created this dataset for the #CourageToCreate Initiative by Google Women Techmakers for International Women's Day 2021.

Content This dataset contains images of 64 Iconic Women in History. This images were scraped using the icrawler python package. Each women has their own sub-folder in the data folder, with the name of the iconic woman as the name of the sub-folder.

In [2]:
```
###!mkdir ~/.kaggle
```

In [4]:
```
##!cp /kaggle.json ~/.kaggle/
```

In [6]:
```
####!chmod 600 ~/.kaggle/kaggle.json
```

In [ ]:
```
####! pip install kaggle
```

In [ ]:
```
####!kaggle datasets download -d fatiimaezzahra/famous-iconic-women
```

In [ ]:
```
####! unzip /content/famous-iconic-women.zip
```

In [10]:
```
import tensorflow as tf
from tensorflow import keras
import numpy as np
```

In [11]:
```
print(tf.__version__)
```

```
2.7.0
```

In [12]:
```
# import the libraries as shown below

from tensorflow.keras.layers import Input, Lambda, Dense, Flatten,Conv2D
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
```

In [13]:
```
from PIL import ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True
```

In [14]:
```
# re-size all the images to this
IMAGE_SIZE = [224, 224]

train_path = '/content/output/train'
valid_path = '/content/output/valid'
```

In [15]:
```
# Import the Vgg 16 library as shown below and add preprocessing layer to the front of
# Here we will be using imagenet weights

mobilenetv2 = MobileNetV2(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobi
lenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5
9412608/9406464 [==============================] - 0s 0us/step
9420800/9406464 [==============================] - 0s 0us/step
```

In [16]:
```
# don't train existing weights
for layer in mobilenetv2.layers:
    layer.trainable = False
```

In [17]:
```
  # useful for getting number of output classes
folders = glob('/content/output/train/*')
```

In [18]:
```
folders
```

Out[18]:
```
['/content/output/train/Jacinda Ardern',
 '/content/output/train/Judy Garland',
 '/content/output/train/Susan Sontag',
```

```
'/content/output/train/Betty White',
'/content/output/train/Nina Simone',
'/content/output/train/Meryl Streep',
'/content/output/train/Bette Davis',
'/content/output/train/Simone de Beauvoir',
'/content/output/train/Virginia Woolf',
'/content/output/train/Sirimavo Bandaranaike',
'/content/output/train/Cher',
'/content/output/train/Estee Lauder',
'/content/output/train/Sonja Henie',
'/content/output/train/Serena Williams',
'/content/output/train/Marilyn Monroe',
'/content/output/train/Madonna',
'/content/output/train/Rachael Heyhoe Flint',
'/content/output/train/Celine Dion',
'/content/output/train/Ruth Handler',
'/content/output/train/Grace Hopper',
'/content/output/train/Lise Meitner',
'/content/output/train/Suzanne Lenglen',
'/content/output/train/Katia Krafft',
'/content/output/train/Angela Merkel',
'/content/output/train/Sheryl Sandberg',
'/content/output/train/Margaret Thatcher',
'/content/output/train/Amelia Earhart',
'/content/output/train/Wangari Maathai',
'/content/output/train/Coco Chanel',
'/content/output/train/Mary Wollstonecraft',
'/content/output/train/Marie Van Brittan Brown',
'/content/output/train/Frida Kahlo',
'/content/output/train/Junko Tabei',
'/content/output/train/Diana, Princess of Wales',
'/content/output/train/Bessie Coleman',
'/content/output/train/Zora Neale Hurston',
'/content/output/train/Wilma Rudolph',
'/content/output/train/Maryam Mirzakhani',
'/content/output/train/Andrea Dworkin',
'/content/output/train/Kamala Harris',
'/content/output/train/Florence Nightingale',
```

```
 '/content/output/train/Gertrude Ederle',
 '/content/output/train/Oprah Winfrey',
 '/content/output/train/Marie Curie',
 '/content/output/train/Katharine Graham',
 '/content/output/train/Marie Stopes',
 '/content/output/train/Alice Milliat',
 '/content/output/train/Whitney Houston',
 '/content/output/train/Anna Akhmatova',
 '/content/output/train/Ada Lovelace',
 '/content/output/train/Emmeline Pankhurst',
 '/content/output/train/Vera Atkins',
 '/content/output/train/Katharine Hepburn',
 '/content/output/train/Rihanna',
 '/content/output/train/Buchi Emecheta',
 '/content/output/train/Joan Robinson',
 '/content/output/train/Raja Easa Al Gurg',
 '/content/output/train/Audrey Hepburn',
 '/content/output/train/Diana Ross',
 '/content/output/train/Angela Burdett-Coutts',
 '/content/output/train/Aretha Franklin',
 '/content/output/train/Nancy Pelosi',
 '/content/output/train/Amrita Priam',
```

In [19]:

```
import os
print(os.listdir('/content/output/valid'))
```

```
['Jacinda Ardern', 'Judy Garland', 'Susan Sontag', 'Betty White', 'Nina Simone', 'Meryl
Streep', 'Bette Davis', 'Simone de Beauvoir', 'Virginia Woolf', 'Sirimavo Bandaranaike
', 'Cher', 'Estee Lauder', 'Sonja Henie', 'Serena Williams', 'Marilyn Monroe', 'Madonna
', 'Rachael Heyhoe Flint', 'Celine Dion', 'Ruth Handler', 'Grace Hopper', 'Lise Meitner
', 'Suzanne Lenglen', 'Katia Krafft', 'Angela Merkel', 'Sheryl Sandberg', 'Margaret Tha
tcher', 'Amelia Earhart', 'Wangari Maathai', 'Coco Chanel', 'Mary Wollstonecraft', 'Mar
ie Van Brittan Brown', 'Frida Kahlo', 'Junko Tabei', 'Diana, Princess of Wales', 'Bessi
e Coleman', 'Zora Neale Hurston', 'Wilma Rudolph', 'Maryam Mirzakhani', 'Andrea Dworkin
', 'Kamala Harris', 'Florence Nightingale', 'Gertrude Ederle', 'Oprah Winfrey', 'Marie
Curie', 'Katharine Graham', 'Marie Stopes', 'Alice Milliat', 'Whitney Houston', 'Anna A
```

```
khmatova', 'Ada Lovelace', 'Emmeline Pankhurst', 'Vera Atkins', 'Katharine Hepburn', 'R
ihanna', 'Buchi Emecheta', 'Joan Robinson', 'Raja Easa Al Gurg', 'Audrey Hepburn', 'Dia
na Ross', 'Angela Burdett-Coutts', 'Aretha Franklin', 'Nancy Pelosi', 'Amrita Priam', '
Rosa Parks']
```

In [20]:

```
lst = os.listdir('/content/output/valid')
lst.sort()
print(lst)
```

```
['Ada Lovelace', 'Alice Milliat', 'Amelia Earhart', 'Amrita Priam', 'Andrea Dworkin', '
Angela Burdett-Coutts', 'Angela Merkel', 'Anna Akhmatova', 'Aretha Franklin', 'Audrey H
epburn', 'Bessie Coleman', 'Bette Davis', 'Betty White', 'Buchi Emecheta', 'Celine Dion
', 'Cher', 'Coco Chanel', 'Diana Ross', 'Diana, Princess of Wales', 'Emmeline Pankhurst
', 'Estee Lauder', 'Florence Nightingale', 'Frida Kahlo', 'Gertrude Ederle', 'Grace Hop
per', 'Jacinda Ardern', 'Joan Robinson', 'Judy Garland', 'Junko Tabei', 'Kamala Harris
', 'Katharine Graham', 'Katharine Hepburn', 'Katia Krafft', 'Lise Meitner', 'Madonna',
'Margaret Thatcher', 'Marie Curie', 'Marie Stopes', 'Marie Van Brittan Brown', 'Marilyn
Monroe', 'Mary Wollstonecraft', 'Maryam Mirzakhani', 'Meryl Streep', 'Nancy Pelosi', 'N
ina Simone', 'Oprah Winfrey', 'Rachael Heyhoe Flint', 'Raja Easa Al Gurg', 'Rihanna', '
Rosa Parks', 'Ruth Handler', 'Serena Williams', 'Sheryl Sandberg', 'Simone de Beauvoir
', 'Sirimavo Bandaranaike', 'Sonja Henie', 'Susan Sontag', 'Suzanne Lenglen', 'Vera Atk
ins', 'Virginia Woolf', 'Wangari Maathai', 'Whitney Houston', 'Wilma Rudolph', 'Zora Ne
ale Hurston']
```

In [21]:

```
# our layers - you can add more if you want
x = Flatten()(mobilenetv2.output)
```

In [22]:

```
prediction = Dense(len(folders), activation='softmax')(x)

# create a model object
model = Model(inputs=mobilenetv2.input, outputs=prediction)
```

In [23]:

```
# view the structure of the model
model.summary()
```

```
Model: "model"
_________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
=========================================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0           []

 Conv1 (Conv2D)              (None, 112, 112, 32)      864         ['input_1[0][0]']

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32)   128         ['Conv1[0][0]']

 Conv1_relu (ReLU)           (None, 112, 112, 32)      0           ['bn_Conv1[0][0]']

 expanded_conv_depthwise (DepthwiseConv2D)  (None, 112, 112, 32)  288  ['Conv1_relu[0][0]']

 expanded_conv_depthwise_BN (BatchNormalization)  (None, 112, 112, 32)  128  ['expanded_conv_depthwise[0][0]']

 expanded_conv_depthwise_relu (ReLU)  (None, 112, 112, 32)  0  ['expanded_conv_depthwise_BN[0][0]']

 expanded_conv_project (Conv2D)  (None, 112, 112, 16)  512  ['expanded_conv_depthwise_relu[0][0]']
```

```
 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64          ['expanded_conv_projec
t[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536        ['expanded_conv_projec
t_BN[0][0]'
                                )                                 ]

 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384         ['block_1_expand
[0][0]']
 ization)                       )

 block_1_expand_relu (ReLU)     (None, 112, 112, 96  0           ['block_1_expand_BN
[0][0]']
                                )

 block_1_pad (ZeroPadding2D)    (None, 113, 113, 96  0           ['block_1_expand_relu
[0][0]']
                                )

 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)  864          ['block_1_pad[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)  384          ['block_1_depthwise
[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0           ['block_1_depthwise_BN
[0][0]']

 block_1_project (Conv2D)       (None, 56, 56, 24)   2304        ['block_1_depthwise_re
lu[0][0]']

 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96          ['block_1_project
[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456        ['block_1_project_BN
```

```
[0][0]']

 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576         ['block_2_expand
[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0           ['block_2_expand_BN
[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296       ['block_2_expand_relu
[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576        ['block_2_depthwise
[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0           ['block_2_depthwise_BN
[0][0]']

 block_2_project (Conv2D)       (None, 56, 56, 24)   3456        ['block_2_depthwise_re
lu[0][0]']

 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96          ['block_2_project
[0][0]']
 lization)

 block_2_add (Add)              (None, 56, 56, 24)   0           ['block_1_project_BN
[0][0]',
                                                                  'block_2_project_BN
[0][0]']

 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456        ['block_2_add[0][0]']

 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576        ['block_3_expand
[0][0]']
 ization)
```

```
 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0           ['block_3_expand_BN
[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0           ['block_3_expand_relu
[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296       ['block_3_pad[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576        ['block_3_depthwise
[0][0]']
 malization)

 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0           ['block_3_depthwise_BN
[0][0]']

 block_3_project (Conv2D)       (None, 28, 28, 32)   4608        ['block_3_depthwise_re
lu[0][0]']

 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_3_project
[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_3_project_BN
[0][0]']

 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768        ['block_4_expand
[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_4_expand_BN
[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728       ['block_4_expand_relu
[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768        ['block_4_depthwise
```

```
[0][0]']
 malization)

 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_4_depthwise_BN
[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_4_depthwise_re
lu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_4_project
[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0           ['block_3_project_BN
[0][0]',
                                                                  'block_4_project_BN
[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_4_add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768        ['block_5_expand
[0][0]']
 ization)

 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_5_expand_BN
[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728       ['block_5_expand_relu
[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768        ['block_5_depthwise
[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_5_depthwise_BN
[0][0]']
```

```
 block_5_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_5_depthwise_re
lu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_5_project
[0][0]']
 lization)

 block_5_add (Add)              (None, 28, 28, 32)   0           ['block_4_add[0][0]',
                                                                  'block_5_project_BN
[0][0]']

 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_5_add[0][0]']

 block_6_expand_BN (BatchNormal  (None, 28, 28, 192)  768        ['block_6_expand
[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_6_expand_BN
[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 29, 29, 192)  0           ['block_6_expand_relu
[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 14, 14, 192)  1728       ['block_6_pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 14, 14, 192)  768        ['block_6_depthwise
[0][0]']
 malization)

 block_6_depthwise_relu (ReLU)  (None, 14, 14, 192)  0           ['block_6_depthwise_BN
[0][0]']

 block_6_project (Conv2D)       (None, 14, 14, 64)   12288       ['block_6_depthwise_re
lu[0][0]']

 block_6_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_6_project
[0][0]']
```

```
 lization)

 block_7_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_6_project_BN
[0][0]']

 block_7_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_7_expand
[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_7_expand_BN
[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_7_expand_relu
[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_7_depthwise
[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_7_depthwise_BN
[0][0]']

 block_7_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_7_depthwise_re
lu[0][0]']

 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_7_project
[0][0]']
 lization)

 block_7_add (Add)              (None, 14, 14, 64)   0           ['block_6_project_BN
[0][0]',
                                                                  'block_7_project_BN
[0][0]']

 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_7_add[0][0]']

 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_8_expand
```

```
[0][0]']
 ization)

 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_8_expand_BN
[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_8_expand_relu
[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_8_depthwise
[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_8_depthwise_BN
[0][0]']

 block_8_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_8_depthwise_re
lu[0][0]']

 block_8_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_8_project
[0][0]']
 lization)

 block_8_add (Add)              (None, 14, 14, 64)   0           ['block_7_add[0][0]',
                                                                  'block_8_project_BN
[0][0]']

 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_8_add[0][0]']

 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_9_expand
[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_9_expand_BN
[0][0]']

 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_9_expand_relu
```

```
[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_9_depthwise
[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0          ['block_9_depthwise_BN
[0][0]']

 block_9_project (Conv2D)       (None, 14, 14, 64)   24576      ['block_9_depthwise_re
lu[0][0]']

 block_9_project_BN (BatchNorma  (None, 14, 14, 64)  256        ['block_9_project
[0][0]']
 lization)

 block_9_add (Add)              (None, 14, 14, 64)   0          ['block_8_add[0][0]',
                                                                 'block_9_project_BN
[0][0]']

 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576      ['block_9_add[0][0]']

 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536      ['block_10_expand
[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0          ['block_10_expand_BN
[0][0]']

 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456      ['block_10_expand_relu
[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536      ['block_10_depthwise
[0][0]']
 rmalization)
```

```
 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_10_depthwise_B
N[0][0]']

 block_10_project (Conv2D)      (None, 14, 14, 96)   36864       ['block_10_depthwise_r
elu[0][0]']

 block_10_project_BN (BatchNorm  (None, 14, 14, 96)  384         ['block_10_project
[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296       ['block_10_project_BN
[0][0]']

 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304       ['block_11_expand
[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0           ['block_11_expand_BN
[0][0]']

 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184       ['block_11_expand_relu
[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304       ['block_11_depthwise
[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0          ['block_11_depthwise_B
N[0][0]']

 block_11_project (Conv2D)      (None, 14, 14, 96)   55296       ['block_11_depthwise_r
elu[0][0]']

 block_11_project_BN (BatchNorm  (None, 14, 14, 96)  384         ['block_11_project
[0][0]']
 alization)
```

```
 block_11_add (Add)             (None, 14, 14, 96)   0           ['block_10_project_BN
[0][0]',
                                                                  'block_11_project_BN
[0][0]']

 block_12_expand (Conv2D)       (None, 14, 14, 576)  55296       ['block_11_add[0][0]']

 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304       ['block_12_expand
[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0           ['block_12_expand_BN
[0][0]']

 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184       ['block_12_expand_relu
[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304       ['block_12_depthwise
[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0          ['block_12_depthwise_B
N[0][0]']

 block_12_project (Conv2D)      (None, 14, 14, 96)   55296       ['block_12_depthwise_r
elu[0][0]']

 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384         ['block_12_project
[0][0]']
 alization)

 block_12_add (Add)             (None, 14, 14, 96)   0           ['block_11_add[0][0]',
                                                                  'block_12_project_BN
[0][0]']

 block_13_expand (Conv2D)       (None, 14, 14, 576)  55296       ['block_12_add[0][0]']
```

```
 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304        ['block_13_expand
[0][0]']
 lization)

 block_13_expand_relu (ReLU)    (None, 14, 14, 576)  0           ['block_13_expand_BN
[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 15, 15, 576)  0           ['block_13_expand_relu
[0][0]']

 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184        ['block_13_pad[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304        ['block_13_depthwise
[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0           ['block_13_depthwise_B
N[0][0]']

 block_13_project (Conv2D)      (None, 7, 7, 160)    92160       ['block_13_depthwise_r
elu[0][0]']

 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_13_project
[0][0]']
 alization)

 block_14_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_13_project_BN
[0][0]']

 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_14_expand
[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_14_expand_BN
[0][0]']

 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_14_expand_relu
```

```
[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_14_depthwise
[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_14_depthwise_B
N[0][0]']

 block_14_project (Conv2D)      (None, 7, 7, 160)    153600      ['block_14_depthwise_r
elu[0][0]']

 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_14_project
[0][0]']
 alization)

 block_14_add (Add)             (None, 7, 7, 160)    0           ['block_13_project_BN
[0][0]',
                                                                  'block_14_project_BN
[0][0]']

 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_14_add[0][0]']

 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_15_expand
[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_15_expand_BN
[0][0]']

 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_15_expand_relu
[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_15_depthwise
[0][0]']
 rmalization)
```

```
 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_15_depthwise_B
N[0][0]']

 block_15_project (Conv2D)      (None, 7, 7, 160)    153600      ['block_15_depthwise_r
elu[0][0]']

 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_15_project
[0][0]']
 alization)

 block_15_add (Add)             (None, 7, 7, 160)    0           ['block_14_add[0][0]',
                                                                  'block_15_project_BN
[0][0]']

 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_15_add[0][0]']

 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_16_expand
[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_16_expand_BN
[0][0]']

 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_16_expand_relu
[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_16_depthwise
[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_16_depthwise_B
N[0][0]']

 block_16_project (Conv2D)      (None, 7, 7, 320)    307200      ['block_16_depthwise_r
elu[0][0]']
```

```
 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280        ['block_16_project
[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600      ['block_16_project_BN
[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['Conv_1[0][0]']

 out_relu (ReLU)                (None, 7, 7, 1280)   0           ['Conv_1_bn[0][0]']

 flatten (Flatten)              (None, 62720)        0           ['out_relu[0][0]']

 dense (Dense)                  (None, 64)           4014144     ['flatten[0][0]']

==================================================================================
===========
Total params: 6,272,128
Trainable params: 4,014,144
```

In [31]:

```
# tell the model what cost and optimization method to use
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

In [32]:
```
# Use the Image Data Generator to import the images from the dataset
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

In [33]:
```
# Make sure you provide the same target size as initialied for the image size
training_set = train_datagen.flow_from_directory('/content/output/train',
                                                 target_size = (224, 224),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')
```

```
Found 2652 images belonging to 64 classes.
```

In [34]:
```
training_set
```

Out[34]:
```
<keras.preprocessing.image.DirectoryIterator at 0x7f728a035ad0>
```

In [35]:
```
test_set = test_datagen.flow_from_directory('/content/output/valid',
                                            target_size = (224, 224),
                                            batch_size = 32,
                                            class_mode = 'categorical')
```

```
Found 494 images belonging to 64 classes.
```

In [36]:
```
test_set
```

Out[36]:
```
<keras.preprocessing.image.DirectoryIterator at 0x7f728a017c50>
```

In [37]:
```
# fit the model
# Run the cell. It will take some time to execute
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=50,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: UserWarning: `Model.fit
_generator` is deprecated and will be removed in a future version. Please use `Model.fi
t`, which supports generators.
Epoch 1/50
83/83 [==============================] - 116s 1s/step - loss: 13.5967 - accuracy: 0.354
1 - val_loss: 9.9134 - val_accuracy: 0.5243
Epoch 2/50
83/83 [==============================] - 108s 1s/step - loss: 3.9260 - accuracy: 0.7357
- val_loss: 8.4701 - val_accuracy: 0.6093
Epoch 3/50
83/83 [==============================] - 108s 1s/step - loss: 2.4263 - accuracy: 0.8281
- val_loss: 9.7641 - val_accuracy: 0.5870
Epoch 4/50
83/83 [==============================] - 108s 1s/step - loss: 1.8751 - accuracy: 0.8624
- val_loss: 9.0653 - val_accuracy: 0.6336
Epoch 5/50
83/83 [==============================] - 109s 1s/step - loss: 1.3631 - accuracy: 0.8948
```

```
- val_loss: 11.5117 - val_accuracy: 0.6093
Epoch 6/50
83/83 [==============================] - 108s 1s/step - loss: 1.0970 - accuracy: 0.9148
- val_loss: 10.8895 - val_accuracy: 0.6559
Epoch 7/50
83/83 [==============================] - 107s 1s/step - loss: 1.2622 - accuracy: 0.9137
- val_loss: 11.2461 - val_accuracy: 0.6336
Epoch 8/50
83/83 [==============================] - 107s 1s/step - loss: 0.7833 - accuracy: 0.9374
- val_loss: 12.1522 - val_accuracy: 0.6336
Epoch 9/50
83/83 [==============================] - 108s 1s/step - loss: 0.6916 - accuracy: 0.9465
- val_loss: 12.7585 - val_accuracy: 0.6316
Epoch 10/50
83/83 [==============================] - 108s 1s/step - loss: 0.9540 - accuracy: 0.9397
- val_loss: 11.9963 - val_accuracy: 0.6781
Epoch 11/50
83/83 [==============================] - 108s 1s/step - loss: 0.6914 - accuracy: 0.9506
- val_loss: 13.1272 - val_accuracy: 0.6356
Epoch 12/50
83/83 [==============================] - 108s 1s/step - loss: 0.7636 - accuracy: 0.9476
- val_loss: 12.6158 - val_accuracy: 0.6377
Epoch 13/50
83/83 [==============================] - 107s 1s/step - loss: 0.6491 - accuracy: 0.9548
- val_loss: 14.9008 - val_accuracy: 0.6478
Epoch 14/50
83/83 [==============================] - 108s 1s/step - loss: 0.7107 - accuracy: 0.9581
- val_loss: 14.7905 - val_accuracy: 0.6417
Epoch 15/50
83/83 [==============================] - 109s 1s/step - loss: 0.5223 - accuracy: 0.9634
- val_loss: 14.9899 - val_accuracy: 0.6518
Epoch 16/50
83/83 [==============================] - 108s 1s/step - loss: 0.7663 - accuracy: 0.9551
- val_loss: 15.0947 - val_accuracy: 0.6559
Epoch 17/50
83/83 [==============================] - 108s 1s/step - loss: 0.7528 - accuracy: 0.9525
- val_loss: 18.2430 - val_accuracy: 0.6032
Epoch 18/50
```

```
83/83 [==============================] - 106s 1s/step - loss: 0.7236 - accuracy: 0.9627
- val_loss: 16.2001 - val_accuracy: 0.6498
Epoch 19/50
83/83 [==============================] - 108s 1s/step - loss: 0.4800 - accuracy: 0.9672
- val_loss: 18.2153 - val_accuracy: 0.6215
Epoch 20/50
83/83 [==============================] - 109s 1s/step - loss: 0.6013 - accuracy: 0.9627
- val_loss: 17.4676 - val_accuracy: 0.6316
Epoch 21/50
83/83 [==============================] - 108s 1s/step - loss: 0.3967 - accuracy: 0.9736
- val_loss: 21.3103 - val_accuracy: 0.6275
Epoch 22/50
83/83 [==============================] - 106s 1s/step - loss: 0.7697 - accuracy: 0.9612
- val_loss: 18.3591 - val_accuracy: 0.6336
Epoch 23/50
83/83 [==============================] - 116s 1s/step - loss: 0.5714 - accuracy: 0.9717
- val_loss: 17.8166 - val_accuracy: 0.6518
Epoch 24/50
83/83 [==============================] - 107s 1s/step - loss: 0.4882 - accuracy: 0.9679
- val_loss: 19.7114 - val_accuracy: 0.6518
Epoch 25/50
83/83 [==============================] - 105s 1s/step - loss: 0.6905 - accuracy: 0.9657
- val_loss: 19.6907 - val_accuracy: 0.6579
Epoch 26/50
83/83 [==============================] - 119s 1s/step - loss: 0.6931 - accuracy: 0.9664
- val_loss: 19.6609 - val_accuracy: 0.6457
Epoch 27/50
83/83 [==============================] - 110s 1s/step - loss: 0.5722 - accuracy: 0.9657
- val_loss: 22.6171 - val_accuracy: 0.6194
Epoch 28/50
83/83 [==============================] - 107s 1s/step - loss: 0.6244 - accuracy: 0.9713
- val_loss: 21.6574 - val_accuracy: 0.6599
Epoch 29/50
83/83 [==============================] - 107s 1s/step - loss: 0.4155 - accuracy: 0.9781
- val_loss: 20.9081 - val_accuracy: 0.6741
Epoch 30/50
83/83 [==============================] - 113s 1s/step - loss: 0.3978 - accuracy: 0.9781
- val_loss: 22.0992 - val_accuracy: 0.6377
```

```
Epoch 31/50
83/83 [==============================] - 106s 1s/step - loss: 0.4555 - accuracy: 0.9717
- val_loss: 23.0920 - val_accuracy: 0.6174
Epoch 32/50
83/83 [==============================] - 106s 1s/step - loss: 0.5813 - accuracy: 0.9744
- val_loss: 21.6818 - val_accuracy: 0.6457
Epoch 33/50
83/83 [==============================] - 107s 1s/step - loss: 0.8232 - accuracy: 0.9653
- val_loss: 21.4054 - val_accuracy: 0.6559
Epoch 34/50
83/83 [==============================] - 118s 1s/step - loss: 0.4122 - accuracy: 0.9827
- val_loss: 22.1703 - val_accuracy: 0.6498
Epoch 35/50
83/83 [==============================] - 108s 1s/step - loss: 0.4900 - accuracy: 0.9729
- val_loss: 24.3012 - val_accuracy: 0.6478
Epoch 36/50
83/83 [==============================] - 108s 1s/step - loss: 0.6212 - accuracy: 0.9736
- val_loss: 24.4139 - val_accuracy: 0.6194
Epoch 37/50
83/83 [==============================] - 108s 1s/step - loss: 0.6755 - accuracy: 0.9755
- val_loss: 26.8997 - val_accuracy: 0.6397
Epoch 38/50
83/83 [==============================] - 109s 1s/step - loss: 0.6006 - accuracy: 0.9740
- val_loss: 28.6756 - val_accuracy: 0.6316
Epoch 39/50
83/83 [==============================] - 109s 1s/step - loss: 0.5606 - accuracy: 0.9800
- val_loss: 26.4944 - val_accuracy: 0.6336
Epoch 40/50
83/83 [==============================] - 107s 1s/step - loss: 0.4865 - accuracy: 0.9800
- val_loss: 24.2302 - val_accuracy: 0.6640
Epoch 41/50
83/83 [==============================] - 109s 1s/step - loss: 0.4521 - accuracy: 0.9796
- val_loss: 27.0038 - val_accuracy: 0.6437
Epoch 42/50
83/83 [==============================] - 109s 1s/step - loss: 0.4985 - accuracy: 0.9830
- val_loss: 26.2606 - val_accuracy: 0.6417
Epoch 43/50
83/83 [==============================] - 109s 1s/step - loss: 0.4677 - accuracy: 0.9789
```

```
- val_loss: 25.6897 - val_accuracy: 0.6660
Epoch 44/50
83/83 [==============================] - 108s 1s/step - loss: 0.5012 - accuracy: 0.9800
- val_loss: 26.9411 - val_accuracy: 0.6822
Epoch 45/50
83/83 [==============================] - 109s 1s/step - loss: 0.4085 - accuracy: 0.9830
- val_loss: 25.9641 - val_accuracy: 0.6781
Epoch 46/50
83/83 [==============================] - 108s 1s/step - loss: 0.4268 - accuracy: 0.9815
- val_loss: 26.2403 - val_accuracy: 0.6538
Epoch 47/50
83/83 [==============================] - 108s 1s/step - loss: 0.4324 - accuracy: 0.9830
- val_loss: 27.6515 - val_accuracy: 0.6640
Epoch 48/50
83/83 [==============================] - 108s 1s/step - loss: 0.4690 - accuracy: 0.9796
- val_loss: 28.9115 - val_accuracy: 0.6700
Epoch 49/50
83/83 [==============================] - 108s 1s/step - loss: 0.4499 - accuracy: 0.9808
- val_loss: 29.6312 - val_accuracy: 0.6397
Epoch 50/50
83/83 [==============================] - 107s 1s/step - loss: 0.3122 - accuracy: 0.9819
```

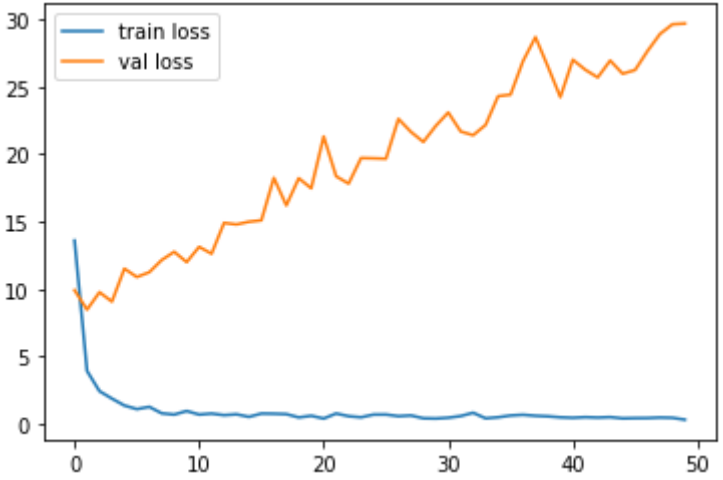
In [46]:

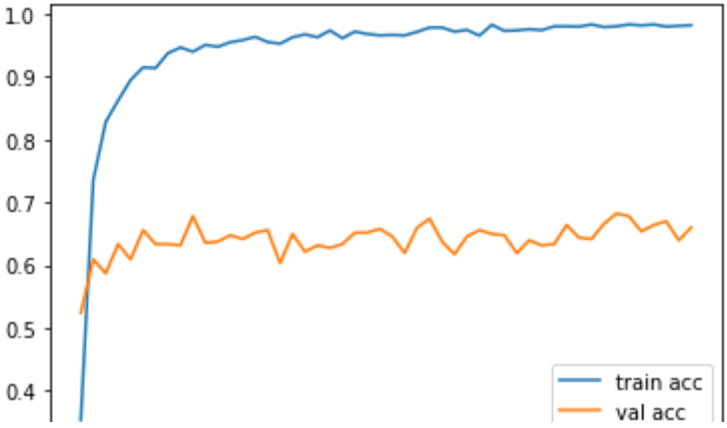
```
#from google.colab import drive
#drive.mount('/content/drive')
```

In [47]:

```
# plot the loss
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')

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





In [48]:
```
# save it as a h5 file

from tensorflow.keras.models import load_model

model.save('model_mobilenetv2_new.h5')
```

```
/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py:1410: CustomMaskWarni
ng: Custom mask layers require a config and must override get_config. When loading, the
custom mask layer must be passed to the custom_objects argument.
  layer_config = serialize_layer_fn(layer)
```

In [49]:
```
y_pred = model.predict(test_set)
```

In [50]:
```
y_pred
```

Out[50]:
```
array([[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00],
```

```
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
 0.0000000e+00, 0.0000000e+00],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
 0.0000000e+00, 1.0000000e+00],
...,
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
 0.0000000e+00, 0.0000000e+00],
[0.0000000e+00, 0.0000000e+00, 6.2252050e-18, ..., 5.5498380e-33,
 1.5125389e-31, 4.6246727e-36],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
 0.0000000e+00, 0.0000000e+00]], dtype=float32)
```

In [51]:

```
import numpy as np
y_pred = np.argmax(y_pred, axis=1)
```

In [52]:

```
# Evaluating model on validation data
evaluate = model.evaluate(test_set)
print(evaluate)
```

```
16/16 [==============================] - 12s 766ms/step - loss: 29.6846 - accuracy: 0.6
599
[29.684602737426758, 0.659919023513794]
```

In [53]:

```
from sklearn.metrics import classification_report, confusion_matrix
def give_accuracy():
    p=model.predict(test_set)
    cm=confusion_matrix(y_true=test_set.classes,y_pred=np.argmax(p,axis=-1))
    acc=cm.trace()/cm.sum()
    print('The Classification Report \n', cm)
    print(f'Accuracy: {acc*100}')
```

In [54]:
```
give_accuracy()
```

```
The Classification Report
 [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 1 ... 0 0 1]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 1]]
Accuracy: 1.2145748987854251
```

In [55]:
```
import numpy as np
from tensorflow.keras.preprocessing import image
test_image = image.load_img('/content/output/train/Margaret Thatcher/000002.jpg', targe
test_image = image.img_to_array(test_image)
test_image=test_image/255
test_image = np.expand_dims(test_image, axis = 0)
result = model.predict(test_image)
```

In [56]:
```
test = np.array(test_image)
```

In [57]:
```
# making predictions
#prediction = np.argmax(cnn.predict(test_image), axis=-1)
prediction = np.argmax(model.predict(test_image))
```

In [58]:
```
prediction
```

Out[58]: 35

In [62]:

```
output = {
0:'Ada Lovelace',
1:'Alice Milliat',
2:'Amelia Earhart',
3:'Amrita Priam',
4:'Andrea Dworkin',
5:'Angela Burdett-Coutts',
6:'Angela Merkel',
7:'Anna Akhmatova',
8:'Aretha Franklin',
9:'Audrey Hepburn',
10:'Bessie Coleman',
11:'Bette Davis',
12:'Betty White',
13:'Buchi Emecheta',
14:'Celine Dion',
15:'Cher',
16:'Coco Chanel',
17:'Diana Ross',
18:'Diana Princess of Wales',
19:'Emmeline Pankhurst',
20:'Estee Lauder',
21:'Florence Nightingale',
22:'Frida Kahlo',
23:'Gertrude Ederle',
24:'Grace Hopper',
25:'Jacinda Ardern',
26:'Joan Robinson',
27:'Judy Garland',
28:'Junko Tabei',
29:'Kamala Harris',
30:'Katharine Graham',
```

```
32:'Katia Krafft',
33:'Lise Meitner',
34:'Madonna',
35:'Margaret Thatcher',
36:'Marie Curie',
37:'Marie Stopes',
38:'Marie Van Brittan Brown',
39:'Marilyn Monroe',
40:'Mary Wollstonecraft',
41:'Maryam Mirzakhani',
42:'Meryl Streep',
43:'Nancy Pelosi',
44:'Nina Simone',
45:'Oprah Winfrey',
46:'Rachael Heyhoe Flint',
47:'Raja Easa Al Gurg',
48:'Rihanna',
49:'Rosa Parks',
50:'Ruth Handler',
51:'Serena Williams',
52:'Sheryl Sandberg',
53:'Simone de Beauvoir',
54:'Sirimavo Bandaranaike',
55:'Sonja Henie',
56:'Susan Sontag',
57:'Suzanne Lenglen',
58:'Vera Atkins',
59:'Virginia Woolf',
60:'Wangari Maathai',
61:'Whitney Houston',
62:'Wilma Rudolph',
63:'Zora Neale Hurston'
```

In [63]:
```
print("The prediction Of the Image is : ", output[prediction])
```

```
The prediction Of the Image is :  Margaret Thatcher
```

In [64]:
```
# show the image
import matplotlib.pyplot as plt
test_image = image.load_img('/content/output/train/Margaret Thatcher/000002.jpg', targe
plt.axis('off')
plt.imshow(test_image)
plt.show()
```

In [65]:
```
prediction
```

Out[65]: 35

In [66]:
```
print("The prediction Of the Image is : ", output[prediction])
```

```
The prediction Of the Image is :  Margaret Thatcher
```

In [ ]:
```
### Similarly,
```

In [67]:
```
import numpy as np
from tensorflow.keras.preprocessing import image
test_image = image.load_img('/content/output/train/Madonna/000002.jpg', target_size = (
test_image = image.img_to_array(test_image)
test_image=test_image/255
test_image = np.expand_dims(test_image, axis = 0)
result = model.predict(test_image)
```

In [68]:
```
test = np.array(test_image)
```

In [69]:
```
# making predictions
#prediction = np.argmax(cnn.predict(test_image), axis=-1)
prediction = np.argmax(model.predict(test_image))
```

In [70]:
```
prediction
```

Out[70]: 34

In [71]:
```
print("The prediction Of the Image is : ", output[prediction])
```

```
The prediction Of the Image is :  Madonna
```

In [72]:
```
# show the image
import matplotlib.pyplot as plt
test_image = image.load_img('/content/output/train/Madonna/000002.jpg', target_size = (
plt.axis('off')
plt.imshow(test_image)
plt.show()
```