```
import sys
import os
```

Za početak je potrebno u Jupyter okruženje instalirati biblioteke koje će koristiti prilikom izrade projek
opencv-utils, opencv-python, mtcnn i tensorflow. U narednim isječcima koda date su instalacije tih bil

```
!{sys.executable} -m pip install opencv-utils
```

```
Requirement already satisfied: opencv-utils in /usr/local/lib/python3.6/dist-packages (0
```

```
!{sys.executable} -m pip install opencv-python
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.6/dist-packages (
```

```
!{sys.executable} -m pip install mtcnn
```

```
Requirement already satisfied: mtcnn in /usr/local/lib/python3.6/dist-packages (0.1.0)
Requirement already satisfied: opencv-python>=4.1.0 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: keras>=2.0.0 in /usr/local/lib/python3.6/dist-packages (1
Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: keras-preprocessing>=1.1.0 in /usr/local/lib/python3.6/di
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.6/dis
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: pyyaml in /usr/local/lib/python3.6/dist-packages (from ke
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from kera
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.6/dist-packages (fro
```

Saved successfully!

```
!{sys.executable} -m pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.6/dist-packages (1.1
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.6/di
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: tensorflow-estimator==1.15.1 in /usr/local/lib/python3.6/
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: gast==0.2.2 in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.6/dis
Requirement already satisfied: google-pasta>=0.1.6 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: tensorboard<1.16.0,>=1.15.0 in /usr/local/lib/python3.6/d
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (fro
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from kera
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.6/dist-packag
```

Projekat se okvirno može podijeliti u tri dijela. U prvom dijelu projekta potrebno je pripremiti dataset, (bi treniranje a treći verifikaciju. Stoga se pozabavimo prvim dijelom projekta, odnosno predprocesiran istog za korištenje u treniranju i validaciji. U Jupyter notebook-ovim fajlovima kreiran je folder pod na uploadovane fotografije iz dataset-a. Ove slike je potrebno učitati u kod. Uz pomoć narednog koda će fajlovi.

Saved successfully!

```
from os.path import isfile, join
import zipfile

zf = zipfile.ZipFile("/content/utkcropped.zip")
zf.extractall('/content/sveslike')

fajlovi = [f for f in listdir('/content/sveslike') if isfile(join('/content/sveslike', f))]
```

Kada u listi imamo imena slika svog dataset-a, ove slike ćemo funkcijom imread() učitati u listu sa na

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#import matplotlib.image as mpimg
#from matplotlib import ticker
from mtcnn import MTCNN
import cv2

slike=list()
```

```
for i in range(0,len(fajlovi)):
  img = cv2.cvtColor(cv2.imread('/content/sveslike/'+fajlovi[i]), cv2.COLOR_BGR2RGB)
  slike.append(img)
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic: more info.

Učitali smo fotografije i sada se treba preći na pretprocesiranje. Iz biblioteke MTCNN koristit ćemo fu se ne detektuje niti jedno lice na slici potrebno je da tu sliku uklonimo.

```
detector = MTCNN()
lista=list()
for i in range(0,len(slike)):
  if len(detector.detect_faces(slike[i]))==0:
    lista.append(i)
lista.sort(reverse=True)
for i in lista:
  fajlovi.pop(i)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
ocal/lib/python3.6/dist-packages/keras/backend/tensorflow_
ocal/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
```


Konačno, pretprocesirani dataset možemo podijeliti u dva dijela, jedan za trening a drugi za verifikaci u omjeru 80%:20% za trening:verifikaciju. Pošto su fotografije sortirane redom po godinama osobe na ravnomjerno raspoređena količina osoba po godinama, svaki peti element ćemo ubaciti u set za verif treniranje.

```
brver=int(len(fajlovi)/5)+1
brtren=len(fajlovi)-brver
verifikacija=np.ndarray((brver,3,64,64),dtype=np.uint8)
```

```
trening=np.ndarray((brtren,3,64,64),dtype=np.uint8)
yver=np.ndarray(brver,dtype=np.uint8)
ytren=np.ndarray(brtren,dtype=np.uint8)
for i in range(0,len(fajlovi)):
  #img = cv2.cvtColor(cv2.imread('/content/sveslike/'+fajlovi[i]), cv2.COLOR_BGR2RGB)
  img = cv2.imread('/content/sveslike/'+fajlovi[i], cv2.IMREAD_COLOR) #cv2.IMREAD_GRAYSCALE
  img=cv2.resize(img, (64, 64), interpolation=cv2.INTER_CUBIC)
  #img=img/255
  a=fajlovi[i].find('_')+1
  b=fajlovi[i].find('_',a,len(fajlovi[i]))
  if i%5==0:
    verifikacija[int(i/5)]=img.T
    yver[int(i/5)]=fajlovi[i][a:b]
  else:
    trening[i-int(i/5)-1]=img.T
    ytren[i-int(i/5)-1]=fajlovi[i][a:b]


trenbrver=int(brtren/5)+1
trenbrtren=brtren-trenbrver
treningver=np.ndarray((trenbrver,3,64,64),dtype=np.uint8)
treningtren=np.ndarray((trenbrtren,3,64,64),dtype=np.uint8)
trenyver=np.ndarray(trenbrver,dtype=np.uint8)
trenytren=np.ndarray(trenbrtren,dtype=np.uint8)
for i in range(0,brtren):
  if i%5==0:
    treningver[int(i/5)]=trening[i]
    trenyver[int(i/5)]=ytren[i]
  else:
    ng[i]
    ]
```

Saved successfully!

```
from keras.models import Sequential
from keras.layers import Input, Dropout, Flatten, Convolution2D, MaxPooling2D, Dense, Activat
from keras.optimizers import RMSprop, Adam
from keras.callbacks import ModelCheckpoint, Callback, EarlyStopping
from keras.utils import np_utils

lr=1e-4
brepoha=30
optimizer = Adam(lr=lr, decay=lr/brepoha)
objective = 'binary_crossentropy'

model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=(3, 64, 64), activation='re
model.add(Convolution2D(32, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering="th"))
```

```
model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering="th"))

model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering="th"))

model.add(Convolution2D(256, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(256, 3, 3, border_mode='same', activation='relu'))
#     model.add(Convolution2D(256, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering="th"))



model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss=objective, optimizer=optimizer, metrics=['accuracy'])
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:14: UserWarning: Update you
Saved successfully!
packages/ipykernel_launcher.py:15: UserWarning: Update you
app as app
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:16: UserWarning: Update you
  app.launch_new_instance()
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:18: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:19: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:20: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:22: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:23: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:24: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:26: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:27: UserWarning: Update you
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:29: UserWarning: Update you
```

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_29 (Conv2D)           (None, 3, 64, 32)         18464
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 3, 64, 32)         9248
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 3, 32, 16)         0
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 3, 32, 64)         9280
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 3, 32, 64)         36928
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 3, 16, 32)         0
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 3, 16, 128)        36992
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 3, 16, 128)        147584
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 3, 8, 64)          0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 3, 8, 256)         147712
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 3, 8, 256)         590080
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 3, 4, 128)         0
_________________________________________________________________
flatten_5 (Flatten)          (None, 1536)              0
_________________________________________________________________
dense_14 (Dense)             (None, 256)               393472
_________________________________________________________________
                             None, 256)                0
_________________________________________________________________
                             (None, 256)               65792
_________________________________________________________________
dropout_6 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_16 (Dense)             (None, 1)                 257
_________________________________________________________________
activation_3 (Activation)    (None, 1)                 0
=================================================================
Total params: 1,455,809
Trainable params: 1,455,809
Non-trainable params: 0
_________________________________________________________________
```

Saved successfully!

```
#brepoha = 30
batch_size = 16
#labs = train_data.iloc[:,1].values.tolist()

## Callback for loss logging per epoch
class LossHistory(Callback):
    def on_train_begin(self, logs={}):
        self.losses = []
```

```
        self.val_losses = []
        self.accuracy = []
        self.val_accuracy = []

    def on_epoch_end(self, batch, logs={}):
        self.losses.append(logs.get('loss'))
        self.val_losses.append(logs.get('val_loss'))
        self.accuracy.append(logs.get('acc'))
        self.val_accuracy.append(logs.get('val_acc'))

early_stopping = EarlyStopping(monitor='val_loss', patience=3, verbose=1, mode='auto')
history = LossHistory()


model.fit(treningtren, trenytren, batch_size=batch_size, epochs=brepoha, validation_data=trer


 Epoch 00014: early stopping
    <keras.callbacks.History at 0x7fcb9e210898>


predikcije = model.predict(verifikacija, verbose=0)
predikcije

 array([[3.2683909e-03],
        [1.7071068e-03],
        [9.8748243e-01],
        ...,
        [6.4253807e-05],
        [9.9999845e-01],
        [9.9980336-01]], dtype=float32)
```

Saved successfully!

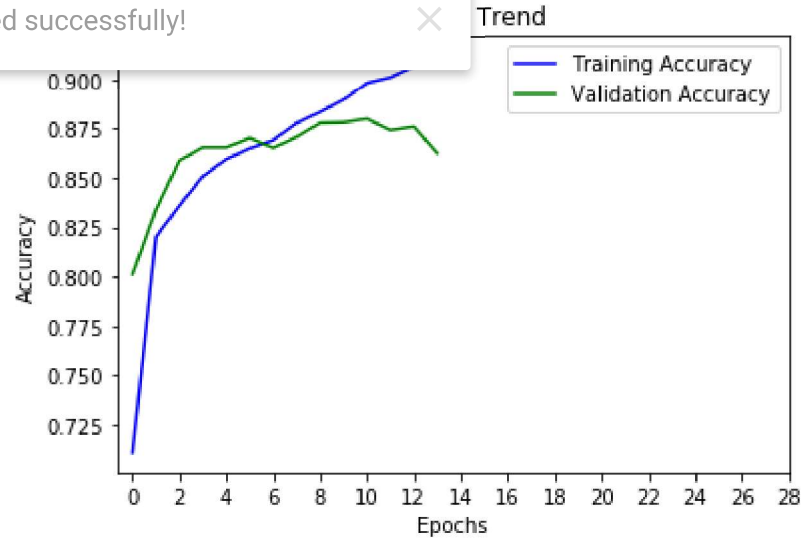
```
val_loss = history.val_losses

plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Trend')
plt.plot(loss, 'blue', label='Training Loss')
plt.plot(val_loss, 'green', label='Validation Loss')
plt.xticks(range(0,brepoha)[0::2])
plt.legend()
plt.show()
```


```
accuracy = history.accuracy
val_accuracy = history.val_accuracy

plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Accuracy Trend')
plt.plot(accuracy, 'blue', label='Training Accuracy')
plt.plot(val_accuracy, 'green', label='Validation Accuracy')
plt.xticks(range(0,brepoha)[0::2])
plt.legend()
plt.show()
```

Saved successfully!



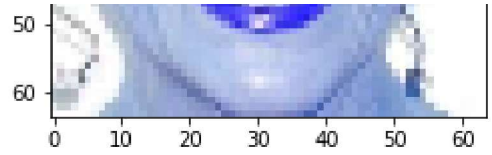

```
ypredikcije=np.ndarray(len(predikcije),dtype=np.uint8)
for i in range(0,len(verifikacija)):
    if predikcije[i, 0] >= 0.5:
        print('I am {:.2%} sure this is a Female'.format(predikcije[i][0]))

    else:
```

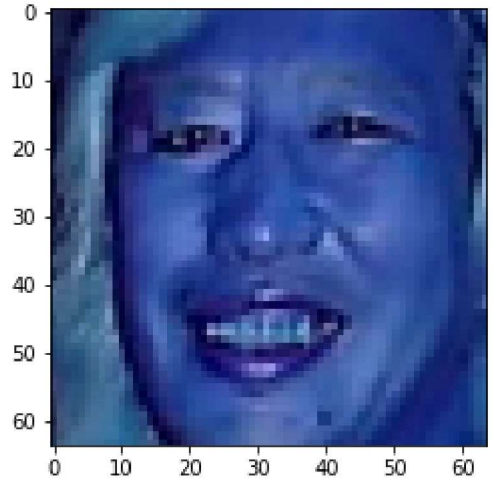
```
            print('I am {:.2%} sure this is a Male'.format(1-predikcije[i][0]))

        plt.imshow(verifikacija[i].T)
        plt.show()
```

Saved successfully!



```
I am 65.01% sure this is a Female
```



```
I am 99.98% sure this is a Male
```

```
I am 91.60% sure this is a Female
```

Saved successfully!

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-97-cb46c9b7e2cd> in <module>()
      8
      9     plt.imshow(verifikacija[i].T)
---> 10     plt.show()

 30 frames

</usr/local/lib/python3.6/dist-packages/decorator.py:decorator-gen-9> in __call__(self,

/usr/local/lib/python3.6/dist-packages/matplotlib/axis.py in _set_artist_props(self, a)
    324
    325     def _set_artist_props(self, a):
--> 326         a.set_figure(self.figure)
    327
    328     def get_view_interval(self):

KeyboardInterrupt:
```

SEARCH STACK OVERFLOW

SEARCH STACK OVERFLOW

Saved successfully!

Saved successfully!