Biologically Inspired Artificial Intelligence

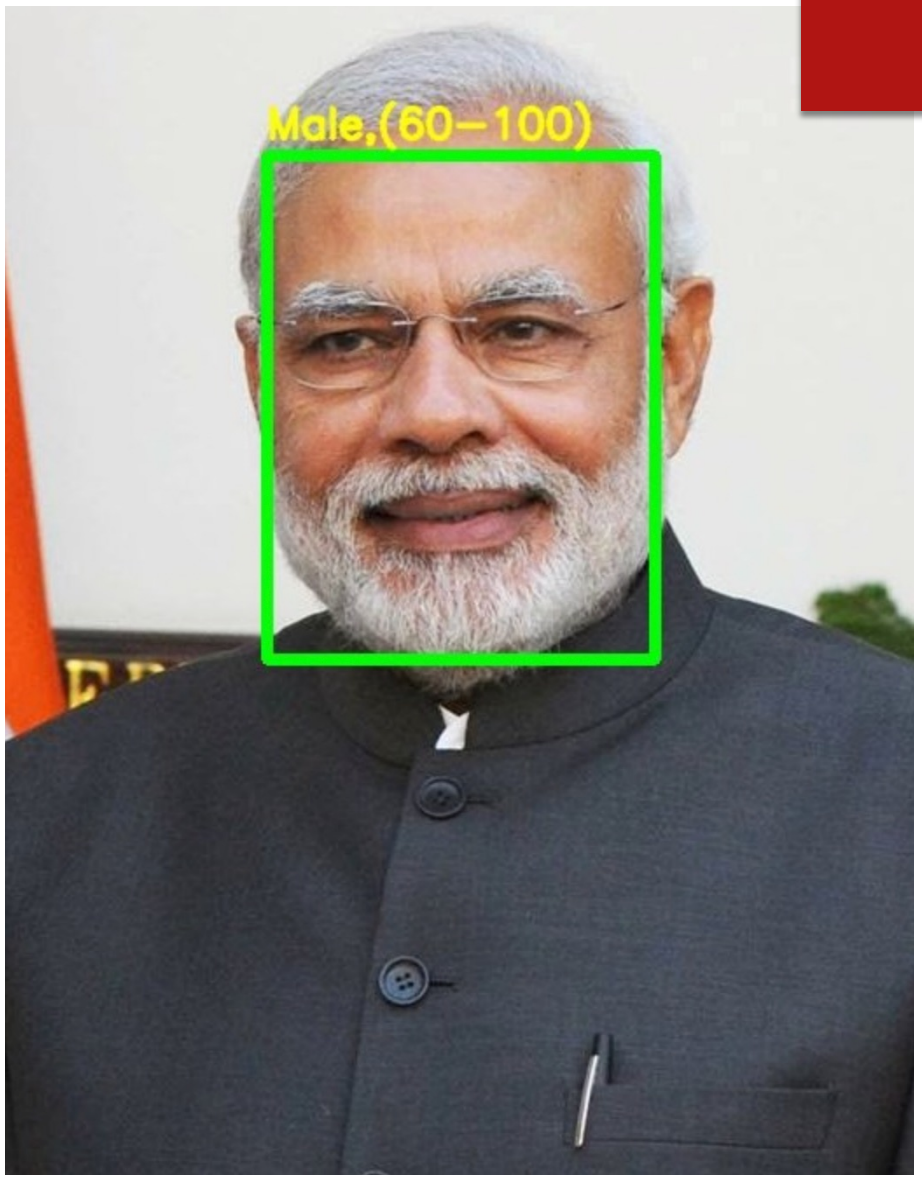
Gender recognition

Authors:
Mikołaj Malich
Adam Warzecha

Supervisor: dr inż. Grzegorz Baron

Topic description

- The main purpose of our project was to recognize gender from previously prepared human face's images.
- To implement and test algorithms we used Kaggle notebook in cooperation with Tensorflow, SKLearn, MatplotLib, NumPy and OpenCV.



Illustrative photo

The Dataset

Dataset was divided to 2 parts:
- training set (about 47 000 photos),
- validation (test) set (about 11 500 photos).

Each part of dataset contained 2 folders - one for women faces images and another one for men's in a ratio of 50:50.

Load image files from the dataset

```
train_images = [f for f in glob.glob(r'../input/gender-classification-dataset/Training' + "/**/*", recursive=True) if not os.path.isdir(f)]
test_images = [f for f in glob.glob(r'../input/gender-classification-dataset/Validation' + "/**/*", recursive=True) if not os.path.isdir(f)]
random.shuffle(train_images)
random.shuffle(test_images)

print("Train size: " + str(len(train_images)))
print("Test size: " + str(len(test_images)))
```

```
Train size: 47009
Test size: 11649
```

https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset

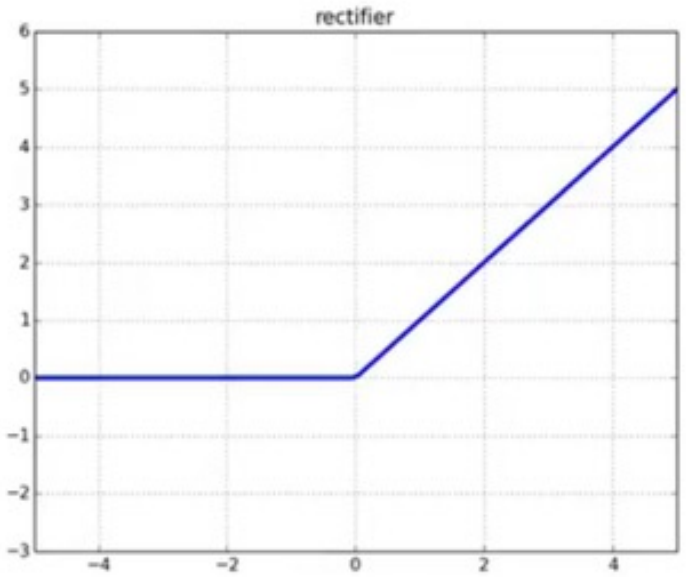
All the photos were imported to algorithm straight from Kaggle cloud database due to its size – ca. 300MB.

Provided dataset cointains only prepared, processed images with human faces, so we focused on gender recognition algorithm.

The Model

ReLU

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \Rightarrow 0 \end{cases}$$



Most important facts:

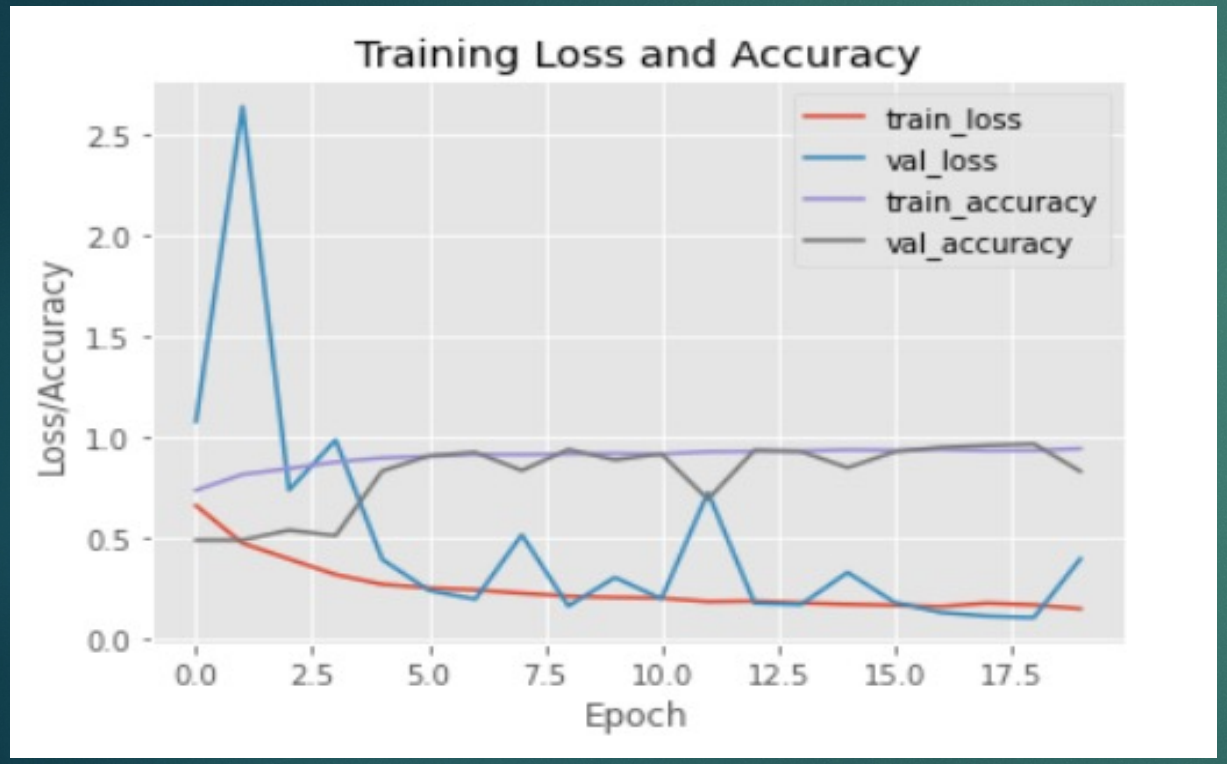
- 5 Convolutional Layers in which 1st one contains 32 filters, next two ones 64 filters, last 2 layers include 128 filters,
- Max pooling 2D is used to reduce spatial dimensions of the output volume,
- ReLU activation function (output the input directly if it is positive, otherwise, it will output zero).

```
model.add(Conv2D(32, (3,3), padding="same", input_shape=inputShape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3,3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

model.add(Conv2D(64, (3,3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3,3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

model.add(Conv2D(128, (3,3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(1024))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(classes))
model.add(Activation("sigmoid"))
```

Few examples of training

Parameters:
- 4700 training photos,
- 1116 test photos,
- 20 epoch

Sample A

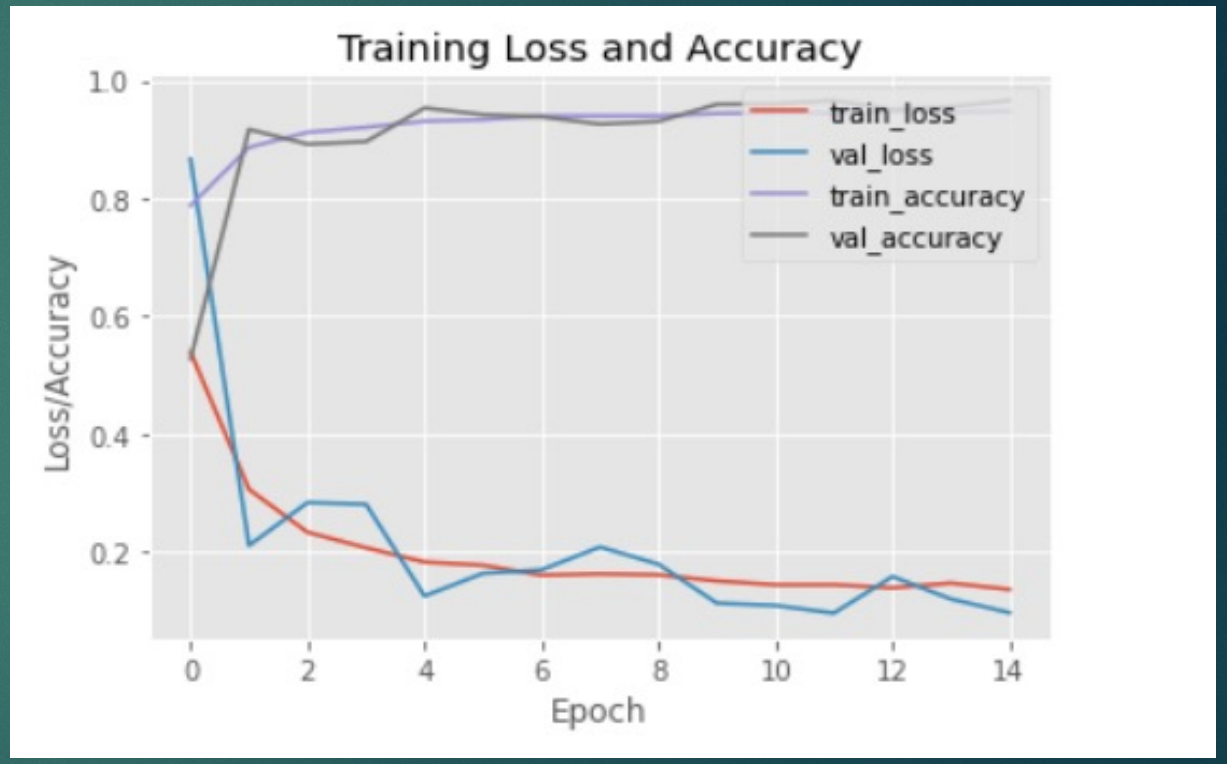


Final results:

- loss: 0.1511
- accuracy: 0.9443
- val_loss: 0.3984
- val_accuracy: 0.8316

Parameters:
- 14102 training photos,
- 3494 test photos,
- 15 epoch

Sample B

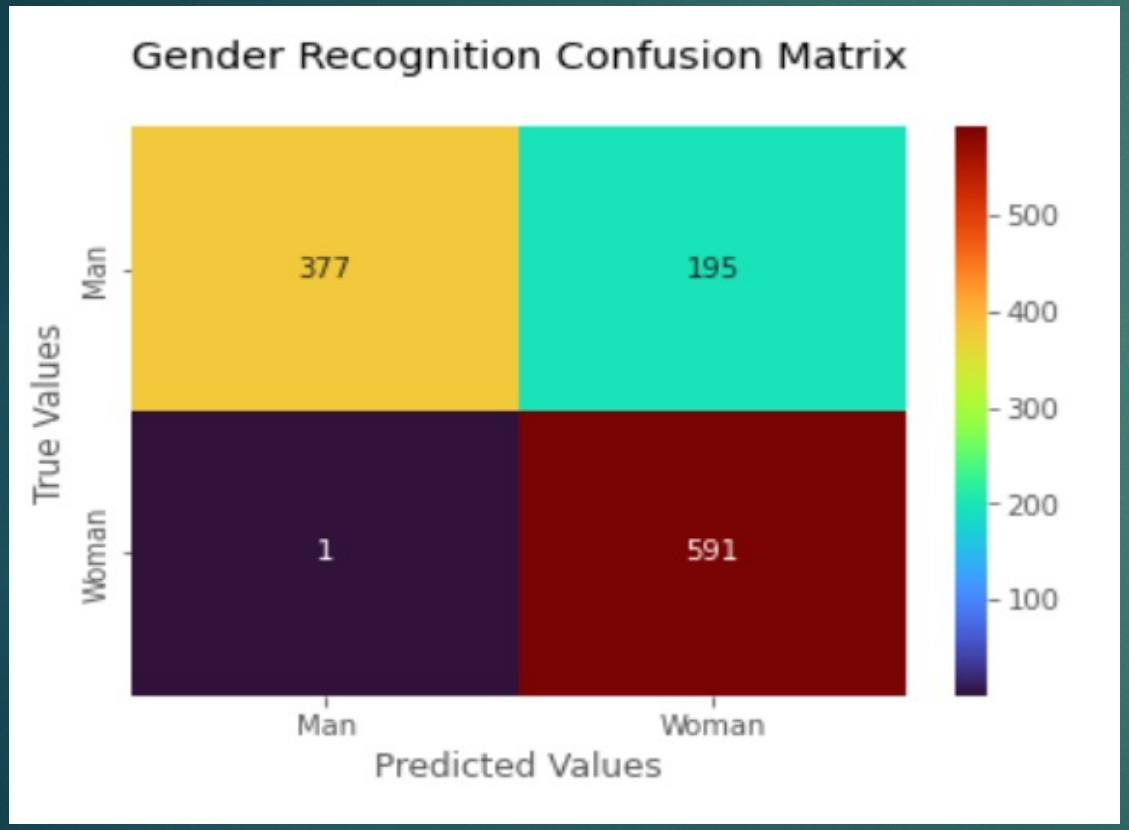


Final results:

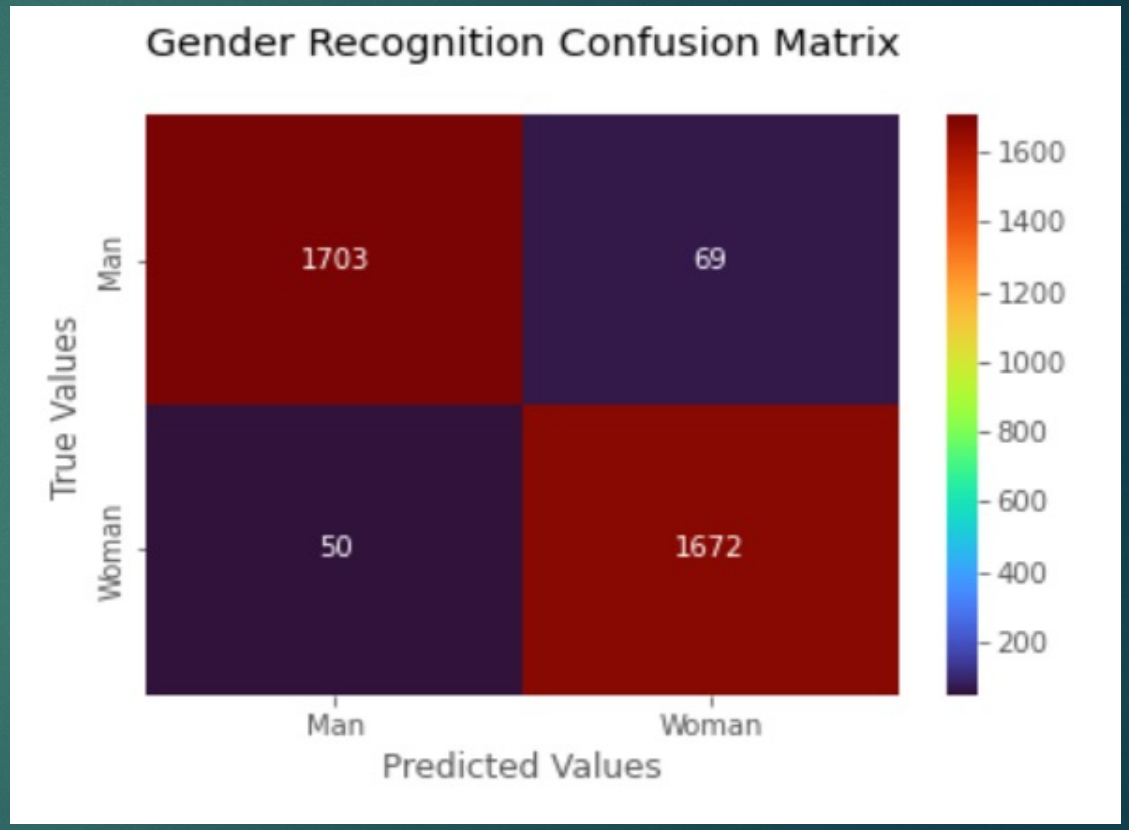
- loss: 0.1359
- accuracy: 0.9479
- val_loss: 0.0964
- val_accuracy: 0.9659

Few examples of training

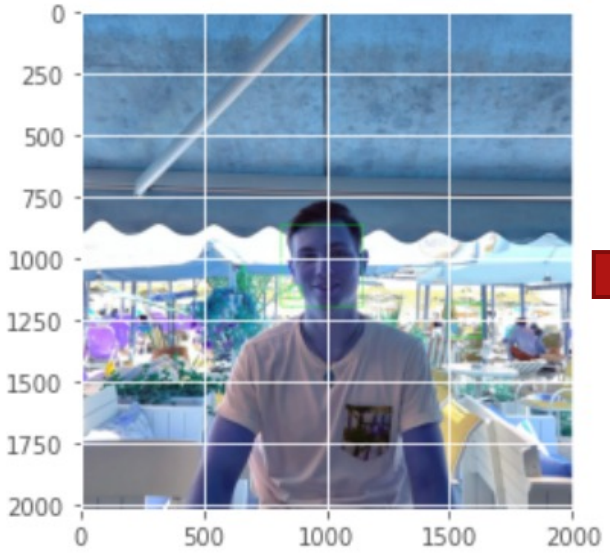
Sample A

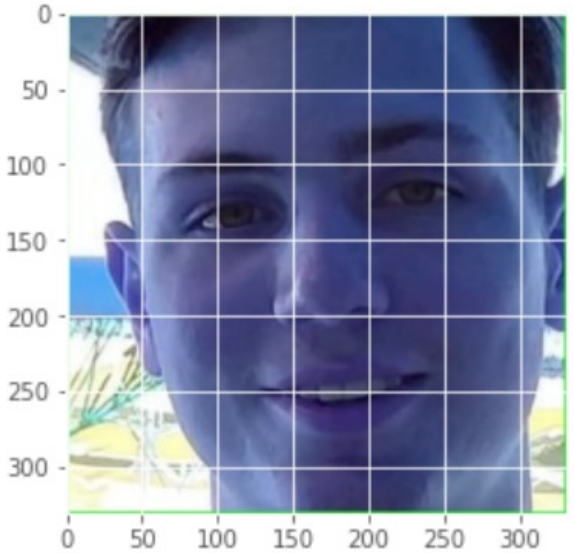


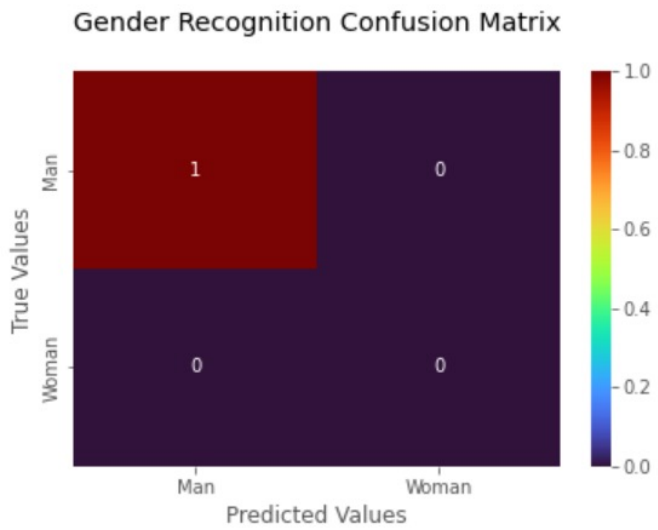
Sample B



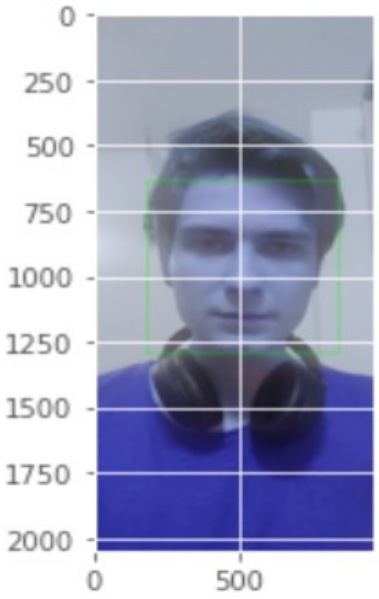
As seen above, probe B gives much better results.

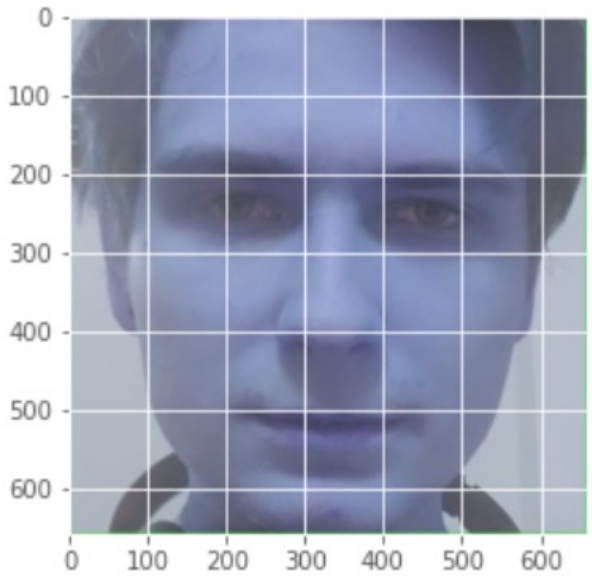


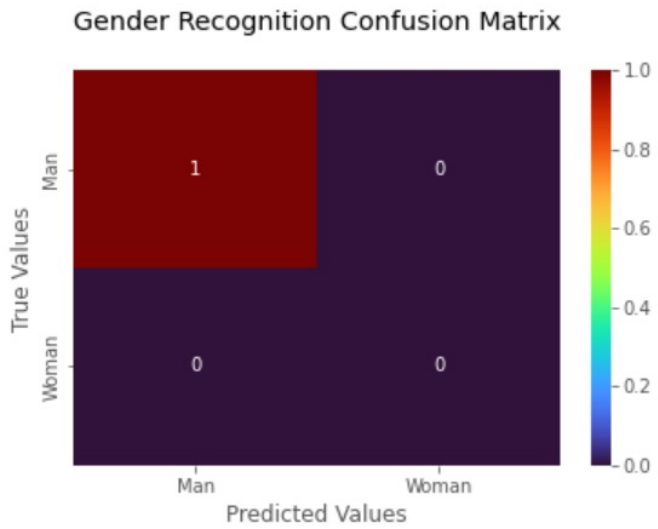




Bonus – prediction outside dataset



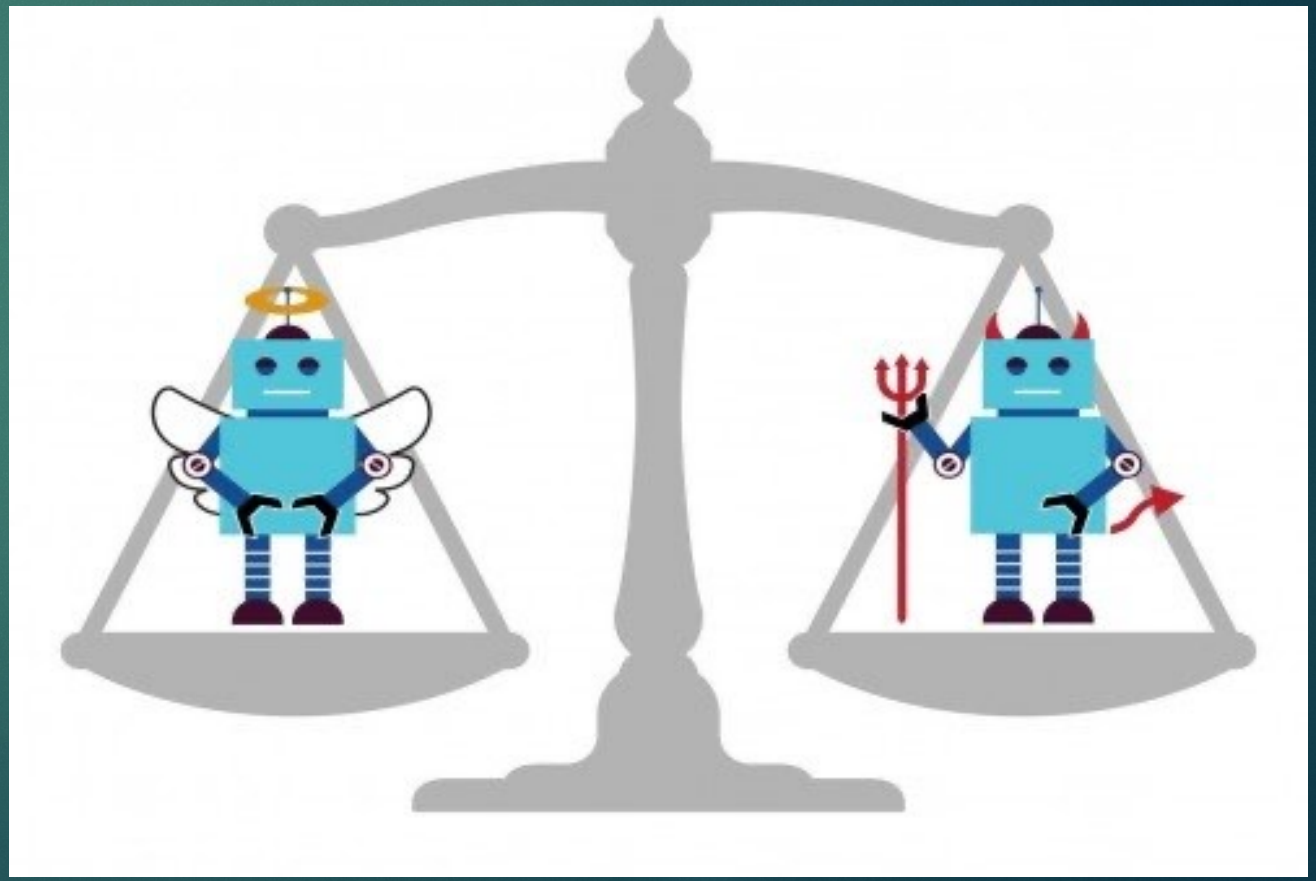




Bonus – prediction outside dataset (2)

Conclusions

- We gained hans-on-experience about machine learning,
- Google Colab gives more flexibility (ex. personal camera usage),
- It is better to have more training images and less epochs than less training images and more epochs,
- Artificial Intelligence is complicated issue but gives invaluable opportunities to make life easier,
- Thanks to AI-powered devices people can save their precious time in everyday activities,
- Project impoved our Python knowledge and coding skills,
- Tutorials and articles about background of Artificial Intelligence were very helpful and interesting.



Thank you for attention!

Authors:
Mikołaj Malich
Adam Warzecha

```
# size = The spatial size that the CNN expects. Options are = (224*224,
# mean = mean substraction values to be substracted from every channel o
# swapRB=OpenCV assumes images in BGR whereas the mean is supplied in RGB
blob = cv2.dnn.blobFromImage(image=face_img, scalefactor=1.0, size=(
    227, 227), mean=MODEL_MEAN_VALUES, swapRB=False, crop=False)
# Predict Gender
gender_net.setInput(blob)
gender_preds = gender_net.forward()
i = gender_preds[0].argmax()
gender = GENDER_LIST[i]
gender_confidence_score = gender_preds[0][i]
# Draw the box
label = "{}-{:.2f}%".format(gender, gender_confidence_score*100)
print(label)
yPos = start_y - 15
while yPos < 15:
    yPos += 15
```

Sources: Google Images, Wikipedia,
https://pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/