

1) Načtěte obrázek pvi\_cv07\_text.bmp a obrázky A-Z z pvi\_cv07\_dir\_znaky.zip. Obrazy segmentujte a z binárních obrázků A-Z spočítejte vektory parametrů pomocí horizontální a vertikální projekce. Pro zjednodušení spojte oba vektory do jednoho, např.:

```
E >>> [5.0, 3.0, 3.0, 3.0, 3.0, 5.0, 1.0, 5.0, 1.0, 5.0]
```

2) V binárním obraze (pvi\_cv07\_text.bmp) detekujte jednotlivé znaky pomocí barvení oblastí. Nalezené oblasti zparametrizujte (pomocí projekce) a porovnejte se vzory (vektory příznaků) A-Z na základě nejmenší vzdálenosti. Výstupem by měl být rozpoznaný text vypsaný do konzole:

```
FAKULTA_MECHATRONIKY_INFORMATIKY_A_MEZIOBOROVYCH_STUDII
```

3) V obraze pvi\_cv07\_people.jpg detekujte jednotlivé obličeje pomocí Viola-Jonesovu detektoru. Referenční boxy jsou uloženy v souboru pvi\_cv07\_boxes\_01.txt ve formátu (x, y, šířka, výška). Na základě vyhodnocení pro IoU = 0,5 spočítejte accuracy, precision a recall.

nápověda:

```
import os
listF = os.listdir('dir_znaky')

fillImage = 'pvi_cv07_people.jpg'
bgr = cv2.imread(fillImage)
gray = cv2.cvtColor(bgr, cv2.COLOR_RGB2GRAY)

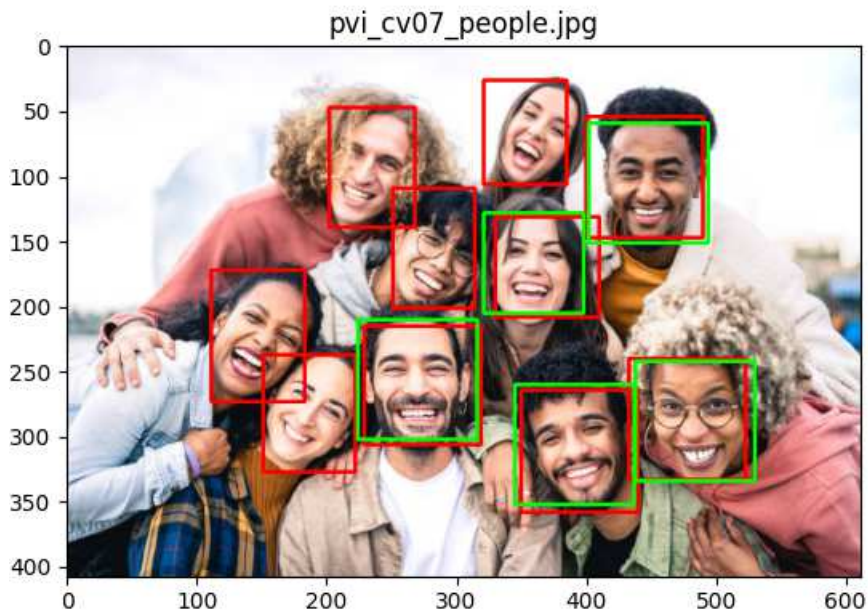
boxes = []
with open('pvi_cv07_boxes_01.txt') as f:
    lines = f.read().splitlines()
    for line in lines:
        vec = line.split(' ')
        vec = [int(x) for x in vec]
        boxes.append(vec)

for (x, y, w, h) in boxes:
    cv2.rectangle(bgr, (x, y), (x + w, y + h), (0, 0, 255), 2)

faceCascade = cv2.CascadeClassifier(
    'pvi_cv07_haarcascade_frontalface_default.xml') # Detect faces in the
image
faces = faceCascade.detectMultiScale(
```

```
gray,  
scaleFactor=1.4,  
minNeighbors=5,  
minSize=(30, 30),  
flags=cv2.CASCADE_SCALE_IMAGE)
```

očekávaný výstup:



Accuracy: 0.5  
Precision: 0.5  
Recall: 1.0