Instructions on how it was trained
Mukhtar Rabayev
Hof University of Applied Sciences

Firstly, collected all photos like that in Jupyter Notebook, after that used Google Collab with faster GPU

Ввод [3]:
```
import uuid
import os
import time
```

Ввод [32]:
```
IMAGES_PATH = os.path.join('YOLO/data', 'images')
labels = ['awake', 'drowsy']
number_imgs = 20
```

Ввод [33]:
```
cap = cv2.VideoCapture(1)

for label in labels:
    print('Collecting images for {}'.format(label))
    time.sleep(5)

    for img_num in range(number_imgs):
        print('Collecting images for {}, image number {}'.format(label, img_num))

        ret, frame = cap.read()

        imgname = os.path.join(IMAGES_PATH, label+'.'+str(uuid.uuid1())+'.jpg')
        cv2.imwrite(imgname, frame)
        cv2.imshow('Image Collection', frame)
        time.sleep(2)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
cap.release()
cv2.destroyAllWindows()
cv2.waitKey(1)
```

```
Collecting images for awake
Collecting images for awake, image number 0
```

Step 1: Install Ultralytics library
!pip install ultralytics==8.0.196

Step 2: Set the current working directory
import os
HOME = os.getcwd()
print(HOME)

Step 3: Clear output in IPython display
from IPython import display
display.clear_output()

Step 4: Import YOLO module from Ultralytics
from ultralytics import YOLO

Step 5: Load YOLOv8 model and make a prediction
model = YOLO(f'{HOME}/yolov8n.pt')
results =
model.predict(source='https://media.roboflow.com/notebooks/examples/dog.jpeg', conf=0.25)

Step 6: Change directory and install Roboflow library

```
%cd {HOME}
!pip install roboflow

# Step 7: Set up Roboflow API and download YOLOv8 dataset
from roboflow import Roboflow
rf = Roboflow(api_key="rn6ihiuQcsvlnZflUNcO")
project =
rf.workspace("hof-university-of-applied-sciences-5x05r").project("unocards-z8f0f"
)
dataset = project.version(3).download("yolov8")

# Step 8: Train YOLOv8 model
%cd {HOME}
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml
epochs=25 imgsz=640

# Step 9: Display confusion matrix image
from IPython.display import Image
image_path = "/content/runs/detect/train/confusion_matrix.png"
Image(filename=image_path, width=600)

# Step 10: Display results image
image_path = "/content/runs/detect/train/results.png"
Image(filename=image_path, width=600)

# Steps 11-13: Additional training, validation, and prediction steps
image_path = "/content/runs/detect/train/val_batch0_pred.jpg"
Image(filename=image_path, width=600)

!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt
data={dataset.location}/data.yaml

!yolo task=detect mode=predict
model=/content/runs/detect/train/weights/best.pt conf=0.25
source={dataset.location}/test/images

# Step 14: Display multiple prediction images
import glob
from IPython.display import Image, display
for image_path in glob.glob(f'{HOME}/runs/detect/predict/*.jpg')[:-1]:
    display(Image(filename=image_path, width=600))
    print("\n")

# Steps 15-18: Repeat training, validation, and prediction with different models
!yolo task=detect mode=predict
model=/content/runs/detect/train2/weights/best.pt conf=0.25
source=/content/IMG_5432.MOV
```

... # Repeat similar steps until train7

%cd {HOME}
!yolo task=detect mode=train model=/content/runs/detect/train7/weights/best.pt data={dataset.location}/data.yaml epochs=10 imgsz=640

!yolo task=detect mode=val model=/content/runs/detect/train8/weights/best.pt data={dataset.location}/data.yaml

!yolo task=detect mode=predict model=/content/runs/detect/train8/weights/best.pt conf=0.25 source={dataset.location}/test/images

Step 19: Display final results image
image_path = "/content/runs/detect/train8/results.png"
Image(filename=image_path, width=600)

Step 20: Create a zip file of the content
!zip -r /content/content.zip /content

Step 21: Download the zip file
from google.colab import files
files.download('/content/content.zip')