```
from google.colab import drive

drive.mount('/content/drive')

%cd /content/drive/MyDrive/new_driver_drowsiness_project

!git clone https://github.com/ultralytics/yolov5

%cd yolov5

%pip install -qr requirements.txt comet_ml

import torch

import utils

display = utils.notebook_init()

!python train.py --img 640 --batch 32 --epochs 100 --data
/content/drive/MyDrive/new_driver_drowsiness_project/yolov5/data/custom.yaml --weights
yolov5s.pt --cache

!python detect.py --source
/content/drive/MyDrive/new_driver_drowsiness_project/yolov5/runs/train/exp/frame_0007.jpg --
weights
/content/drive/MyDrive/new_driver_drowsiness_project/yolov5/runs/train/exp/weights/best.pt

import torch

import pathlib

temp = pathlib.PosixPath

pathlib.PosixPath = pathlib.WindowsPath

import torch

import pathlib

temp = pathlib.PosixPath

pathlib.PosixPath = pathlib.WindowsPath

model = torch.hub.load(r"C:\Users\manga\driver_drowsy_project_\yolov5", 'custom',


path=r"C:\Users\manga\driver_drowsy_project_\yolov5\runs\train\exp\weights\best.pt",

                source='local',

                force_reload=True)

import matplotlib.pyplot as plt

import numpy as np

results = model(r"C:\Users\manga\driver_drowsy_project_\Dataset\train\images\frame_0004.jpg")

results.print()
```

```
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
import matplotlib.pyplot as plt
import numpy as np
results = model(r"C:\Users\manga\driver_drowsy_project_\Dataset\train\images\frame_0331.jpg")
results.print()
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
import matplotlib.pyplot as plt
import numpy as np
results = model(r"C:\Users\manga\driver_drowsy_project_\Dataset\train\images\frame_0378.jpg")
results.print()
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
!pip install torch
import torch
import pathlib
temp = pathlib.PosixPath
pathlib.PosixPath = pathlib.WindowsPath
%cd  yolov5
model = torch.hub.load(
    '',
    'custom',
    path=r"C:\Users\manga\driver_drowsy_project_\yolov5\runs\train\exp\weights\best.pt",
    source='local',
    force_reload=True
)
print(model)
```

```
import torch
import cv2
import numpy as np
import time
import winsound
import warnings
warnings.filterwarnings("ignore")
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video1.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 2
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
```

```
        break
    frame_count += 1
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f"🚨 ALERT! Driver is {class_labels[cls_id]} for more than 2 seconds! 🚨")
                winsound.Beep(1000, 500)
        else:
            distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
```

```
cv2.destroyAllWindows()
import torch
import cv2
import numpy as np
import time
import winsound
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video2.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 2
class_labels = {
   0: "Call Detection",
   1: "Focused",
   2: "Distracted",
   3: "Drowsy"
}

alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
   ret, frame = cap.read()
   if not ret:
      break
```

```
    frame_count += 1
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f"🚨 ALERT! Driver is {class_labels[cls_id]} for more than 2 seconds! 🚨")
                winsound.Beep(1000, 500)
        else:
            distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

```
import torch
import cv2
import numpy as np
import time
import winsound
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video2.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 2
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
```

```
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f"🚗 ALERT! Driver is {class_labels[cls_id]} for more than 2 seconds! 🚗")
                winsound.Beep(1000, 500)
        else:
            distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

```
import torch
import cv2
import numpy as np
import time
import winsound
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video3.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 10
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    if frame_count % frame_skip != 0:
```

```
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f"🚨 ALERT! Driver is {class_labels[cls_id]} for more than 10 seconds! 🚨")
                winsound.Beep(1000, 500)
        else:
            distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```