In the video, you can talk about functional modules you are in-charge of, also talk about things you are contributing to the overall project.

* Explain overall project
* My role in team as organizer meeting(as I hame unlimited zoom access), sending when2meet link to find available time, recording of meetings, gathering at first as I know everyone individually team which is why I was leader at first but I passed it to SVEN
* Explain the functional mode you did last time(video processing flowchart) and this time object detection

**1. Objective**

The goal was to train a YOLOv5 model on a custom dataset to detect room objects such as beds, chairs, doors, and tables, and use the model to identify objects in real-time using a camera.

**2. Actions Taken**

**Step 1: Setting up YOLOv5**

* **Installed YOLOv5 dependencies** using pip:

bash

Копировать код

pip install -r requirements.txt

**Challenges:**

* + The pip version was outdated. It was updated using:

bash

Копировать код

pip3 install --upgrade pip

**Step 2: Downloading Images**

* **Installed the bing-image-downloader package** for bulk downloading images:

bash

Копировать код

pip install bing-image-downloader

* **Downloaded images for categories like bed, table, door, etc.:**

python

Копировать код

from bing\_image\_downloader import downloader

categories = ['bed', 'chair', 'desk', 'lamp', 'table', 'wardrobe']

for category in categories:

downloader.download(category, limit=100, output\_dir='dataset', adult\_filter\_off=True, force\_replace=False, timeout=60)

**Challenges:**

* + SSL certificate error while downloading images.
  + Fixed by disabling SSL verification in the bing\_image\_downloader script.

**Step 3: Organizing Dataset**

* Created train and val folders to organize the dataset.
* **Split downloaded images into train (80%) and val (20%):**

python

Копировать код

import os

import shutil

from sklearn.model\_selection import train\_test\_split

categories = ['bed', 'chair', 'desk', 'lamp', 'table', 'wardrobe']

for category in categories:

images = os.listdir(f'dataset/{category}')

train\_images, val\_images = train\_test\_split(images, test\_size=0.2, random\_state=42)

os.makedirs(f'dataset/train/{category}', exist\_ok=True)

os.makedirs(f'dataset/val/{category}', exist\_ok=True)

for img in train\_images:

shutil.move(f'dataset/{category}/{img}', f'dataset/train/{category}/{img}')

for img in val\_images:

shutil.move(f'dataset/{category}/{img}', f'dataset/val/{category}/{img}')

**Step 4: Annotating the Dataset**

* **Tried labelImg for manual annotation:**
  + Installed labelImg:

bash

Копировать код

pip install labelImg

* + Encountered issues with PyQt5 installation and fixed them using:

bash

Копировать код

brew install python3

pip install pyqt5

* Annotated images from both train and val datasets.
* Exported annotations from **Roboflow** as a YOLO-formatted dataset in a ZIP file.

**Step 5: Preparing Data for YOLOv5**

* Extracted the ZIP file exported from Roboflow and organized it into the YOLOv5 dataset/ folder:

kotlin

Копировать код

yolov5/

├── dataset/

├── train/

│ ├── images/

│ └── labels/

├── val/

├── images/

└── labels/

* Edited the data.yaml file to reflect the dataset structure:

yaml

Копировать код

train: dataset/train/images

val: dataset/val/images

nc: 6 # Number of classes

names: ['bed', 'chair', 'desk', 'lamp', 'table', 'wardrobe']

**Step 6: Training the Model**

* Ran the YOLOv5 training script:

bash

Копировать код

python3 train.py --img 640 --batch 16 --epochs 50 --data dataset/data.yaml --weights yolov5s.pt

**Challenges:**

1. **File Not Found (data.yaml):**
   * Resolved by ensuring the correct path to data.yaml and matching folder structure.
2. **SSL Certificate Error (Arial.ttf font):**
   * Fixed by manually creating the required directory and placing the Arial.ttf font:

bash

Копировать код

mkdir -p ~/Library/Application\ Support/Ultralytics/

mv Arial.ttf ~/Library/Application\ Support/Ultralytics/

**Step 7: Next Steps**

* **Real-time Detection:** Once training is complete, you will:
  + Use the trained weights (runs/train/exp/weights/best.pt) to test the model on a live camera feed.
  + Script to use the camera:

python

Копировать код

import torch

from yolov5.utils.general import check\_img\_size

from yolov5.models.common import DetectMultiBackend

from yolov5.utils.torch\_utils import select\_device

from yolov5.utils.datasets import LoadStreams

from yolov5.utils.plots import Annotator

model = torch.hub.load('ultralytics/yolov5', 'custom', path='runs/train/exp/weights/best.pt')

cap = cv2.VideoCapture(0)

while cap.isOpened():

ret, frame = cap.read()

results = model(frame)

results.render()

cv2.imshow('YOLOv5 Detection', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

**Challenges Faced**

1. **SSL Certificate Issues:**
   * Fixed by disabling SSL verification for downloads and manually creating required directories.
2. **Dataset Preparation:**
   * Ensured train/val split and annotation format match YOLOv5 requirements.
3. **Labeling:**
   * Initially faced issues with labelImg but switched to Roboflow for faster and more efficient annotation.
4. **Training Errors:**
   * Fixed by correcting paths in data.yaml and downloading missing files like Arial.ttf.

**Conclusion**

You have successfully set up YOLOv5, prepared a custom dataset, annotated images, and initiated training. Once training is complete, the model will be ready for real-time object detection. Let me know if you need further assistance!