**ממשק בין אפליקציית מב"ת והטכניון**

1. כללי:
   1. מסמך זה מפרט את צורת ההפעלה של אפליקציית צוות הטכניון באמצעות אפליקציית מב"ת, מבחינת לוגיקת הפעלה, קלט ופלט.
   2. לצורך השתלבות במערכת, צוות הטכניון יספק את האפליקציה בתור מודול פייתון שמממש מחלקה (class), רצוי בתור package להתקנה.
   3. צוות הטכניון אחראי לתאימות המודול לסביבת העבודה בכרטיס Jetson Orin NX, מבחינת גרסת פייתון ו-dependencies, וכן לעמידה בזמני ריצה סבירים של עד 100ms למסגרת (TBR) בסביבה זו בעומס מעבד נומינלי.
   4. הממשק לאפליקציית העיגון נפרד מהממשק המפורט במסמך זה.
   5. מימוש המודול ואופן השימוש:

האפליקציה תימסר כמודול פייתון, המממש מחלקה בשם ScanManager.  
המחלקה תכיל שלוש פונקציות ציבוריות (public methods) אשר מממשות את שלושת מצבי ההפעלה של המערכת, כמפורט להלן:

* + - אתחול הclass:

detector = ScanManager()

* + - ביצוע מיפוי שטח ראשוני:

phase0(frame: np.ndarray, metadata: dict) -> None

* + - ביצוע סריקת שטח (Phase 1):

phase1(frame: np.ndarray, metadata: dict) -> list[dict]

מחזירה רשימת תוצאות כמפורט בסעיף 3.

* + - ביצוע חקירת נקודה (Phase 2):

phase2(frame: np.ndarray, metadata: dict) -> dict

מחזירה תוצאת חקירה בודדת כמפורט בסעיף 4.

דוגמא להפעלת הפונקציות:

from wildfire\_detector.function\_class import ScanManager

detector = ScanManager(config\_path="config.yaml") # config\_path is optional

detector.phase0(frame, metadata)

results\_phase1 = detector.phase1(frame, metadata)

results\_phase2 = detector.phase2(frame, metadata)

1. לוגיקת ההפעלה:
   1. אפליקציית הטכניון תופעל בשלוש דרכים בהתאם למוד ההפעלה של המערכת:
      * מיפוי שטח ראשוני (שלב מקדים) – מול תמונה תרמית.
      * סריקת שטח (phase 1) – מול תמונה תרמית.
      * חקירת נקודה (phase 2) – מול תמונת צבע.

מאחר שדרכי ההפעלה נפרדות זו מזו, רצוי שיופעלו בתור פ פונקציות נפרדות.

* 1. הפעלת מוד חקירה תהיה בתום סריקה (עבור מספר נקודות חשודות ברמה נמוכה), או באמצע סריקה (עבור נקודה חשודה ברמה גבוהה).
  2. החקירה תתבצע במצלמת הצבע, בשדה ראייה צר, כאשר שדה הראייה המבוקש לחקירה ייקבע ע"י אפליקציית הטכניון בשלב הסריקה ויישלח כחלק מהפלט של פונק' הסריקה.

1. פונק' מיפוי שטח ראשוני וסריקת שטח (phase 1):
   1. קלט:

**Frame** – numpy ndarray, (720, 1280), dtype=np.uint8

**Metadata** – dict with the following structure (example, TBR):

metadata = {

"uav": {

"altitude\_agl\_meters": 2400.0,

"roll\_deg": 0.5,

"pitch\_deg": -1.2,

"yaw\_deg": 45.0,

},

"payload": {

"pitch\_deg ": -12.0,

"azimuth\_deg ": 128.0,

"field\_of\_view\_deg ": 18.0,

"resolution\_px": [1280, 720],

},

"geolocation": {

"latitude": 31.0461,

"geomapping\_function": TBD,

"longitude": 34.8516,

},

"scan\_parameters": {

"current\_scanned\_frame\_id": 35,

"total\_scanned\_frames": 173,

},

"timestamp": "2025-04-08T12:30:45.123Z", # ISO 8601 format

}

* 1. פלט – עבור פונק' סריקת שטח:

results – list of dictionaries, with the following structure (example, TBR):

results= [

{

"latitude": 31.0461,

"longitude": 34.8516,

"bounding\_box ": [top\_left\_lat, top\_left\_lon, bottom\_right\_lat, bottom\_right\_lon]

"confidence\_pct": 80.0,

"required\_fov\_deg ": 2.5

"current\_IR\_fov": 17.5

},

{

"latitude": 31.0421,

"longitude": 34.8316,

"bounding\_box ": [top\_left\_lat, top\_left\_lon, bottom\_right\_lat, bottom\_right\_lon]

"confidence\_pct": 50.0,

"required\_fov\_deg": 18.0

"current\_IR\_fov": 17.5

}

]

רשימה ריקה תוחזר במידה שלא נמצאו נקודות חשודות.

1. פונק' חקירת נקודה:
   1. קלט:

**Frame** – numpy ndarray, (1080, 1920, 3), dtype=np.uint8

**Metadata** – dict with the following structure (example, TBR):

metadata = {

"uav": {

"altitude\_agl\_meters": 2400.0,

"roll\_deg": 0.5,

"pitch\_deg": -1.2,

"yaw\_deg": 45.0,

},

"payload": {

"pitch\_deg ": -12.0,

"azimuth\_deg ": 128.0,

"field\_of\_view\_deg ": 2.5,

"resolution\_px": [1920, 1080],

},

"geolocation": {

"latitude": 31.0461,

"geomapping\_function": TBD,

"longitude": 34.8516,

},

"investigation\_parameters": {

"detection\_latitude": 31.0421,

"detection\_longitude ": 34.8516,

"detected\_bounding\_box ": [top\_left\_lat, top\_left\_lon, bottom\_right\_lat, bottom\_right\_lon]

},

"timestamp": "2025-04-08T12:30:45.123Z", # ISO 8601 format

}

* 1. פלט:

results – list of dictionaries, with the following structure (example, TBR):

results= [

{

"fire\_existence": 1, # 0 – no fire, 1 - fire

"latitude": 31.0461,

"longitude": 34.8516,

"bounding\_box ": [top\_left\_lat, top\_left\_lon, bottom\_right\_lat, bottom\_right\_lon]

"confidence\_pct": 90.0,

},

{

"fire\_existence": 1, # 0 – no fire, 1 - fire

"latitude": 31.0421,

"longitude": 34.8316,

"bounding\_box ": [top\_left\_lat, top\_left\_lon, bottom\_right\_lat, bottom\_right\_lon]

"confidence\_pct": 50.0,

}

]