

# **המעבדה לבקרה רובוטיקה ולמידה חישובית**

## **דו"ח פרויקט מסכם**

**שם הפרויקט:** Solar panel mapping and localization

**סמסטר:** חורף תשפ"ג

**מנחה:** Koby Kohai

**סטודנטים:** Nadi Najjar & Luna Khoury

## תוכן

1. תקציר.....	3
2. מבוא.....	3
3. סביבת עבודה.....	4
4. תיאור אלגוריתם.....	4
5. סכמת מלבנים.....	5
6. תוצרים.....	10
7. מסקנות.....	10

## 1. תקציר

פרויקט זה משלב בין עולם הרובוטיקה ועולם עיבוד וניתוח תמונות תוך שימוש במערכות לומדות. כך שהרעיון המרכזי מאחורי פרויקט זה הינו שימוש ברובוט שכבר קיים כיום בשוק, ושדרוג אופן פעולתו על ידי כך שנתקין עליו מצלמה שכל מטרתה היא לדעת להגיד היכן הרובוט נמצא בכל עת ומהו מיקומו ביחס לסביבה כלשהי שהרובוט נמצא בה. קבלת מידע על הסביבה בה פועל הרובוט ועל מיקום הרובוט בזמן אמת הינו דבר חשוב ומועיל שבעזרתו נוכל לדעת לנתח כל מני תופעות ואף למנוע בעיות שעלולות לקרות בשטח, גם לרובוט עצמו. הכלי שנרצה לפתח הינו כלי שימושי וגנרי, וניתן יהיה להשתמש באלגוריתם שנפתח על כל מצלמה שתותקן על רובוט.

## 2. מבוא

בפרויקט זה נעסוק בתחומי עיבוד תמונה ומערכות לומדות על מנת שנוכל לבנות מפה דו-ממדית של הרובוט בשטח בעזרת המצלמה. כך שבעזרת עיבוד תמונה נוכל לנתח את התמונות שמתקבלות מהמצלמה ולקבל מהם מידע על מיקום הרובוט בשטח בכל זמן נתון, לאפיין ולזהות בעזרתם כל מני בעיות שעלולות להיווצר לרובוט בשטח או סביבו. בנוסף לכך, נצטרך מערכות לומדות למטרות יותר מתקדמות, בעיקר בשלבי פיתוח הקוד שבעזרתו נשלוט על המצלמה ולנצח בעזרתו תוצאות המידע שנקבל. בנוסף לכך נרצה להיעזר העקרונות מערכות לומדות בעיקר למטרת שיפור פעולת המצלמה וייעול המערכת בכללי, כך שעם הזמן המצלמה תהיה יותר "חכמה" ותדע לבד ללמוד מתוצאות ניתוחים קודמים כדי לדייק בתוצאות המתקבלות ובכך נקבל מפה יותר מדויקת ומפורטת.

נכון להיום רובוטים נמצאים בשטח ואין לנו מידע עליהם בזמן אמת איפה הם נמצאים בתוך השטח. ולכן מטרת הפרויקט הינה לבנות מפה דו-ממדית לכל רובוט שנמצא בשטח בעזרת מצלמה שתותקן עליו, שתפעל כל הזמן ותשדר תמונות לשרת. תמונות אלה יעברו ניתוח ועיבוד תמונה בעזרת אלגוריתם שאנחנו ניפתח במהלך הפרויקט, שבעזרתו נוכל לנתח מידע מהתמונות ולקבל את מיקום הרובוט ואיפה הוא נמצא ביחס לשטח בכל זמן נתון ובכך נוכל לבנות מפה לכל רובוט. המטרה הינה לבנות מפה כמה שיותר מדויקת כך שנוכל לקבל מידע בכל עת על מיקום הרובוט בשטח, לשם כך אני נפתח את האלגוריתם שלנו בעזרת מערכות לומדות כדי לשפר ולייעל את פעולת ניתוח המידע, בעזרת כך המצלמה עם הזמן תוכל ללמוד את הסביבה להשתמש במידע שכבר יש לה על מנת לדייק יותר ולקבל תוצאות מדויקות ככל שניתן.

### **3. סביבת עבודה**

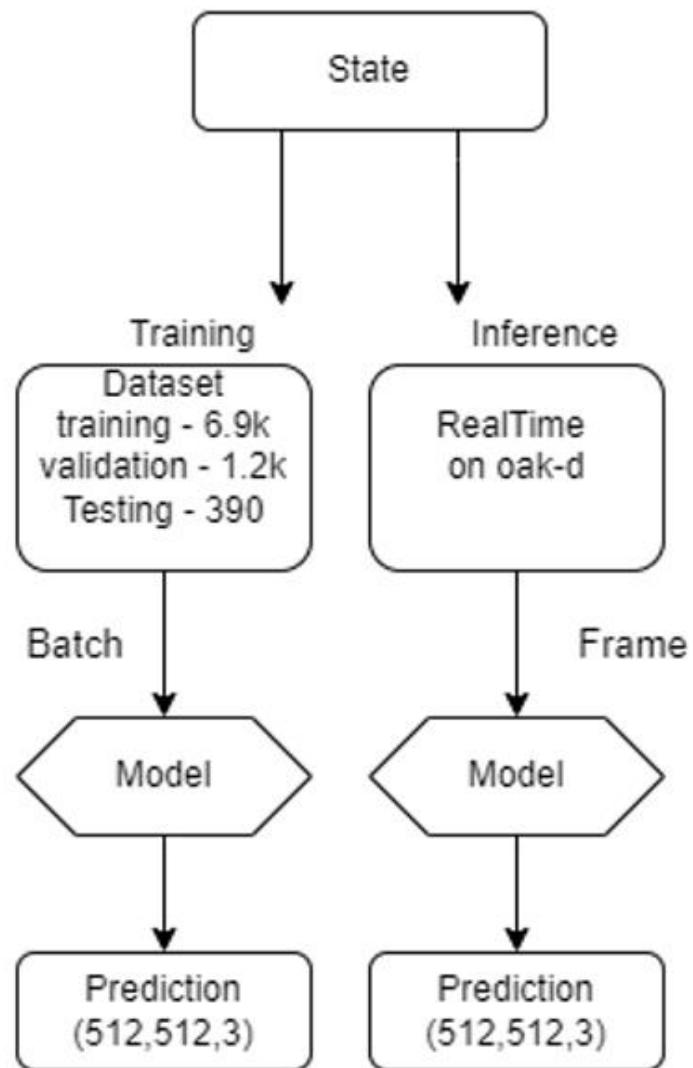
הפרויקט כולל מצלמת OAK-D Lite שתותקן על הרובוט שנמצא בשטח ותהיה מחוברת למחשב ותשדר אליו את התמונות. בשלבי פיתוח הפרויקט עבדנו עם PyCharm וכתבנו את הקוד בשפת python שהוא רץ על המצלמה.

### **4. תיאור אלגוריתם**

האלגוריתם שנפתח מטרתו העיקרית הינה לדעת לבנות מפה לרובוט מהתמונות שהוא מקבל מהמצלמה. אלגוריתם זה בתור התחלה יקבל את התמונות ויבצע סגמנטציה לשידור בזמן אמת, ינתח את המידע שהתקבל, יאסוף אותו במהלך הרצה וסריקה ראשונית של שטח העבודה של הרובוט על מנת לבנות את הסגמנטציה בזמן אמת.

לאחר מכן, נשתמש בעיבוד תמונה בשביל לזהות מעברי פאנלים ואז נדע איפה הרובוט נמצא ביחידות פאנלים מנקודת ההתחלה. בעזרת עקרונות מערכת לומדת האלגוריתם שלנו עם הזמן ועם המידע שכבר יש לו, הוא ידע ללמוד את הסביבה בה הרובוט חיי, ובהתאם הוא ידייק יותר את המפה שכבר ייצר.

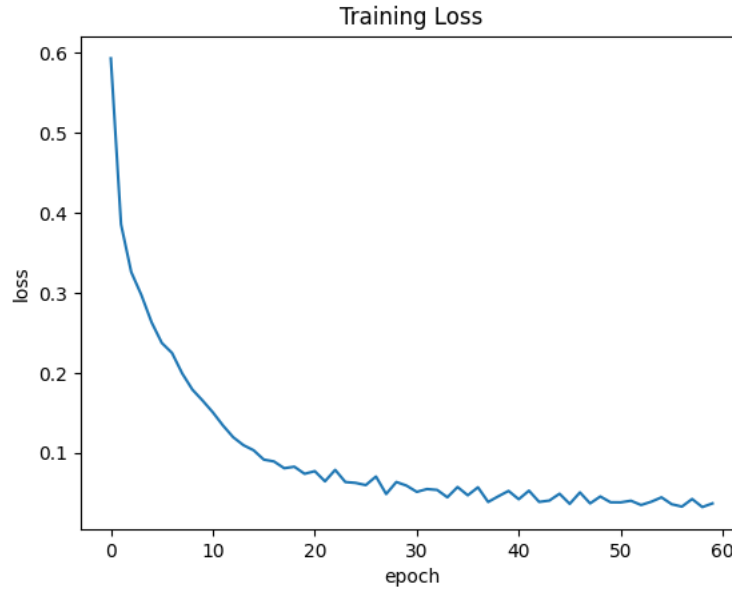
## 5. סכמת בלוקים



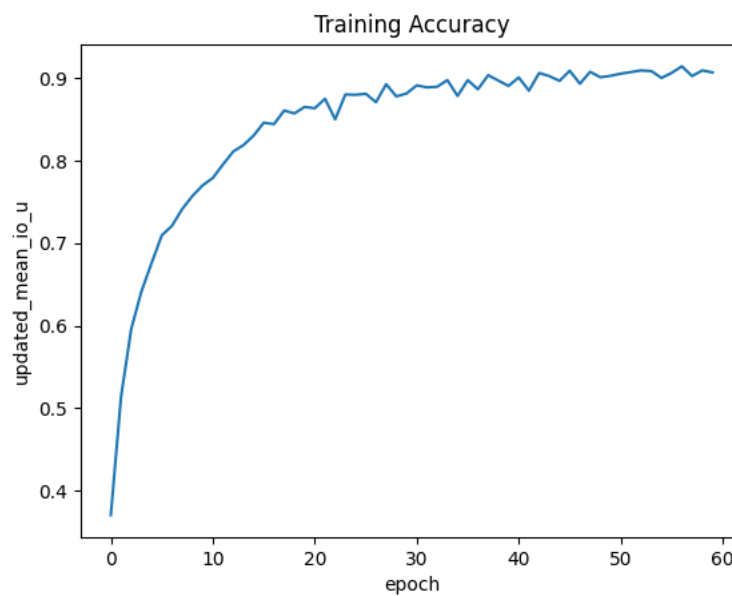
## 6. ניתוח ביצועי המודלים

- Unet16 Architecture – Image size: 256

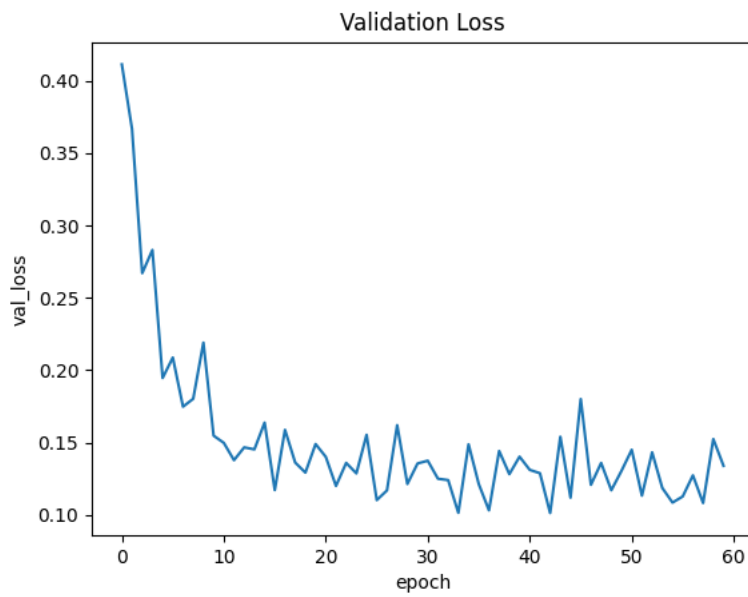
### Training Loss graph –



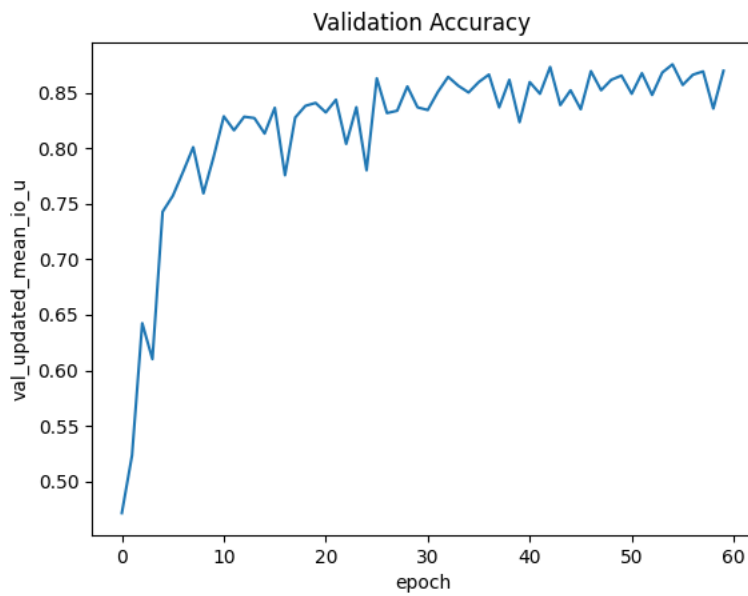
### Training Accuracy graph –



### **Validation Loss graph –**

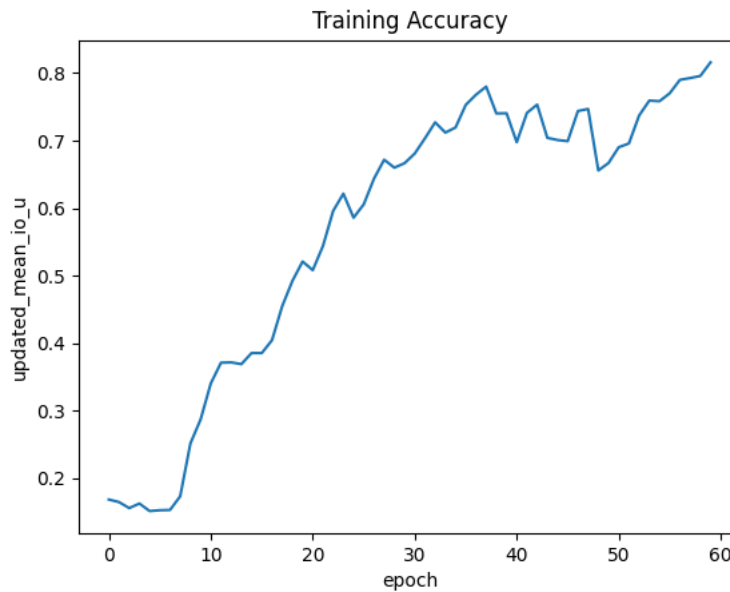


### **Validation Accuracy graph –**

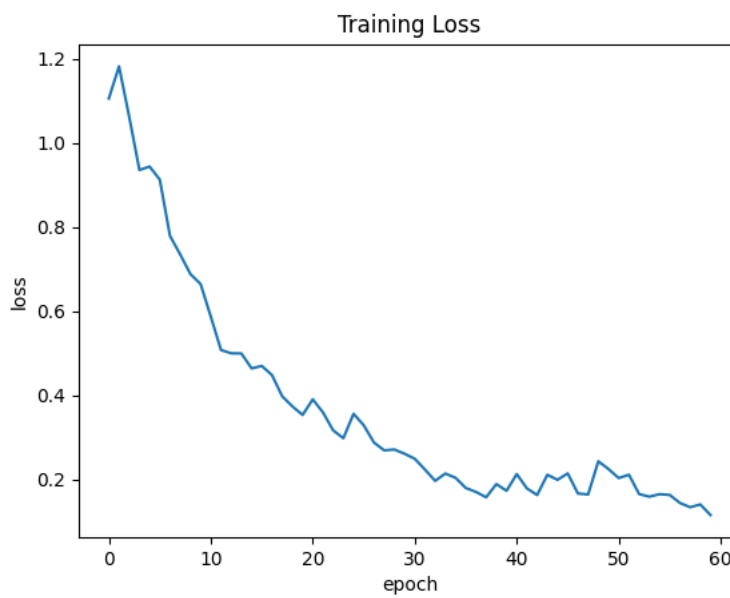


- DeeplabV3+ Architecture – Image size: 256

### **Training Loss graph –**

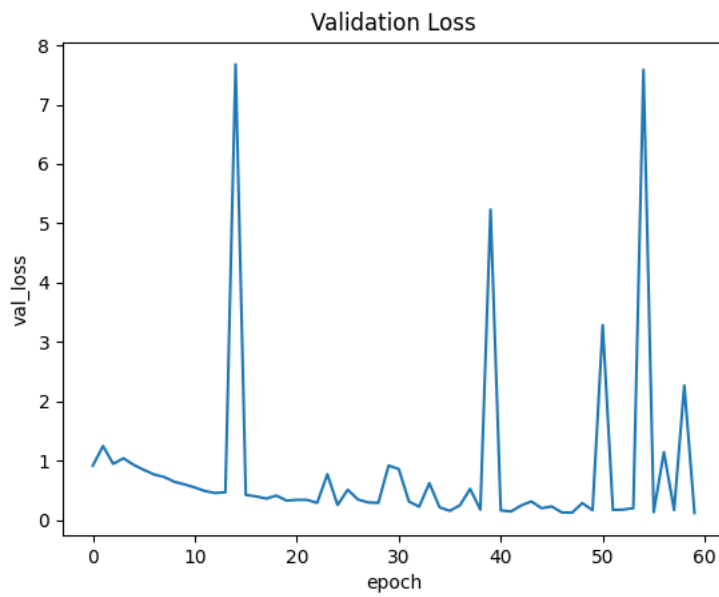


### **Training Accuracy graph –**

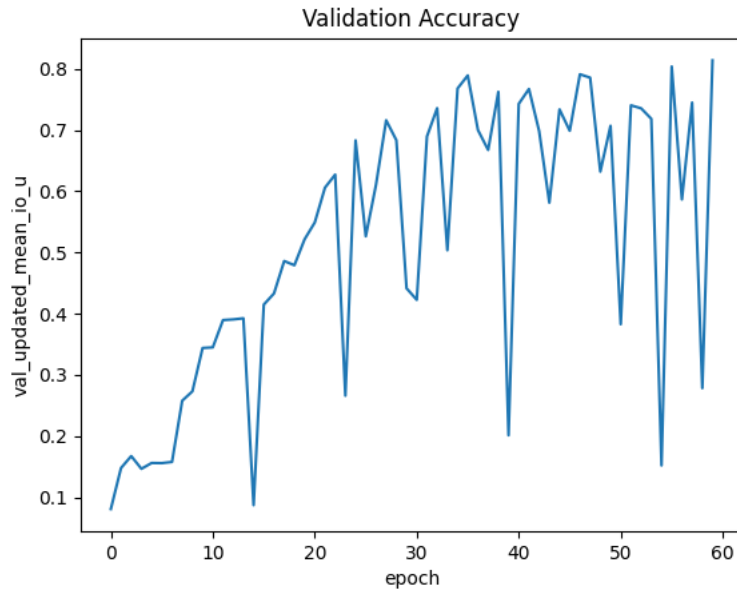




### Validation Loss graph –

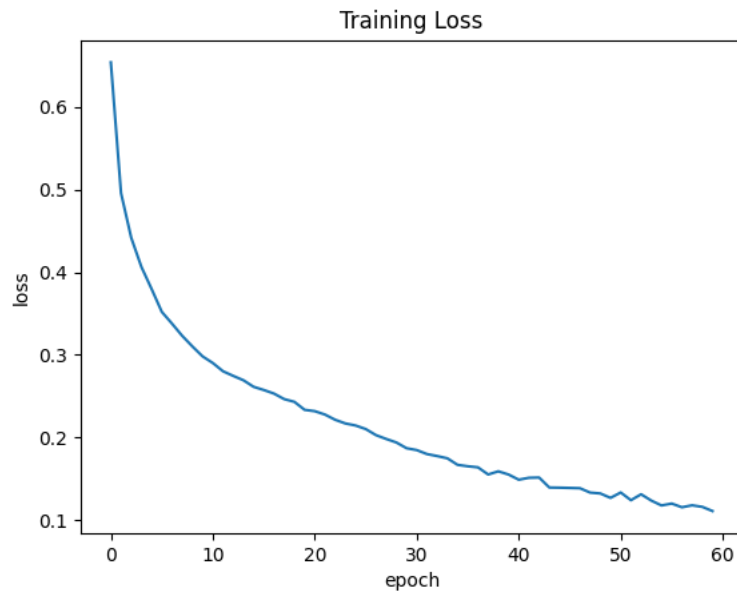


### Validation Accuracy graph –

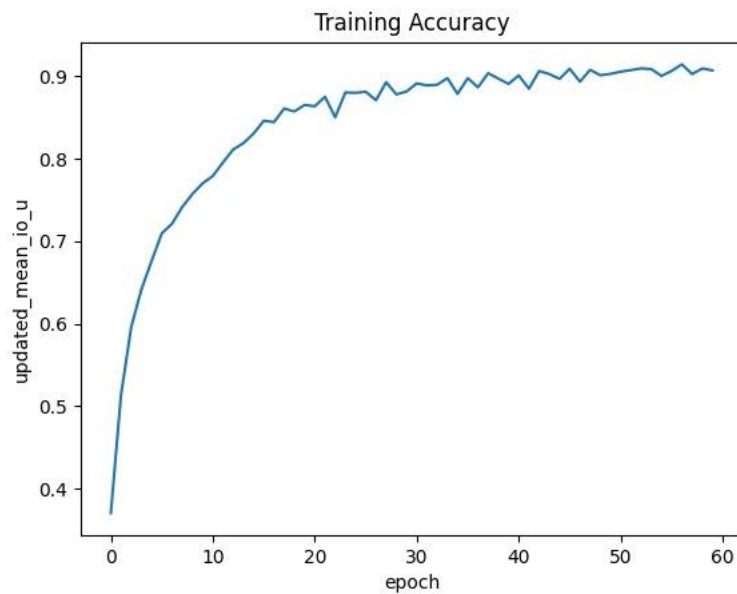


- Unet16 Architecture – Image size: 1024

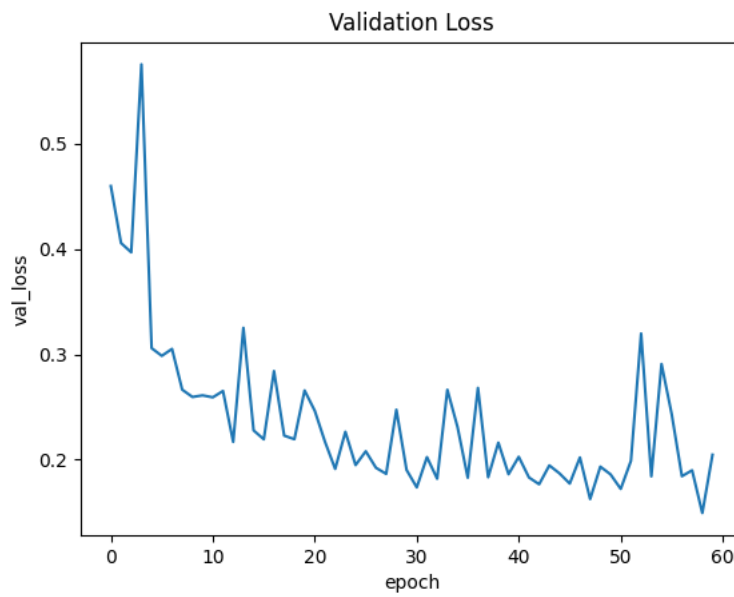
### **Training Loss graph –**



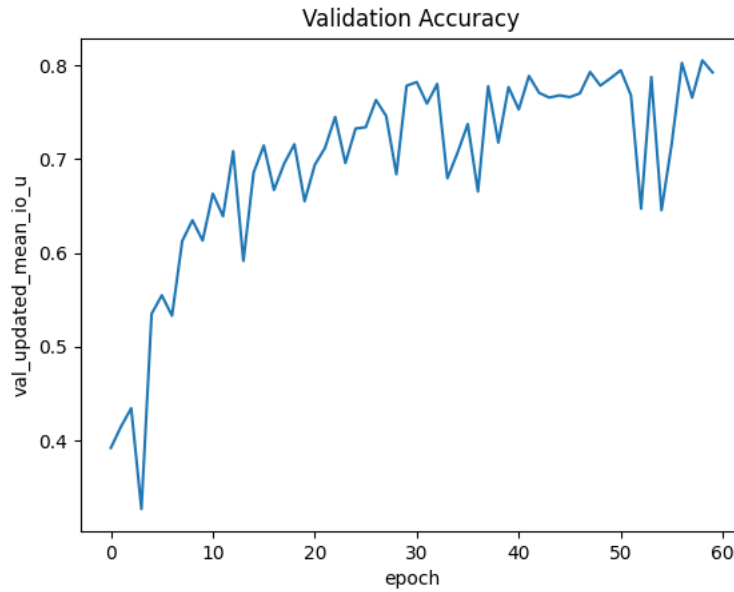
### **Training Accuracy graph –**



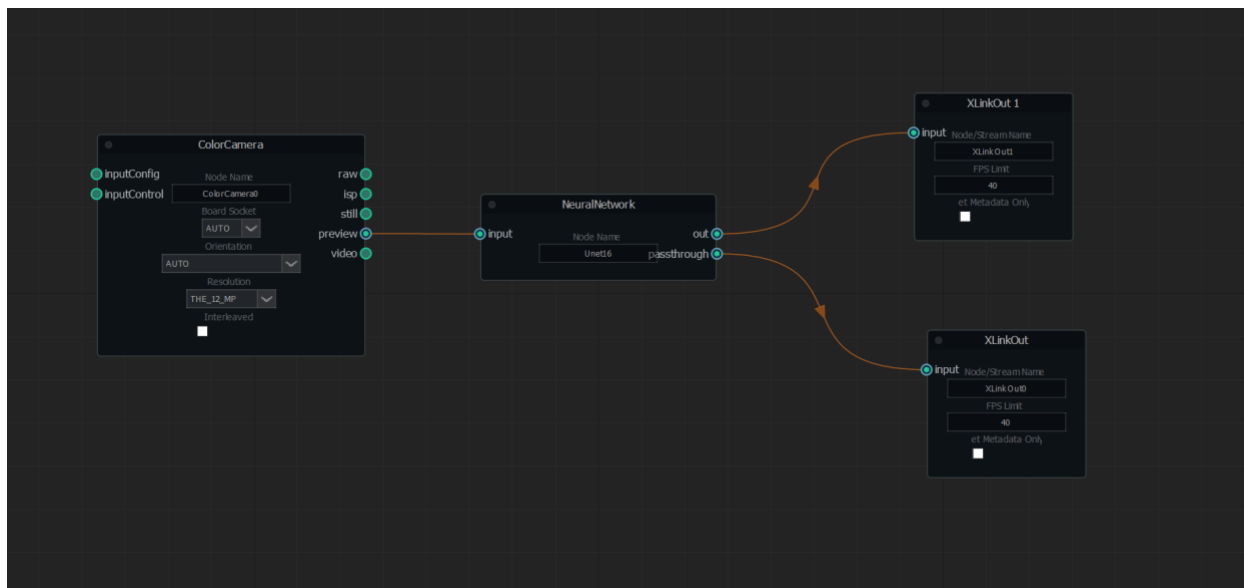
### **Validation Loss graph –**



### **Validation Accuracy graph –**



## 7. ארכיטקטורת המצלמה "Oak-d lite"



## 8. אופן פעולת המערכת

הגדרת אופן הפעולה המלא של המערכת מצורפת בקובץ **README** שנמצא בגיט, נסביר כאן באופן כללי:

- שלב ראשון: הכנת ה-Dataset, נטען אותו לקוד האימון Training optimization, שישפר את חיזויי המודל. לאחר הריצה קוד זה שומר את המודל בפורמט מתאים כדי לטעון אותו למצלמה.
- שלב שני: טעינת המודל בפורמט המתאים לקוד של ה-inference ולוודות שהמצלמה מחוברת למחשב ואז הכל פעולת המודל תתחיל באופן דינמי וה-inference יתחיל לרוץ בחומרה של המצלמה.

## 9. Challenges:

- Dataset annotation

מכיוון שאין datasets מוכנים לאובייקט של הפאנל הסולארי ושאר האובייקטים הקיימים בשדה שבו עובד הרובוט, היינו צריכים לאסוף מלא תמונות (אספנו בסביבות 1000 תמונה). לצורך כך, הלכנו לשטח שבו עובדים הרובוטים, וצילמנו בעזרת המצלמה שאנו משתמשים בה בפרויקט, תמונות בזמן אמת ואז העברנו אותם למחשב וכדי להוציא מהם את ה-dataset שרצינו ביצענו להם באופן ידני annotation (באתר Roboflow.com) לפי סוגי האובייקטים הקיימים:  
Home, Bridge, Solar Panel, Background

#### • Background mask creation

היה לנו בעיה בהתחלה שהוא לא מזהה את הרקע בצורה נכונה, לאחר כמה ניסויים כדי להבין מה קרה אז הגענו למסקנה שלא היינו צריכים לסמו את הרקע ב- annotation כאובייקט בפני עצמו כי זה גרם למצלמה לחשוב שהרקע אמור להיות מצורה כלשהי, אלה היה צריך לסמן את כל האובייקטים, ומה שנשאר המצלמה תתייחס אליו כרקע.

## 10. תוצרים

בסופו של דבר נרצה להתקין את המצלמה על הרובוט שמבצע mapping לשטח, למטרת בחינת התוצאות של האלגוריתם נוכל לבדוק את זה בכל שטח שנגדיר לרובוט, למשל סביבת המעבדה. והמטרה היא שבעזרת מצלמה זו נוכל לקבל מידע בזמן אמת על הרובוט מבחינת איפה הוא נמצא בכל עת, איך נראה השטח סביבו, לקבל מידע מתוך התמונות שהוא ישדר בעזרת המצלמה כך שבעזרתם נוכל לזהות איפה נמצא הרובוט, והאם התוצאות שהתקבלו אכן תואמות למציאות.