הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל ע"ש אנדרו וארנה ויטרבי

המעבדה לבקרה, לרובוטיקה וללמידה חישובית

**ספר פרויקט**

**מצלמה ליד תותבת**

מבצעים:

נוואף סלמאן Nawaf Salman

נאהל עוידאת Nahel Awidat

מנחה:

קובי קוחי Koby Kohai

סמסטר רישום: אביב תשפ"ג

תאריך הגשה:

**תוכן**

[**תקציר** 2](#_Toc191742491)

[**Abstract** 3](#_Toc191742492)

[**Introduction** 4](#_Toc191742493)

[**Technologies Used** 5](#_Toc191742494)

[**Haifa 3D Robotic Hand:** 8](#_Toc191742495)

[**General Architecture:** 9](#_Toc191742496)

[**Complete Implementation:** 11](#_Toc191742497)

[**Failed Attempts:** 12](#_Toc191742498)

[**Possible Future Expansions:** 13](#_Toc191742499)

[**References** 14](#_Toc191742500)

# **תקציר**

האפשרות לשלב את היכולת לנצל את העוצמה של בינה מלאכותית ולמידת מכונה ברובוטיקה לצורך הבנת נתונים נכנסים מהסביבה, באמצעות חיישנים ומצלמות מסוגים שונים – כפי שנעשה, למשל, בסגמנטציה של תמונות וחפצים – מצביעה על כך שאנו צועדים לעבר עתיד של תלות גוברת במכשירים חכמים ומחוברים בחיי היומיום שלנו.  
מטרת הפרויקט היא לתכנן ולפתח **יד רובוטית אוטונומית** המסוגלת להסתובב ולאחוז בחפצים כדי לסייע לקטועי ידיים. זהו דוגמה מצוינת לאופן שבו רובוטיקה ובינה מלאכותית יכולים לתרום לחברה. על ידי אוטומציה של פעולת האחיזה, היד הרובוטית מפחיתה את העומס הפיזי על קטועי ידיים ומאפשרת להם לחיות ולתפקד בקלות רבה יותר ביום-יום. הפרויקט מדגים את הפוטנציאל של רובוטיקה ובינה מלאכותית לשיפור איכות החיים של אנשים עם מוגבלויות, ומדגיש את החשיבות של פיתוח והטמעה אחראיים של טכנולוגיות אלו.  
היד הרובוטית, אשר פותחה על ידי עמותת **Haifa3D**, מצוידת במצלמת **Oak-D Lite** ומופעלת על ידי **מיקרו-בקר Jetson Nano**. ה-Jetson Nano מעבד את הנתונים המתקבלים מהמצלמה ומפעיל מודל סגמנטציה לזיהוי חפצים, קביעת כיוונם ומרחקם. לאחר שהאובייקט מזוהה, ה-Jetson Nano שולח פקודות ליד הרובוטית כדי לסובב אותה לזווית מתאימה או לאחוז בחפץ.  
היד הרובוטית נועדה להחליף את היד הקטועה, והפרויקט הזה מהווה צעד משמעותי לקראת השבת תפקודיות ועצמאות לקטועי ידיים, תוך שיפור איכות חייהם באמצעות טכנולוגיית פרוסתטיקה מתקדמת.

# **Abstract**

The prospect of integrating the ability to utilize the power of AI and machine learning into robotics for understanding incoming data collected from the surrounding environment by providing them with various types of sensors and cameras, as is the case in image and object segmentation for example, means that we are heading towards a future of even greater dependency on intelligent connected devices in our daily lives.   
The aim of this project is to design and develop an autonomous robotic hand capable of rotating and grabbing objects to help hand amputees. This is an excellent example of how robotics and AI can be used to benefit society. By automating the task of grabbing objects, the robotic hand reduces the physical burden on hand amputees and allows them to live and function easier daily. This project demonstrates the potential of robotics and AI to improve the quality of life for people with disabilities and highlights the importance of responsible development and deployment of these technologies.   
The robotic hand, which was made by Haifa3D association and the Oak-D Lite camera mounted to it, are controlled by a Jetson Nano microcontroller. The Jetson Nano processes the input from the camera and utilizes a segmentation model to detect objects, their orientation and distance. Once the object is detected, the Jetson Nano sends signals to the robotic hand to either rotate to an appropriate angle or grab the object. The robotic hand is designed to substitute the amputated hand. This project is a step towards restoring functionality and independence for amputees, improving their quality of life through advanced prosthetic technology.

# **Introduction**

All the code of this project is open source and available for the general public on GitHub Yamen-Safadi/iRobot-Table-Tennis-ball-collector repository.

The project was suggested in the CRML department in the Technion. The goal is to automate a robotic 3D printed hand capable of rotating and grabbing objects to help hand amputees. While similar projects were done in the past, it had very different architecture and goals.

Similar projects built previously in the faculty:

• foot-controlled hand, shoulder sensors machine learning(R2G)...

Our project did 3 things differently:

1- Our robotic hand must be fully autonomous and there is no need for humans to give it commands its goal was predetermined.

2- The robotic hand had to interact with the real world differently since it can grab real objects on its own.

3- We supplied the robot with the ability to understand its surroundings using AI-based and non-AI-based algorithms for object segmentation and distance measuring.

# **Technologies Used**

**ערכת הפיתוח NVIDIA Jetson Nano:**

ערכת הפיתוח NVIDIA Jetson Nano היא מחשב חד-לוחיתי קומפקטי המיועד ליישומי בינה מלאכותית ורובוטיקה. היא כוללת מעבד גרפי NVIDIA Maxwell ומסוגלת להריץ רשתות נוירונים מרובות במקביל. ערכת הפיתוח מהווה פלטפורמה חסכונית לפיתוח ובדיקת פרויקטים בתחום הבינה המלאכותית, ומתאימה ליוצרים, סטודנטים וחובבים.

הערכה מבוססת על Jetson Nano NVIDIA מערכת-על-מודול, כאשר ה-GPU שלה כולל 128 ליבות CUDA. בנוסף, היא כוללת מעבד מרובע ליבות ARM A57, זיכרון LPDDR4 בנפח 4GB, חיבור רשת Gigabit Ethernet, כניסות USB 3.0, ויציאת HDMI.

מלבד יכולות ה-AI שלה, ערכת הפיתוח Jetson Nano כוללת גם ממשקי GPIO ו-UART, המאפשרים שילוב בפרויקטים רובוטיים מגוונים.

**NVIDIA Jetson Nano Developer Kit:**The NVIDIA Jetson Nano Developer Kit is a compact single-board computer designed for AI and robotics applications. It features an NVIDIA Maxwell GPU and is capable of running multiple neural networks in parallel. The developer kit is a low-cost platform for developing and testing AI projects, and is suitable for makers, students and hobbyists. It is based on NVIDIA Jetson Nano System-in-Module and its GPU includes 128 CUDA cores. Also includes a quad-core ARM A57 CPU, 4 GB LPDDR4 memory, Gigabit Ethernet, USB 3.0 and HDMI.

In addition to its AI capabilities, the Jetson Nano Developer Kit is also equipped with GPIO and UART interfaces that make it suitable for a wide range of robotic projects.

A close-up of a computer chip

AI-generated content may be incorrect.

בתמונה הקודמת ניתן לראות את המחשב המיניאטורי של NVIDIA (Jetson Nano). השקע שנמצא בפינה השמאלית התחתונה של הלוח הוא כניסת המתח – נדרש מתח DC של 5V עם לפחות 4 אמפר כדי להפעיל את המכשיר בצורה חלקה.

In the previous picture we can see the Nvidia mini computer (Jetson Nano). The jack on the lower left side of the board is the power supply input: 5V DC power with at least 4 amps is needed to smoothly power the device.

**מצלמת Oak-D Lite :**

בפרויקט השתמשנו במצלמת **Oak-D Lite** – מצלמה קטנה לזיהוי עומק, המיועדת ליישומי ראייה ממוחשבת. המצלמה פותחה על ידי חברת **Luxonis**, המתמחה בפיתוח כלי חומרה ותוכנה לראייה ממוחשבת.

המצלמה מצוידת ב-**Intel Movidius Myriad X VPU**  (יחידת עיבוד חזותי) ובחיישן עומק **Sony DepthSense IMX556PLR** הפועל בטכנולוגיית **ToF (Time of Flight)**. שילוב רכיבים זה מאפשר למצלמה ללכוד מידע עומק ולעבד אותו בזמן אמת, מה שהופך אותה לאידיאלית עבור יישומים כגון **זיהוי אובייקטים, מעקב והכרה חזותית**.

בנוסף, המצלמה כוללת **מגוון כלים ותוכנות לפיתוח**, כולל **Luxonis DepthAI Python API**, המאפשר למפתחים לשלב את המצלמה בקלות בפרויקטים ובאפליקציות שלהם. המצלמה תואמת גם למסגרות למידת מכונה פופולריות כגון **TensorFlow ו-PyTorch**, ומאפשרת יצירה ופריסה של מודלים מותאמים אישית לראייה ממוחשבת.

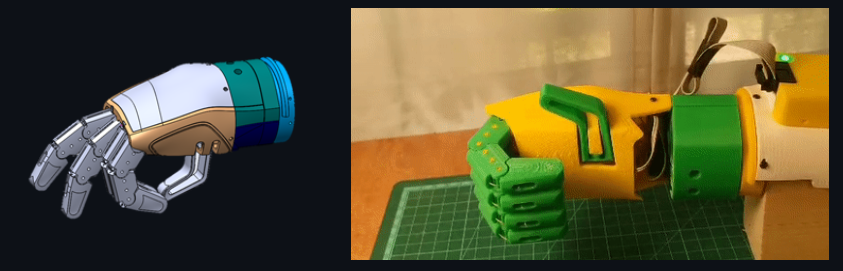
בסך הכול, מצלמת **Oak-D Lite** היא כלי חזק ורב-תכליתי ליישומי ראייה ממוחשבת, עם עיצוב קומפקטי וקל משקל ההופך אותה למתאימה לשימוש במגוון רחב של סביבות ויישומים.

**Oak-d lite Camera:**We used the oak-d Lite camera. The Oak-D Lite camera is a small depth-sensing camera designed for computer vision applications. It is built by the company Luxonis, which specializes in creating computer vision tools and hardware. The Oak-D Lite camera features an Intel Movidius Myriad X VPU (Visual Processing Unit) and a Sony DepthSense IMX556PLR back-illuminated ToF (Time of Flight) depth sensor. These components enable the camera to capture depth information and process it in real-time, making it ideal for applications such as object detection, tracking, and recognition. The Oak-D Lite camera also includes a variety of software tools and libraries to facilitate development, including the Luxonis DepthAI Python API, which allows developers to easily integrate the camera into their projects and applications. Additionally, the camera is designed to be compatible with popular machine learning frameworks, such as TensorFlow and PyTorch, enabling developers to create custom computer vision models and deploy them on the device. Overall, the Oak-D Lite camera is a powerful and versatile tool for computer vision applications, with a compact and lightweight design that makes it suitable for use in a wide range of settings and environments.

A black rectangular object with a black rectangular object

AI-generated content may be incorrect.

# **Haifa 3D Robotic Hand:**

The Oded Hand can perform a variety of user-defined hand configurations using an autonomous adaptive grip along with a variety of grip force levels. The mechanical design consists of four fingers with four different tendon-driven mechanisms actuated by four different micro-DC motors, a passive thumb with two discrete positions using two magnets and wrist rotation driven by planetary gears and a DC motor. Each finger is an independent module, hence from a developer point-of-view, you can check new finger designs and change the number of fingers in your hand model. For more details, see the <https://github.com/Haifa3D/hand-mechanical-design>.  


# **General Architecture:**

The project consists mainly of 3 parts:

* 1. • NVIDIA Jetson Nano developer kit
     + 1. Since the Jetson Nano does not include a Wireless Bluetooth, we added a USB Bluetooth dongle.
  2. • A camera we used the oak-d lite camera.
  3. • A mechanical Hand (attached to it is the camera).

Let’s delve a bit more into the general architecture. Further details about the technologies and implementation will be discussed in later parts of the booklet.

**NVIDIA Jetson Nano Developer Kit**

Jetson Nano is a powerful small computer used for building embedded applications. In our project, it was used as “the brain” of the robotic hand. On this small computer, an object segmentation model and a controller are running. The object segmentation model is used to detect objects in the frame that the camera captured and measured its distance from the hand (abiding by certain thresholds). The controller is used to control the robotic hand which is responsible for rotating and grabbing.

A camera is connected to the Jetson Nano from which the running controller gets further information about the surrounding environment.

**Oak-d lite Camera:**

using its depth estimation feature given by the oak-d Lite camera and the library depth-ai, we measured the distance from the center of the frame to the camera in order to determine need for rotation/aligning and grabbing. It’s connected to the jetson nano via a (USB-A -> USB-C) cable.

**Robotic hand:**

We used the Haifa 3D Bluetooth operated hand, following the manual on the movements of the fingers and how to activate each engine which is found on their Github, after the camera determines a distance from the center of the frame as close enough, we activate our segmentation model on the jetson nano, which then sends commands to the hand via Bluetooth, either to rotate to a certain angle or to grab or to remain idle.

**Overall:**

Haifa 3D Hand

Jetson nano

Oak-d Lite

# **Complete Implementation:**

All the code of this project is open source and available for the general public on GitHub Yamen-Safadi/iRobot-Table-Tennis-ball-collector repository.  
  
Measuring Distance:

Object Segmentation:  
 -edge Detection:

-Segmentation model:

Controlling the Hand with the Jetson Nano:

Integration:  
All the tasks must be done in parallel and on time, because of the nature of the problem we are trying to solve. We used the python “threading” library where we created a thread to control each task:

• A thread to analyse the images captured by the camera and measure distances in the frame, and if needed send the frame into the segmentation model.

• A thread to send the controlling commands to the iRobot Create 3.

After the jetson analyses the frame given to it by the camera feed and it knows where the object is and its orientation it sends the appropriate command to the hand to move accordingly until its ready for its grabbing motion if needed. If we get close enough (our threshold is set to 12cm) the jetson sends a signal to the hand to close.

# **Failed Attempts:**

**C:**

**The Built-in Neural Network in the Oak-d Lite:**

The Oak-d Lite Camera comes with a built-in neural network model to identify objects, using the python library depth-ai

Although the network works well on its own, when running it to try and detect objects we need the object to be predefined in the models training, which doesn’t work well with us since we are searching for any object and only care about its shape and not its label, the code isn’t open source so we couldn’t modify it to our liking.

In addition to that the camera comes with depth estimation, which is achieved using stereo vision, which involves capturing two images of a scene from slightly different perspectives and then using computer vision algorithms to calculate the depth information from the disparity between the two images. This allows the camera to create a 3D representation of the scene. At first we got to precise measurement of as close as 18cm, which was considered too far for grabbing objects, this problem was later resolved by changing some of the camera settings.

# **Possible Future Expansions:**

**Enhanced depth detection:**

**Upgrading the project to consider the object 3D dimension:**

**A more versatile arm:**

**A more powerful GPU for enhanced performance**

# **References**

TODOs:  
1- add a BIG section for the “Algorithm” (3-4 pages)  
 - introduction to the algorithm, what we used (fastsam,…)  
 - very detailed explanation of how the algorithm works (suggestion below).  
 - add flow diagram and sequence diagram of the algorithm  
2- fix the high level flow diagram (Camera -> Jetson NANO -> hand). Current one is correct but missing details.  
3- short explanation about the 2 threads communication – queues/block/unblock/…  
  
  
suggestion:   
separate explanation for calc\_angle thread and hand\_control thread.  
  
calc angle:  
1- take a frame and process it (more details)  
2- calc depth (with more details: challenges (12cm), moving average, median filtering, …)  
3- fastsam segmentation (with more details)  
4- pick the optimal segmentation mask from fastsam results (explain how)  
5- calc angle (how? - PCA…)  
6- communicate the angle to the hand\_control thread (explain that we send the relative angle to the current one, explain that we have a margin (10?), if angle didn’t change more than a margin, we send 0)  
7- if depth <= 12cm (?), tell the hand\_control thread that it should close the hand and block current thread.  
8- …  
  
hand control:  
1- receive the angle from the queue, if angle > 0, rotate hand by this angle  
2- if need to close the hand, close the hand, then open it, and tell the main thread to unblock/continue.  
3 …