IMPERIAL

MSc Individual Project

Imperial College London

Department of Bioengineering

TactEx: A Multimodal Robotic Pipeline for Human-Like Touch and Hardness Estimation

Author:
Felix Verstraete

Supervisor:
Dr. Dandan Zhang

Co-Supervisor:
Wei Lan, Wen Fan

September 9, 2025

Acknowledgements

Everywhere I come, I seek out challenges to grow. This passion to learn and discover is what ultimately drives me in life. Therefore, when I began my second master's degree at Imperial College London, I was determined to explore fields in AI where my knowledge was limited but my interest boundless. I must now say, especially coming from a non-robotics background, that this project was indeed challenging and ambitious. Building a multimodal robotic pipeline from scratch in less than four months became my daily motivation, and piecing it together step by step has resulted in something I am proud of. Looking back, I realize just how much I have learned along this incredible journey.

I must express my sincerest gratitude to Dr. Dandan Zhang for welcoming me in her research group and trusting me with the responsibility of working with the robotic arm. I am also deeply thankful for the immense input you gave during this project. Above all, I am grateful for the opportunities you created to connect with and learn from other researchers. Thanks to you, I was able to participate to the MedTechOne conference, the AI in Healthcare symposium or the GenAI talk, all events that enriched my experience. Finally, I greatly appreciate your support for publishing at the ICRA conference. I know this is one of the top AI and Robotics conferences in the world and having your motivation in writing the paper boosted my confidence enormously.

During my journey, I was fortunate of meeting my co-supervisors Fan Wen and Wei Lan. From the onset of this project, I felt welcomed by you both. Wen, thanks for guiding this project. I realize the start was challenging for me, especially in defining the research question, but your input and support have been instrumental in writing this thesis. Lan, I enjoyed your help in identifying the model collapsing problems and providing me with input for writing the paper. I am grateful to have met both of you and wish you both all the luck in obtaining your PhD.

I want to thank my parents and brothers for their support in studying at Imperial College London. It is a cliché, but this year would really have not been possible without you. Your interest in my research and constant encouragement have meant the world to me. Finally, I thank my girlfriend, for cheering me on till the last day of this project.

To all mentioned, and unmentioned who have touched this academic journey, thank you from the bottom of my heart.

Felix Verstraete, September 2025

Acknowledgements

Achieving human-level touch remains a major limitation within robotics. This study introduces TactEx, a multimodal pipeline combining vision, language, tactility and robotic actions for fruit hardness estimation. After a user request, visual servoing (YOLO or GSAM-based) localizes the object. A GelSight sensor mounted on a robotic arm then registers deformation images. Three deep learning architectures (VGG16-LSTM, ResNet50-LSTM and Transformer) are compared for hardness prediction from these images. They were pretrained on a published dataset and fine-tuned on a custom dataset (7 objects, 40 poses each). Finally, an LLM answers the request. The results demonstrate that the ResNet50-LSTM3 and Transformer outperform the current baseline (VGG16-LSTM3). The ResNet50-LSTM3 was integrated within TactEx, as it achieved the best fine-tuning performance (RMSE 4.30, R^2 0.73, ρ 0.88) and successfully distinguished fruit ripeness levels ($p<0.01$). While GSAM requires more computation, it exceeds YOLO in segmentation score and versatility. The final pipeline, employing GSAM and ResNet50-LSTM3, achieves high success rate (90%) in simple scenarios, but struggles in complex scenarios (30%) where fruits are not explicitly stated. This study demonstrates a training strategy and model architecture that generalizes to real-world tasks. The multimodal integration represents a next step toward more human-like perception within robots. While latency and gripper absence remain limitations, we believe that the design and insights of TactEx may find its applications within household robotics, industry and healthcare.

Keywords: Vision-Based Tactile Sensor, GelSight, Hardness Estimation, Multimodal Integration, Robotic Reasoning

Contents

List of Figures

List of Tables

1 Introduction

1.1 Towards human-level touch in robotics

Humans experience the world through a combination of five different senses: taste, touch, vision, sound and smell [1]. While it can be argued that vision takes the central role in the perception of the physical world, research suggests that touch, or tactile feedback, is crucial for effective object manipulation and scenario reasoning [2].

Indeed, as humans, we rely on touch in everything we do. Take the example of grasping a wet or dry soap bar, or manipulating a plastic or glass bottle. Similarly, when selecting the ripest fruit from the table or describing tactile properties from a tissue, it becomes clear that vision-only is limited.

State-of-the-art multimodal robotic systems have made remarkable progress in integrating vision and language [3, 4]. In this context, language often acts as the glue between the robot and the user. Although these systems achieve impressive robotic control, they still lack the integration of tactility, which limits their success in complex reasoning and manipulation tasks [3]. Therefore, if robots want to achieve human-like perception, reasoning and enhanced manipulation, they must integrate accurate tactile sensing with vision and language.

1.2 Tactile integration

1.2.1 Tactile sensors

To enable robots with tactile sensing capabilities, research has focused on the development of tactile sensors. Traditional methods include force-sensitive and piezoelectric sensors [5]. These systems succeed in giving context on the interaction with an object by voltage or force variations, such as contact detection or strength of grasping. However, they struggle with providing clear descriptions on the texture of the object and report difficulties in data transmission [1, 5].

In 2017, researchers provided a breakthrough by introducing a vision-based tactile sensor (VBTS) called GelSight [1, 6]. This sensor uses the interaction of LED light with a soft elastomeric gel. When touching an object, the gel deforms and a camera underneath registers an image. These high-resolution deformation images provide accurate insights into the object shape, texture and even surface tension [6]. This is illustrated in Figure 1 where contact of a marker-based GelSight with a banana, mango and lime are displayed. This shows the difference in surface properties which can be obtained.

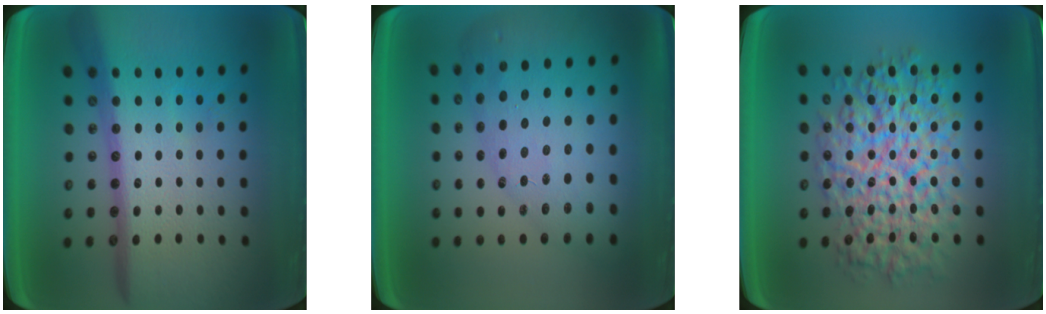


Figure 1: GelSight image collected in this study when in contact with a a) banana, b) mango and c) lime.

Besides GelSight, many other tactile sensors have emerged. For instance, Digit incorporates a more robust elastomeric gel and ViTacTip registers both tactile and visual deformations through its camera [7, 8]. While each sensor has their advantages, GelSight has been the standard choice for many applications. The following section will discuss the various applications of tactile sensors in robotics. Given its popularity and performance in this domain, this study will focus on using GelSight sensors to extract tactile properties.

1.2.2 The use of transfer learning in tactile applications

The advantage of acquiring tactile images by VBTS is that they can easily be used in deep learning schemes [9]. Transfer learning has been the choice of many researchers using tactile sensors. Within this context, the weights of pretrained neural networks on huge datasets like ImageNet are transferred to the research domain. This improves performance, decrease computation time and allow for larger generalizability.

For instance, Gao et al. (2023) have investigated different pretrained convolutional neural networks (CNN) to detect slip [9]. Their conclusion was that the ResNet models were most optimal, followed by the VGG16 and the Inception-V3. From this point, Yang et al. (2024) made improvements by showing that a transformer network, employing self-attention, performed better than a ResNet18 with Long Short-Term Memory (LSTM) for slip detection [10]. Additionally, Calandra et al. (2017) noted the use of pretrained ResNet50 models to predict the success of grasping from GelSight images [11]. They concluded that the combination of tactile data with vision outperformed the results from either modality individually.

Furthermore, foundation models have found their use with tactile data as well. These are general-purpose models trained on large and diverse datasets. For example, Ueda et al. (2024) showed how a robot embodied with tactile sensors could use the zero-shot capability of vision-language models for object recognition [12]. While the above works highlight valuable applications, they overlook the extraction of fundamental tactile properties, making them less insertable in new applications. A more principled way would be to first study these properties and then leverage them for real-world applications, which is the strategy we will adopt here.

1.2.3 Hardness estimation

One of the most crucial tactile properties to predict is hardness. Humans use hardness to recognize an object, manipulate it precisely and reason about it. In robotics, hardness can therefore determine the success of many tasks. For fruits and vegetables for instance, hardness is related to their ripeness level [13]. Automating this fruit ripeness detection by robots has huge potential within agriculture, where it enhances crop harvesting efficiency. Moreover, ripeness estimation also benefits domestic robots, enabling them to take over various kitchen tasks.

Different strategies have been explored to estimate the hardness of an object, yet each of them struggles with shortcomings. Some researchers focus on hardness classification, either from GelSight or traditional sensors [14, 15, 16]. However, these approaches often fail to distinguish differences in closely related objects. For example, as many fruits and vegetables are situated on the higher end of the shore 00 scale, this could cause the system to being unable to differentiate between an unripe and ripe banana.

Other strategies have noted the use of large-language models (LLM) within the field of tactile communication, where based sensor images tactile properties can be requested [17, 14]. However, these touch-vision-language models often require extensive fine-tuning on tactile images, thereby limiting fast implementation within a pipeline.

Another approach is to track changes in the physical properties of the image. Yuan et al. (2024) introduced a numerical model to estimate hardness based on brightness and force changes [13]. Liao et al. (2025) further

applied these force dynamics as a way to track fruit ripeness [18]. While these methods are valid, they use handcrafted features, have shape constraints or show limited generalisability for other applications.

Finally, most hardness estimation models from VBTS rely on deep learning with large datasets. For instance, Nam et al. (2024) uses a custom CNN to predict hardness using a TacTip sensor [19], but they reported systematic degradation in the upper shore 00 scale, which is one of the main hardness scales used in literature. The current baseline model using GelSight is from Yuan et al. (2017), which uses a VGG16-LSTM [20]. While this model achieves high performance ($R^2 > 0.95$) when employed in robots, the integration with other modalities is neglected. Moreover, they did not show statistical confidence when applied to fruit ripeness levels, thereby lacking proof of its practical relevance in a real-world scenario.

Among all the methods noted, deep learning shows the most promising performance, as it can track subtle differences in the image sequence. Therefore, the focus of this project will be on addressing the current shortcomings with deep learning architectures.

1.2.4 Multimodal integration

Recent deep learning methods also works on integrating touch, vision, language and action. Zhao et al. (2023) approaches this with Matcha, a modular pipeline encompassing tactile, language, vision, sound and weight modules [21]. However, their implementation of tactility relies on traditional sensors, thereby lacking the ability to track fine contact changes. In contrast, Guo et al. (2025) have introduced multimodality with a GelSight [22]. However, they focus on classification, making it unsuitable for deployment in fruit ripeness estimation.

Concluding, we hypothesize that the low implementation of previous hardness models results from the combination of following four reasons. First, classification results or degradation effects in the upper shore 00 scale have contributed to unreliability in the output. Second, many current pipelines either require large data collection for CNN training or time-consuming fine-tuning of foundation models. Third, the current deep learning models have not shown statistical confidence in necessary precision hardness tasks like fruit ripeness. Finally, the power of hardness applications lies in its integration with vision and language, which is either neglected in previous research or poorly exploited.

1.3 Contributions

The aim of this study is to resolve the problems stated in previous section. The contributions are as follows:

1. Demonstrating an alternative deep learning architecture which mimics human touch by, for the first time, proving its potential in distinguishing fruit ripeness levels with statistical confidence. The architecture choices are further motivated by including an ablation study.

2. Providing a training strategy that leverages pretraining and a minimal self-collected dataset, avoiding the need for extensive data collection or long fine-tuning as mentioned in literature [23]. This has the potential to transform research field.

3. Integrating state-of-the-art tactile sensing with vision and language within a multimodal pipeline to achieve robust, high-success performance across diverse scenarios.

2 Method

2.1 Model overview

The first two contributions of this project focus on the development of a reliable hardness estimation model and strategy for comparing fruit hardnesses. This will be discussed in section 2.2. The developed model was subsequently integrated with vision and language into a multimodal pipeline, referred to as TactEx ("The Tactile Explainer"), as visible in Figure 2. This completes the third contribution.

Through a chatbot built with Streamlit [24], users can view the input scene and prompt their request, for example to estimate the ripeness of a banana. This request is interpreted and the banana is located using a component integrating vision, language and action. This module is described in section 2.3. After locating the object, the robot hoovers toward it and touches it from the top with a the GelSight sensor. When the hardness is predicted (section 2.2), the request is answered in the chatbot using an LLM (section 2.4).

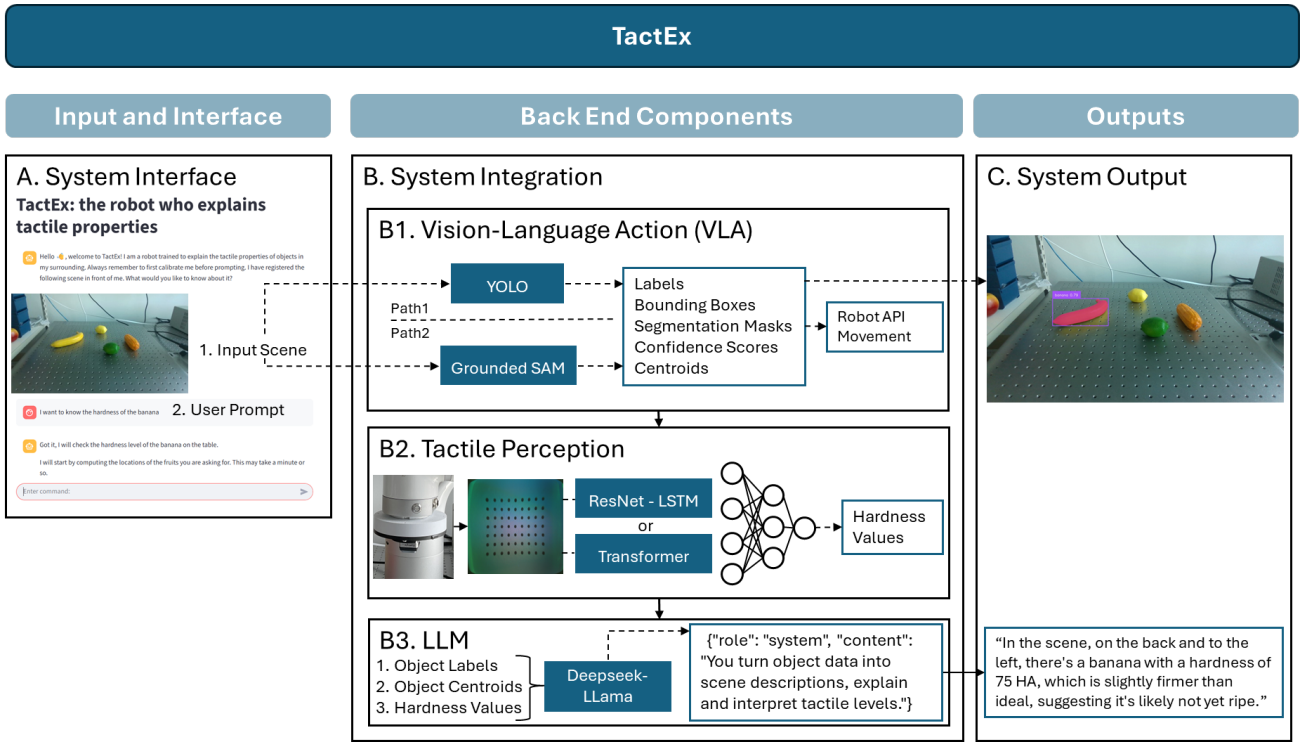


Figure 2: Overview of the TactEx pipeline integrating vision, language, tactility and robotic action into a modular design. Component B2, B1 and B3 are discussed in section 2.2, section 2.3 and section 2.4, respectively.

2.2 Tactile perception

This component (B2 in Figure 2) focuses on predicting hardness from image sequences recorded during the interaction between the GelSight sensor and the object.

2.2.1 Data collection

This study uses two datasets in order to develop the hardness estimation models. One dataset was retrieved from Yuan et al. (2017) [20] and used for pretraining, while the other dataset was collected by this study and used for

fine-tuning the model to the target use case of comparing fruit hardnesses. The collection of image sequences from both datasets mimicked each other, as described underneath.

The pretraining data encompassed about 5000 regular-shaped objects, thereby covering the full shore 00 hardness scale. Contact was defined at a structural similarity score (SSIM) below 0.90 with a reference image [19, 25]. This threshold differs from other studies as Yuan et al. (2017) used a custom marker-based Gelsight sensor rather than a commercial one [20]. Eventually, the 8 frames starting from contact were extracted from the video.

The fine-tuning data in this study was collected with a commercial marker-based GelSight mounted on the end effector of a uFactory 850 robotic arm. Contact with the object followed earlier studies with a SSIM threshold at 0.96 and a mean marker displacement above 2 pixels [19, 26]. Just as in pretraining, 8 images starting from contact were collected at subsequent depths of 0.25mm below each other. This is shown in Figure 3

The collected data for fine-tuning encompassed 5 rubber cubes from different hardnesses (66-80 HA on shore 00 scale), complemented with an elastic band at 88 HA and a glasses pouch at 62 HA. Consequently, this strategy covered most of expected fruit hardnesses range (60-90 HA). Their hardness was measured with a shore A durometer and converted to shore 00 [27]. We collected 40 poses per object, varying the x-y coordinate randomly with $\pm$5mm from the initial position and the yaw between 0 and 45 degrees.

The 8 contact images after pretraining and fine-tuning were transformed into an image sequence where the difference between the selected images and the initial contact was computed. For a 2 image-sequence, the 2nd and 8th image were used, while in the case of 4 images, the 4th and 6th image were added (Figure 3).

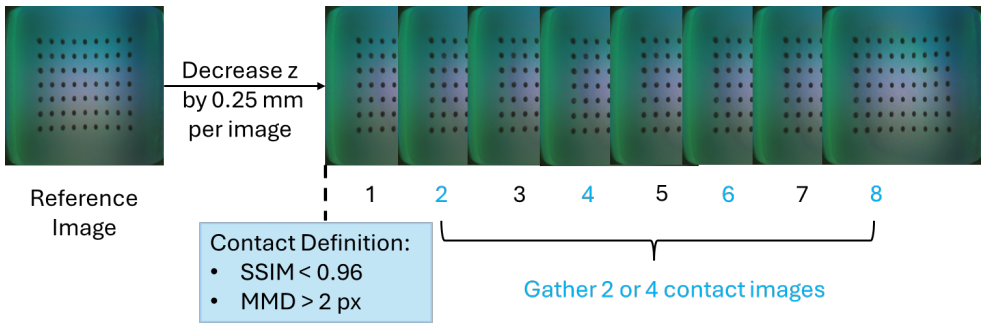


Figure 3: Data collection methodology: images were compared to a reference image. If the contact criteria of structural similarity (SSIM) and mean marker displacement (MMD) were met, 8 images were captured. These were then eventually transformed into a 2 or 4 image sequence.

2.2.2 Model architecture

Three main models are compared in this study (see Figure 4). They all consist of a CNN backbone to get meaningful features from the images and a temporal encoder to interpret the image sequence. First of all, a VGG16-LSTM model was chosen according to Yuan et al. (2017) [20]. Second, as discussed previously, researchers have stated that a ResNet backbone works more optimally to analyse GelSight images [9, 11]. Therefore a ResNet50-LSTM model was included. Lastly, this study also modelled a ConvNext backbone in combination with a transformer model. Instead of recurrent nodes, transformers use self-attention over the whole sequence at once. This should model temporal differences in contact better, as discussed in literature [10].

A further ablation study was conducted to reveal the most optimal design in terms of contact images (2 vs 4) and LSTM layers (1 vs 3). The base models employed 2 images and 3 LSTM layers. The effect of the CNN backbone was also examined by including the ResNet34 and ResNet101 models into the study.

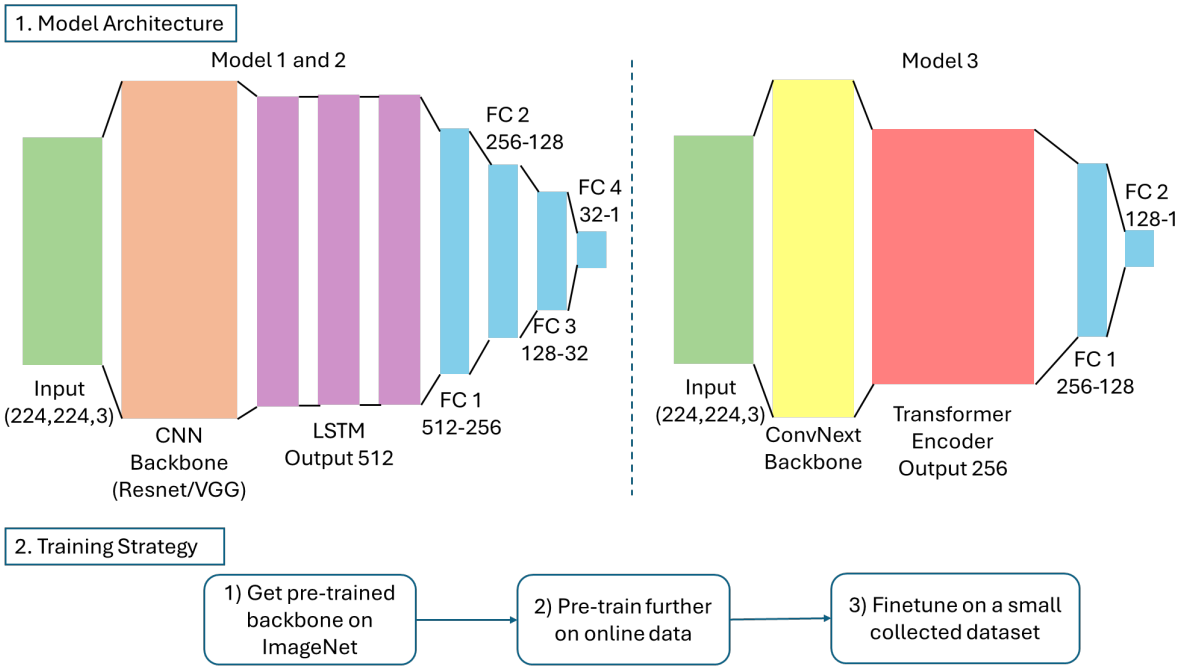


Figure 4: Overview of the used model architectures in this study, together with the training strategy scheme. CNN: Convolutional Neural Network, LSTM: Long Short-Term Memory, FC: Fully Connected

2.2.3 Training strategy

All model training strategies followed the same pattern. First, the pretrained CNN backbone on ImageNet was employed (see Figure 4). Second, the complete models were further pretrained on the online dataset for 80 epochs. Finally, the models were fine-tuned for another 15 epochs on the collected dataset.

All images were augmented by horizontal flips and colour jitter. The latter employed a brightness, contrast and saturation variation of 10% and hue by 1%. Eventually, they were resized into the appropriate dimensions for each network.

The effect of direct training (80 epochs) versus following the pretraining strategy is included in the ablation study. Furthermore, the ablation study also included results of pretraining only on the upper half of the shore 00 scale (>39). It was investigated whether this could lead to better results on downstream tasks such as fruits hardness ranking.

Notably, during multiple occasions, one could notice model collapsing. Three extra measures were taken to counteract this. First, dropout layers (factor 0.1 to 0.2) were added to the network. Additionally, the AdamW optimizer used a higher learning rate in later layers (1e-3) in comparison with early layers (5e-5). Finally, a custom loss function (equation 1) was constructed based on the mean squared error (MSE) and a penalty for low variability (Var) within the outputs. The full code is available at `https://github.com/fv124/TactEx_Submission.git`. A full overview of the hyperparameters employed in the architecture and during training can be found in Appendix A.

$$\mathcal{L} = \text{MSE}(predictions, labels) + 4 \cdot \min\left(\frac{1}{\text{Var}(predictions) + 10^{-6}}, 1000\right) \tag{1}$$

2.2.4 Component evaluation

The test set for pretraining comprised 20% of the online data (N=962). For fine-tuning, the model run 7 times with a leave-one-out procedure. During this setup, the model was fine-tuned on 6 objects and tested on the remaining one. These predictions were subsequently collected.

In order to select the most optimal model after pretraining and fine-tuning, the root mean squared error (RMSE), coefficient of determination (R^2) and spearman correlation (ρ) were noted. The latter one reflects the model's ability to keep ranks between the objects.

Eventually, the full-trained model was validated on a real-world scenario with fruits. The dataset included 3 fruit pairs (mango, lime and tomato) and 2 fruit trios (banana and avocado) at different ripeness stages. For each individual fruit, 20 samples were collected.

As it is difficult to determine the ground-truth shore 00 value for fruits [20], a test was set up to determine if predictions followed the ranks. Since the Shapiro-Wilk test revealed non-normality in some data groups, a non-parametric Wilcoxon rank-sum test was conducted to analyse whether, within one fruit sort, the median on the harder fruit was significantly higher than the softer fruit. The null hypothesis states that there is no difference in median hardness between the harder and softer fruit. As medians are compared, the interquartile range (IQR) will be given. In case of multiple comparisons (bananas and avocados), a Bonferroni-Holms correction was applied.

2.3 Visual servoing with language

This component (B1 in Figure 2) identifies the objects requested by the user and guides the robot to their centroid.

2.3.1 Calibration

In this project, we use an eye-to-hand setup, meaning that the camera (eye) is static while the end effector (hand) is moving. This study used an Intel Realsense Depth camera for locating the fruits. The calibration results in a transformation matrix between coordinates in the camera and robot frame. A detailed workflow behind the method can be found in Appendix B, together with a link to the notebook.

2.3.2 Natural language processing

Basic natural language processing (NLP) was used in order to analyse the user request within Streamlit. Using regular expressions, three key questions were answered:

1. Which property is the user requesting? The only property implemented in this study is hardness (equal to softness or ripeness), but we use it for future generalisability purposes.

2. Which intent does the user have: to detect a single fruit, compare multiple fruits of the same/different types or analyse all fruits in the scene? The intention detection helped avoiding mistakes.

3. Which fruits are requested?

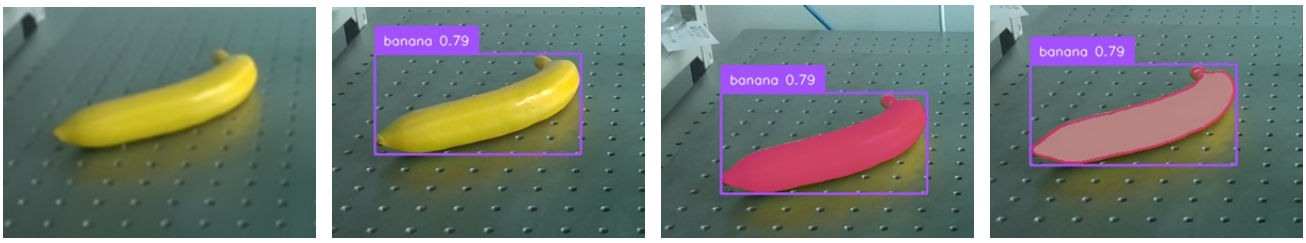
The initial response of the system is hard-coded, with the robot repeating the fruits it will analyse and announcing the start of computation. The last response of the system summarizing the hardness levels and locations will be discussed in section 2.4.

2.3.3 Models: YOLO vs GSAM

This study compares the potential of two methods to segment fruits. The first method uses a YOLO (You Only Look Once) model. This is a widely used CNN-based algorithm for object detection and annotation [28, 29]. More specifically, we used a yolov8n-segmentation model to obtain the object masks.

In order to train the YOLO model, we recorded 60 images among a collection of 12 fruits and vegetables. The ground-truth images were annotated manually using Roboflow. Eventually, the model was trained for 100 epochs. At inference time, we selected a confidence level of 0.40 for fruit segmentation. As YOLO automatically labels all fruits, we use the NLP logic (section 2.3.2) to filter out the requested fruits.

Secondly, a more recent model, called Grounded-Segment-Anything-Model (GSAM), is used [30]. This model first applies Grounding DINO to fit bounding boxes around the requested objects (see Figure 5). Afterwards, SAM is used to generate pixel-accurate masks given the bounding boxes. GSAM is more versatile than YOLO as it can segment anything with any prompt. As there is no additional training in GSAM, we implemented a high confidence threshold (0.60) and employ the NLP logic to remove unwanted objects.



(a) *(b)* *(c)* *(d)*

Figure 5: Example of the Grounded SAM procedure from detecting the object till computing the inner mask. (a) original scene, (b) results Grounding DINO, (c) results SAM and (d) inner mask for computing the centroid.

2.3.4 Centroid computation

In order to prevent the influence of inaccurate image depths by the camera on the centroid computation of the mask, three measures were taken. First, depths were calculated by taking the median across 10 images. Second, as false edges of the mask often caused wrong centroid positions, an inner mask was computed by excluding the edge region (see Figure 5d). This edge was identified by a Canny Edge detector. Finally, the centroid was computed by taking the median of the mask coordinates instead of the mean. This was found to be more robust.

2.3.5 Component evaluation

Both models were tested on 40 single fruits (4 of each of 10 fruits) among a scene of up to 5 fruits. They are evaluated using three metrics: confidence score, segmentation score (intersection over union or IoU) and the distance between the ideal midline of the fruit and the computed centroid. The ground-truth masks were manually annotated using Roboflow. These three metrics were compared using an independent t-test with unequal variances, as Levene's test revealed p-values below 0.01. The null hypothesis of these t-tests is that there is no difference in the means between YOLO and GSAM. Although normality (Shapiro-Wilk) was not met for all groups, we assume the central limit theorem holds true given the large sample size (40).

Additionally, the errors were compared using a one-sample t-test to a threshold of 5mm to test if tactile perception would be valid. This threshold was indeed the deviation from the centre we allowed during data collection

(see section 2.2.1). Finally, also the success rate (SR) was noted (equation 2). In this case, success is defined as the model hoovering toward the correct fruit.

$$SR = \frac{\text{Correct Hoovers}}{\text{Number of Trials}} \tag{2}$$

2.4 LLM component

The goal of the LLM in this study (B3 in Figure 2) is to communicate the hardnesses to the user based on the object labels and centroids computed in section 2.3 and the predictions from section 2.2.

2.4.1 Fine-tuning the LLM

A DeepSeek-R1-Distill-Llama-70B model was used for generating the answer [31]. Two aspects of the prompt are highlighted here, with a full description provided in Appendix C. First, ideal ripeness ranges for bananas, limes, and lemons are included in the prompt and interpretation of ripeness in these cases is requested. These ranges were defined by comparing ripe and unripe fruits with reference hardness objects. Second, the locations are described on a rule based manner (left/center/right and front/center/back). Relative description of positions between nearby fruits is encouraged.

2.4.2 Component evaluation

The LLM is evaluated on 100 prompts from randomly generated scenes (1-6 fruits). An LLM-as-a-judge is used. Within this setup, a dedicated LLM model (Llama-4-Maverick-17b-128e-Instruct [32]) is instructed to score the prompt from 1 to 5 based on three metrics [33]:

1. Accuracy: are the objects, hardnesses and (relative) positions correctly described?
2. Completeness: are all objects from the request mentioned and is info given if an object was not found? Is the ripeness interpreted for the correct cases?
3. Clarity and Coherence: is the description understandable, concise and fluent?

Full transparency behind the prompts and evaluation strategy is given in Appendix B. Connection to the models is enabled via Groq [34].

2.5 Pipeline evaluation

Following research, the complete pipeline will be evaluated based on four scenarios varying in complexity [35, 1, 11]. The complexity level is determined by three factors: 1) number of objects prompted [35, 11, 14], 2) number of distinct objects requested [36] and 3) language reasoning. The latter one is determined by whether the fruits are explicitly mentioned in the request. Overall complexity is then categorized as low, medium and high, as shown in Table 1.

Table 1: Scenario complexity breakdown. [object] = chosen items for the task, [property]= hardness, ripeness or softness.

	Prompt	**Number of Objects**	**Number of Distinct Objects**	**Fruits explicitly stated?**	**Overall Complexity**
1	Identify the [property] of [object].	1	1	Yes	Low
2	Identify the most [property] [object] in the scene.	2	1	Yes	Medium
3	Summarize the [property] of the [object], [object] and [object].	3	3	Yes	Medium-High
4	Summarize the [property] of all fruits in the scene	5	3/4/5	No	High

In these scenarios, tactility values will not be tested again. Contrarily, an object-level (OL-SR) and scenario-level success rate (SL-SR) will be used. The OL-SR is defined as the average percentage of fruits the model was able to accurately identify, measure and communicate through the LLM. The SL-SR is more restrictive: it is defined as the percentage of times the total scenario was correctly executed and communicated. For instance, if 1 of 5 objects is mislocated, the OL-SR would be 4/5 while SL-SR 0. Finally, the latencies are reported as the average among the succeeded trials per scenario. A breakdown in latencies will reveal which steps take the longest.

3 Result

3.1 Tactile perception

3.1.1 Model selection and ablation study

The results of the baseline models and ablation study after pretraining and fine-tuning are reported in Table 2. The model employing a VGG16 backbone and 3 LSTM layers (VGG-LSTM3) showed model collapsing, even after introducing the adpative measures. Therefore, it was not further examined in the subsequent ablation studies. The regression results for the two best models after fine-tuning, namely the ResNet50-LSTM3 pretrained on the full scale and the Transformer pretrained on the half scale, are illustrated in Figure 6, both before and after fine-tuning. These are also the two selected models chosen in the deployment of the pipeline and fruit rank test.

Table 2: Root mean squared error (RMSE), coefficient of determination (R^2) and spearman correlation (ρ) for different models under various conditions. Results are shown both after pretraining (80 epochs on online data) and fine-tuning (15 epochs on collected data). ResNet50-LSTM3: baseline model employing a ResNet50 as CNN backbone and 3 LSTM layers. The baseline models use 2 contact images.

Model	**Pretraining**			**Fine-tuned**		
	RMSE	R^2	ρ	RMSE	R^2	ρ
Baseline Results						
ResNet50-LSTM3	7.18	0.93	0.95	4.30	0.73	0.88
Transformer	6.87	0.93	0.94	6.23	0.44	0.77
VGG-LSTM3	27.98	-0.01	-0.03	20.99	-5.32	0.02
Effect When Training on Half Scale						
ResNet50-LSTM3	9.11	0.63	0.77	5.01	0.64	0.86
Transformer	9.03	0.65	0.81	4.33	0.73	0.89
Effect of Different ResNet Backbone						
ResNet34-LSTM3	6.83	0.94	0.96	6.50	0.41	0.89
ResNet101-LSTM3	7.46	0.92	0.95	7.79	0.16	0.86
Effect of More Contact Images (4)						
ResNet50-LSTM3	7.13	0.93	0.96	8.80	-0.11	0.73
Transformer	6.86	0.93	0.95	6.27	0.44	0.76
Effect of LSTM Depth (1 layer)						
ResNet50-LSTM1	7.43	0.93	0.95	10.63	-0.65	0.78
Effect of Direct Training						
ResNet50-LSTM3	-	-	-	9.40	10.11	0.21

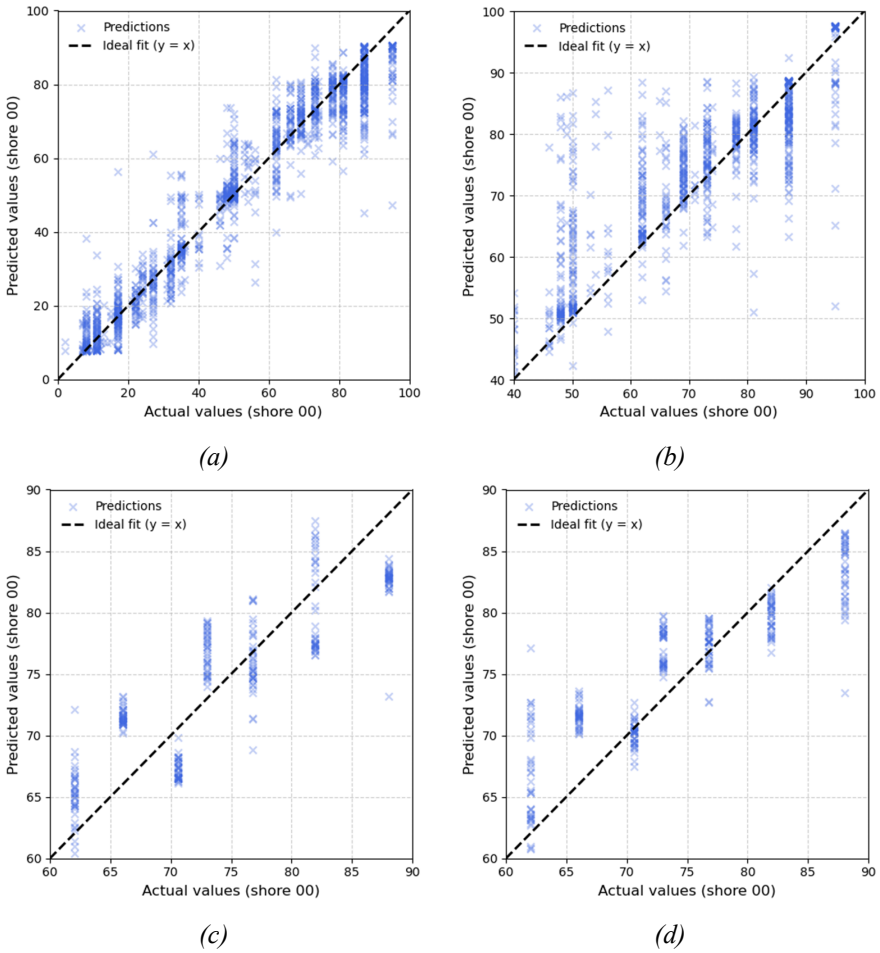


Figure 6: Results of selected hardness prediction models: (a) Pretraining results on full-range trained ResNet50-LSTM3 and (b) half-range trained Transformer model, (c) Fine-tuned results on full-range trained ResNet50-LSTM3 and (d) half-range trained Transformer.

3.1.2 Fruit rank tests

The Wilcoxon rank-sum tests on the fruit pairs and trios are reported in table 3 for the two selected models. A test with p<0.01 affirms the hypothesis that the harder fruit is significantly harder than the softer fruits.

3.2 Visual servoing

In case the objects were identified, the independent t-test revealed that the confidence score of the YOLO model (0.921, 95% CI:[0.886, 0.956]) was significantly higher (t=12.84, p<0.01, n_1=39, n_2=36) than the score for the GSAM model (0.645, [0.606, 0.685]). This is reflected in a SR of 0.9 for YOLO and 0.85 for GSAM. In the other cases, either the object was not found or another object had a higher confidence score than the target fruit.

However, in the correct cases, the segmentation score (IoU) of GSAM (0.942, [0.927, 0.956])] was significantly higher (t=9.01, p<0.01, n_1=39, n_2=36) than the score from YOLO (0.786, [0.752, 0.820]). Although the errors for YOLO (7.194mm, [5.534, 8.855]) were statistically not different (t=1.39, p=0.17, n_1=39, n_2=36) than GSAM (5.645mm, [4.314, 6.981]), only GSAM succeeded in having an error not statistically different than 5mm (t=1.35, p=0.19, $n = 34$). This is not the case for YOLO, which noted a statistically higher error than 5mm (t=2.80, p<0.01, n=36).

Table 3: Median and interquartile range of predictions from full-range trained ResNet50-LSTM3 and half-range trained Transformer on different fruit pairs and trios. Hard, Medium, and Soft correspond to the empirical ripeness stages. Wilcoxon rank-sum test indicates whether the harder fruit is significantly harder than the softer fruits.

Condition	**Transformer**				**ResNet50-LSTM3**			
	Median	25th	75th	Wilcoxon	Median	25th	75th	Wilcoxon
				Mango				
Hard (1)	79.88	78.86	81.03	$u = 362, p < 0.01$	79.47	72.89	84.59	$u = 343, p < 0.01$
Soft (0)	72.99	71.78	74.30		67.75	65.60	72.78	
				Lime				
Hard (1)	89.09	75.84	96.36	$u = 289, p < 0.01$	64.13	63.78	64.64	$u = 285, p = 0.011$
Soft (0)	79.21	78.40	81.24		63.84	63.73	63.93	
				Tomato				
Hard (1)	77.50	71.74	81.47	$u = 0.69, p < 0.01$	71.02	65.69	79.91	$u = 308, p < 0.01$
Soft (0)	68.49	65.28	74.40		64.14	63.13	65.98	
				Banana				
Hard (2)	75.22	71.25	79.44	2 vs 1: $u = 256, p = 0.067$	72.63	67.50	82.53	2 vs 1: $u = 288, p < 0.01$
Medium (1)	71.86	69.26	77.59	1 vs 0: $u = 324, p < 0.01$	66.87	66.31	67.62	1 vs 0: $u = 362, p < 0.01$
Soft (0)	65.95	64.55	67.61	2 vs 0: $u = 378, p < 0.01$	63.05	62.85	63.89	2 vs 0: $u = 368, p < 0.01$
				Avocado				
Hard (2)	67.91	66.13	70.24	2 vs 1: $u = 182, p = 0.692$	65.25	64.02	65.92	2 vs 1: $u = 299, p < 0.01$
Medium (1)	68.06	65.49	71.60	1 vs 0: $u = 328, p < 0.01$	63.54	63.39	64.12	1 vs 0: $u = 359, p < 0.01$
Soft (0)	64.72	64.07	65.82	2 vs 0: $u = 336, p < 0.01$	61.73	60.97	62.15	2 vs 0: $u = 373, p < 0.01$

3.3 LLM evaluation

The LLM-as-a-judge scores on 5 reveal a solid performance of the LLM answer: the accuracy was 4.19 ± 0.59, completeness 4.94 ± 0.24 and conciseness and clarity 4.92 ± 0.44.

3.4 Pipeline test

The pipeline was tested on different scenarios. The success rate and latencies are reported in Table 4. A significant drop in SL-SR is noted in scenario 4. Performance in all scenarios was heavily impacted by the difficulty of both models in recognizing a kiwi. When discarding the kiwi from the ten optional fruits and vegetables, the SL-SR increases across the four different scenarios. Finally, an example pipeline procedure is visualized in Figure 7.

Table 4: Comparison of YOLO-based and GSAM-based pipelines across different scenarios (Sc1–Sc4). Results show Object-Level and Scenario-Level Success Rate (SL-SR, OL-SR) and a breakdown of the latencies (s). The tactile prediction model implemented in this pipeline is the full-range trained ResNet50-LSTM3.

Metric	**YOLO-based pipeline**				**GSAM-based pipeline**			
	Sc1	Sc2	Sc3	Sc4	Sc1	Sc2	Sc3	Sc4
				Success Rates				
SL-SR	0.8	0.7	0.6	0.3	0.9	0.9	0.7	0.3
OL-SR	0.8	0.8	0.83	0.78	0.9	0.9	0.87	0.72
SL-SR w/o kiwi	0.89	0.78	0.85	0.50	1	1	0.88	0.55
				Latencies (s)				
Centroid Computation	1.99±0.81	2.27±0.93	2.58±0.67	4.77±0.32	47.71±9.52	48.08±2.29	46.48±5.48	60.70±12.69
Tactile Collection	43.07±10.85	81.09±11.67	134.85±14.57	160.00±20.75	35.64±8.71	73.99±14.15	106.38±20.73	203.36±27.41
Movement	8±0	16±0	24±0	40±0	8±0	16±0	24±0	40±0
Tactile Prediction	9.07±2.63	9.48±1.46	12.30±1.30	13.03±1.58	8.45±3.17	14.24±9.02	13.55±1.60	17.51±3.02
Rest	7.75±3.05	6.18±2.59	6.10±2.93	9.01±2.05	7.18±1.44	6.22±2.39	5.63±2.32	3.99±4.70
Total Latency	69.89±11.68	115.02±12.65	179.83±13.94	226.80±19.83	106.97±8.90	158.54±18.44	196.04±25.43	325.55±42.13

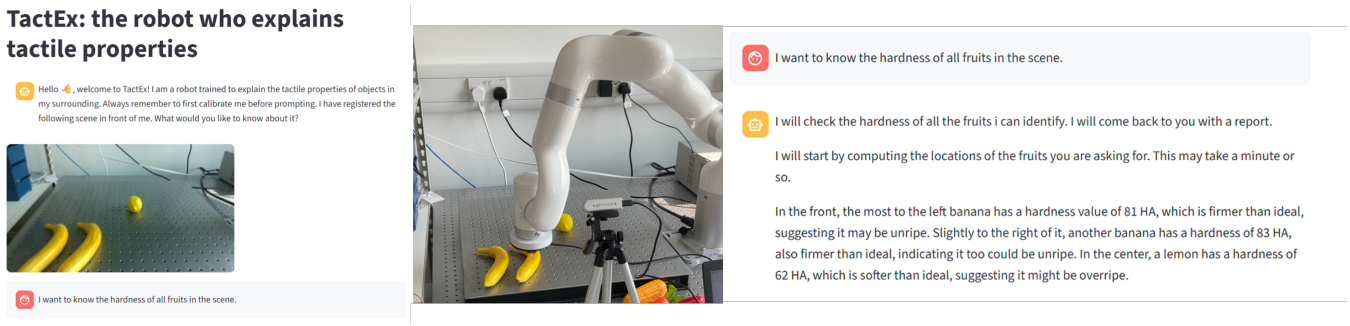


Figure 7: Complete procedure going from Streamlit request (left) to hoovering and tactile sensing (middle) till final LLM communication to the user (right).

4 Discussion

This thesis explored how tactile sensing combined with vision and language enables more human-like perception and action. More specifically, this thesis focused on the estimation of hardness, a challenging but crucial property for a range of applications from manufacturing to medicine and household robots. This can ultimately lead to more effective manipulation of objects, such as distinguishing plastic and glass bottles, or support complex reasoning to overcome limitations in object recognition and tactile communication.

The contributions of this project were threefold: (1) demonstrating and validating an alternative deep learning architecture for precise hardness estimation, (2) providing a training strategy which avoids the need for large data collection and (3) demonstrating the ability of integrating state-of-the-art tactility with vision, language and robot actions into a multimodal pipeline. Below, we discuss the most important findings for future researchers aiming at implementing hardness estimation within their robot system.

4.1 Contribution 1 and 2: model architecture, practical relevance and training strategy

4.1.1 Baseline results reveal more robust architectures

This study opens the door for novel architectures that outperform the model proposed by Yuan et al. (2017) [20]. Specifically, the baseline results in Table 2 reveal that the VGG16-LSTM is prone to model collapsing, despite the extra measures taken in section 2.2.3. This likely explains the low adaptation within existing literature.

In contrast, the ResNet50-LSTM3 and Transformer models avoided model collapsing, thereby indicating that the applied measures were effective. The ResNet50-LSTM3 models achieved the best performance (lowest RMSE, highest R^2 and spearman correlation (ρ)) of the three models, both after pretraining and fine-tuning. We hypothesize two reasons for the differences between the models. First, the number of parameters in the CNN backbone likely influences the risk of model collapsing: about 150M parameters in VGG16 versus 25M in ResNet50 and 88 million in ConvNext. Additionally, the finding that the ResNet50 models are more robust due to their residual connections, also aligns with results reported in literature [9].

4.1.2 An effective training strategy for limited data

Besides revealing the most optimal architecture, the results in Table 2 suggest a new effective training strategy for future researchers. Four findings support this statement. Firstly, the directly trained ResNet50-LSTM3 performed poorly (RMSE 9.40) compared to the pretraining strategy (RMSE 4.30). Second, while the pretrained models achieve slightly higher RMSE than related works (6.87 vs 5.18), the R^2 values are consistent with state-of-the-art models [20, 19].

Third, the fine-tuned models demonstrate that even with a low amount amount of collected data the model seems to generalize well to a new robot setup. Indeed, the spearman correlation remains high, with only minor inconsistencies visible in Figure 6. This suggests that the model is great at ranking hardness values, with minimal deviations from the true values (RMSE 4.30). Fourth and finally, none of the visualized models in Figure 6 reveal systematic degradation for harder objects, suggesting that the Gelsight sensor is better suited than the TacTip used by Nam et al. (2024) [19].

We remark that the drop in RMSE after fine-tuning could partly be devoted to the smaller range during fine-tuning. The impact of the above findings may nevertheless not be underestimated: we demonstrate as one of the first studies how pretraining and fine-tuning enable reliable hardness estimation within robots.

4.1.3 Ablation results revealed that three LSTM layers, a short image sequence and a ResNet50 backbone are most optimal

The glass pouch object distorted the R^2 and RMSE values after fine-tuning, suggesting the need for larger datasets ($\geq$10 objects). Nevertheless, the conclusions from the ablation study remain valid, as they are mostly based on the spearman correlation. Three design choices emerged as most effective: (1) four contact images do not improve the model after pretraining and fine-tuning suggesting that earlier research was right in that the first and last contact image are most crucial [19], (2) three LSTM layers better capture the subtle variation between contact images and (3) while we believe the other ResNet backbones also work for our use case ($\rho \geq 0.80$), the ResNet50 seems to balance complexity and robustness better (lower RMSE and higher R^2).

With regard to the pretraining range, results are dubious: while the transformer model improved when pretrained on half the scale (ρ of 0.89 vs 0.77), the ResNet50-LSTM3 model did not (ρ of 0.86 vs 0.88). Therefore, our recommendation is as follows: if the application of interest has a predefined target range, it is worthy to investigate whether pretraining on that range may help. In this study, the full-range ResNet50-LSTM3 and the half-range Transformer were eventually selected for detecting the ripest fruit, as they achieved the lowest RMSE scores, and highest R^2 and ρ after fine-tuning.

4.1.4 The real-world demonstrates practical relevance

When adapted to the fruit rank scenario, both models perform remarkably well. Except for the lime, the median predictions between both models are less than 6 HA off (Table 3). The variation in predictions was lower in the Transformer versus the ResNet. We hypothesize this is due to two factors: first, the pretraining range described earlier and second, the attention mechanism of the Transformer, which is less prone to light variations during the day [10].

However, the ResNet50-LSTM3 performed better on the Wilcoxon rank-sum tests with all comparisons being statistically significant ($p<0.01$). This demonstrates that the model can correctly interpret which fruit is softest, thereby mimicking human touch. This statistical significance is something not earlier discovered in literature [13, 18]. We therefore conclude that the ResNet50-LSTM3 was most optimal for this application, closely followed by the Transformer model. We believe both are valid alternatives to the VGG16 for many research applications.

4.2 Contribution 3: an integrated pipeline

4.2.1 Motivating the modular design

The pipeline showcases a novel multimodal framework for describing ripeness values of fruits. It goes beyond existing models within tactile communication by integrating vision, language, action and state-of-the-art tactile sensing within one framework [14, 21]. While simple, the modular design is practical, as it allows for easy integration of new modalities such as temperature and sound [21, 37]. Also, the components can be easily fine-tuned for other use cases.

4.2.2 Kiwi as a problematic case

The mismatch between SL-SR and OL-SR (Table 4) reveals that both models have problems with certain fruits, more specifically a kiwi. This strongly impacts the SL-SR. The effect was further amplified in more complex scenarios where kiwis appeared more frequently. Indeed, discarding the kiwi cases improved the performance

dramatically. For YOLO, this likely reflects the low amount of kiwi images (15) and too little training (100 epochs). We encourage future researchers to include at least 25 images per class and train for 150 epochs [38].

4.2.3 Motivating the success across different complexities

The decrease across higher complexity is consistent with literature [35]. The SL-SR in Table 4 demonstrates that when the fruits are explicitly stated, strong performance is achieved. This shows the model's ability to select the hardest fruit on the table. In the case of a banana, lime and lemon, the LLM interprets these hardnesses as ripe or unripe. It does this with high accuracy ($4.19 \pm 0.59/5$), making it practical in household applications.

In the fourth scenario however, the SL-SR drops significantly, even without kiwis. This is explained by the fact that the prompt did not mention the fruits explicitly. In contrary, a list of 20 fruits and vegetables was prompted in GSAM. However, without defined fruits of interest, we cannot use the NLP logic from section 2.3.2 to filter out mistakes. This ultimately leads to more misdetections. GSAM suffers more from this than YOLO, as highlighted by the OL-SR. This is explained by the lower confidence score reported in section 3.2.

In general however, GSAM outperforms YOLO by 10 percent point. This is explained by the lower error and better segmentation reported in section 3.2, which makes tactile prediction more reliable. Furthermore, GSAM is more versatile and insertable into new application compared to YOLO as it does not require any training.

In conclusion, we believe that the GSAM-based pipeline integrating a ResNet50-LSTM3 model for hardness prediction and Deepseek-Llama model for language response, represents a first step toward human-like touch in robots.

4.3 Limitations and future work

While the contributions are present, several limitations remains. First and foremost, the current pipeline is limited by the high latencies reported in this study. Table 4 revealed that the limiting factors were tactile collection and in the case of GSAM, the centroid computation. Time was indeed not a focus of this study. We believe future work can improve it by (1) decreasing the security height above the object, (2) increasing the speed of the arm, (3) decreasing the pausing fragments after movement or (4) using FastSAM for centroid computation [39]. Also, future work should enable real-time scene updates instead of reloading the scene for each new request.

Additionally, we remark that the performance of the model is heavily impacted by the accuracy of the calibration. Furthermore, the precision of the centroid computation is limited by using a single camera. Indeed, two or multi-angle camera snapshots should improve contact point calculation and help detect occluded fruits. We encourage future researchers to identify methods for estimating the ideal contact point.

Moreover, hardness prediction is likely to fail on more complex shaped objects. Future work should therefore focus on extending the datasets, both in pretraining and fine-tuning.

Last but not least, a missing link in this study is the integration of a gripper. In Appendix D, a linkage has been designed to enable integration of the GelSight within the fingers of a robot. We encourage future studies to investigate this further. If the hardness estimation model could then be integrated within models like OpenVLA, the use of tactile properties could not only be limited to describing tactile properties but also enhancing robot manipulation [4]. We believe this represents the key direction of future robotic research.

5 Conclusion

This thesis tackled four of the main challenges identified in literature. First, we demonstrated that the ResNet50-LSTM3 model is a robust and reliable alternative for many hardness estimation applications. Second, the developed training strategy provides a user-friendly framework for researchers. It involves pretraining on a published dataset and then fine-tuning on a limited dataset specific to the use case, while ensuring that both datasets are closely aligned in terms of their image sequence extraction. Third, the applicability to comparing real-world fruit ripenesses illustrated statistical confidence and thereby practical relevance. By doing so, we are one of the first studies within deep learning to prove the capability of human-like feeling within robots. Finally, by integrating tactility with vision, language and robotic actions, we demonstrate the effectiveness of a multimodal pipeline. The modular design could be easily adopted for many applications within agriculture, industry, medicine or household settings. While challenges remain, this thesis represents a meaningful step toward more human-like perception and manipulation within robotics.

References

[1] F. Verstraete, "Project planning: Enhancing robot manipulation and scenario reasoning through multimodal perception," tech. rep., Imperial College London, 4 2025.

[2] I. Camponogara and R. Volcic, "Integration of haptics and vision in human multisensory grasping," *Cortex*, vol. 135, pp. 173–185, 2 2021.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 7 2023.

[4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," 6 2024.

[5] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing-from humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, pp. 1–20, 2 2010.

[6] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," 12 2017.

[7] M. Lambeta, P. W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. Calandra, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, pp. 3838–3845, 7 2020.

[8] W. Fan, H. Li, W. Si, S. Luo, N. Lepora, and D. Zhang, "Vitactip: Design and verification of a novel biomimetic physical vision-tactile fusion sensor," in *IEEE International Conference on Robotics and Automation (ICRA 2024)*, 1 2024.

[9] J. Gao, Z. Huang, Z. Tang, H. Song, and W. Liang, "Visuo-tactile-based slip detection using a multi-scale temporal convolution network," 2 2023.

[10] J. Yang, M. Chen, W. Chen, Q. Lu, H. Wei, and Y. Zhang, "Vt-vt: a slip detection model for transformer-based visual-tactile fusion," *Advanced Robotics*, vol. 38, pp. 1177–1187, 2024.

[11] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The feeling of success: Does touch sensing help predict grasp outcomes?," tech. rep., 2017.

[12] S. Ueda, A. Hashimoto, M. Hamaya, K. Tanaka, and H. Saito, "Visuo-tactile zero-shot object recognition with vision-language model," in *International Conference on Intelligent Robots and Systems*, IEEE/RSJ, 9 2024.

[13] W. Yuan, M. A. Srinivasan, and E. H. Adelson, "Estimating object hardness with a gelsight touch sensor," tech. rep., 2024.

[14] S. Yu, K. Lin, A. Xiao, J. Duan, and H. Soh, "Octopi: Object property reasoning with large tactile-language models," 5 2024.

[15] Y. Sharma, S. Akhbari, C. Guo, P. Ferreria, and L. Justham, "Real-time hardness prediction using cots tactile sensors in robotic grippers," p. 111, MDPI AG, 5 2025.

[16] J. Chen, A. Kshirsagar, F. Heller, M. G. Andreu, B. Belousov, T. Schneider, L. P. Y. Lin, K. Doerschner, K. Drewing, and J. Peters, "Investigating active sampling for hardness classification with vision-based tactile sensors," 5 2025.

[17] L. Fu, G. Datta, H. Huang, W. C.-H. Panitch, J. Drake, J. Ortiz, M. Mukadam, M. Lambeta, R. Calandra, and K. Goldberg, "A touch, vision, and language dataset for multimodal alignment," in *Proceedings of the 41 st International Conference on Machine Learning*, 2024.

[18] Z. Liao, Y. Du, J. Duan, H. Liang, and M. Y. Wang, "Quantitative hardness assessment with vision-based tactile sensing for fruit classification and grasping," 5 2025.

[19] S. Nam, T. Jack, L. Y. Lee, and N. F. Lepora, "Softness prediction with a soft biomimetic optical tactile sensor," in *2024 IEEE 7th International Conference on Soft Robotics, RoboSoft 2024*, pp. 121–126, Institute of Electrical and Electronics Engineers Inc., 2024.

[20] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson, "Shape-independent hardness estimation using deep learning and a gelsight tactile sensor," 4 2017.

[21] X. Zhao, M. Li, C. Weber, M. B. Hafez, and S. Wermter, "Chat with the environment: Interactive multimodal perception using large language models," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 3590–3596, Institute of Electrical and Electronics Engineers Inc., 2023.

[22] Z. Guo, H. Chen, X. Mai, Q. Qiu, G. Ma, Z. Kappassov, Q. Li, and N. Chen, "Robotic perception with a large tactile-vision-language model for physical property inference," 6 2025.

[23] Y. Sun, N. Cheng, S. Zhang, W. Li, L. Yang, S. Cui, H. Liu, F. Sun, J. Zhang, D. Guo, W. Han, and B. Fang, "Tactile data generation and applications based on visuo-tactile sensors: A review," *Information Fusion*, vol. 121, p. 103162, 2025.

[24] "Streamlit: A faster way to build and share data apps." https://streamlit.io/, 2025. Accessed: 2025-08-20.

[25] X. Zhang, T. Yang, D. Zhang, and N. F. Lepora, "Tacpalm: A soft gripper with a biomimetic optical tactile palm for stable precise grasping," *IEEE Sensors Journal*, 2024.

[26] J.-i. Lee, S. Lee, H.-M. Oh, B. R. Cho, K.-H. Seo, and M. Y. Kim, "3d contact position estimation of image-based areal soft tactile sensor with printed array markers and image sensors," *Sensors*, vol. 20, no. 13, 2020.

[27] Administrator, "Hardness table (conversion)," *MEREFSA - Meet Your Silicone*, May 2025.

[28] Q. Bai, S. Li, J. Yang, Q. Song, Z. Li, and X. Zhang, "Object detection recognition and robot grasping based on machine learning: A survey," *IEEE Access*, vol. 8, pp. 181855–181879, 2020.

[29] P. Durdevic and D. Ortiz-Arroyo, "A deep neural network sensor for visual servoing in 3d spaces," *Sensors (Switzerland)*, vol. 20, 3 2020.

[30] T. Ren, S. Liu, A. Zeng, J. Ling, H. Cao, K. Li, J. Chen, X. Huang, F. Yan, and Y. Chen, "Grounded sam: Assembling open-world models for diverse visual tasks international digital economy academy (idea) & community," tech. rep., International Digital Economy Academy (IDEA) & Community, 1 2024.

[31] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 1 2025.

[32] Meta AI, "meta-llama/llama-4-maverick-17b-128e-instruct." https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E-Instruct, 2024. Accessed: 2025-08-20.

[33] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, and J. Guo, "A survey on llm-as-a-judge," 3 2025.

[34] "Groqdocs overview." https://console.groq.com/docs/overview. Accessed: 2025-08-20.

[35] L. Verbaan, Y. B. Eisma, and R. van Leeuwen, *Perception and Control with Large Language Models in Robotic Manipulation Developing and assessing an integrated Large Language Model System on environmental and task complexity*. PhD thesis, Delft University of Technology, 8 2024.

[36] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," 1 2023.

[37] Q. Mao, Z. Liao, J. Yuan, and R. Zhu, "Multimodal tactile sensing fused with vision for dexterous robotic housekeeping," *Nature Communications*, vol. 15, 12 2024.

[38] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 12 2016.

[39] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," 6 2023.

[40] "Llm-as-a-judge simply explained: The complete guide to run llm evals at scale - confident ai." https://www.confident-ai.com/blog/why-llm-as-a-judge-is-the-best-llm-evaluation-method, 2025. Accessed: 2025-08-23.

A Hyperparameters

This appendix details the choice of the most important hyperparameters employed in this project in order to repeat the experiments. Table 5 details the augmentation and data preprocessing parameters. The train-test split was only used during pretraining as fine-tuning were gathered after a leave-one-out procedure.

Table 5: Data Preprocessing and Augmentation Hyperparameters

Parameter	**Value**
Resize dimensions	224×224
Color jitter (brightness)	0.1
Color jitter (contrast)	0.1
Color jitter (saturation)	0.1
Color jitter (hue)	0.01
Horizontal flip probability	0.5
Train-test split ratio	0.2
Train-test split random state	1

Table 6 details the hyperparameters chosen for the model architecture. We refer the reader to Figure 4 in the report to analyse which layers are present in which models.

Table 6: Model Architecture Hyperparameters. FC = Fully Connected Layer, LSTM = Long-Short Term Memory

Parameter	**Value**
Dropout in Transformer Encoder (if present)	0.2
Dropout in LSTM layers (if present	0.2
Dropout between FC1–FC2	0.2
Dropout between FC2–FC3 (if present)	0.1

Finally, the hyperparameters during actual training are detailed in Table 7. This employs the used optimizer and scheduler. As stated in the report, the full code is available at `https://github.com/fv124/TactEx_Submission.git`.

Table 7: Training Hyperparameters. LSTM = Long-Short Term Memory, FC = Fully Connected, LR = Learning Rate

Parameter	**Value**
Optimizer	AdamW
Learning rate (backbone, LSTM/Transformer encoder)	3e-5
Learning rate (FC layers)	5e-5
Weight decay	1e-4
Batch size	8
Epochs (pretraining)	80
Epochs (fine-tuning)	15
LR scheduler Name	ReduceLROnPlateau
LR scheduler Factor	0.2
LR scheduler Patience	1
LR scheduler Mode	min

B Calibration Procedure

The goal of the calibration is to find an accurate transformation between camera coordinates and the coordinates in the base frame of the robot. In this project, we use an eye-to-hand setup, meaning that the camera (eye) is static while the end effector (hand) is moving. The complete code can be found in the notebook on the GitHub page (`https://github.com/fv124/TactEx_Submission.git`), thereby making it accessible for future researchers in the lab. The process is detailed underneath and illustrated in Figure 8.

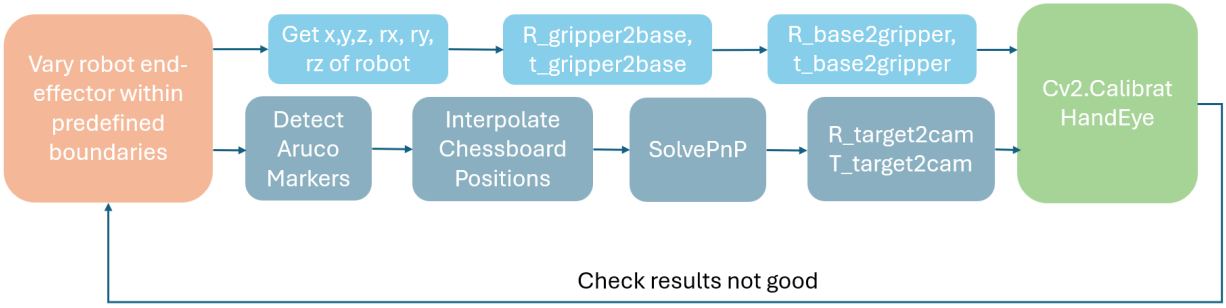


Figure 8: A detailed look into the calibration method. This results in the calculation of the transformation matrix.

The transformation matrix that is searched solves the problem denoted in equation 3. A represents the transformation of the robot hand with respect to the base frame. This matrix can be calculated using a series of rotation matrices $R_{gripper2base}$ and the translation vectors $t_{gripper2base}$. B represents the movement of the target. To estimate it, one needs to find the rotation matrices $R_{target2cam}$ and translation vectors $t_{target2cam}$. To find X in equation 3, we use the function cv2.HandCalibrateEye. This function requires the 2 rotation and 2 translation matrices. However, as it assumes by default a hand-in-eye setup, little adaptations to the calculation were needed: by inverting the matrix $T_{gripper2base}$ composed of the rotation and translation vectors, one could effectively mimic the hand-in-eye calculation and use it for the hand-to-eye setup. Equation 4 illustrates the calculation of the inverse transform used in this project.

$$A * X = X * B \tag{3}$$

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} R^T & -R^T * t \\ 0 & 1 \end{bmatrix}^{-1} \tag{4}$$

In order to find the rotations and translations, we use a ChArUco board mounted centrally to the end effector. We identified that it calculates the chessboard positions more accurately than a normal chessboard. It accomplishes this by first detecting the Aruco markers and then interpolating to the positions (see Figure 9). Furthermore, by having markers, the board does not need to be fully visible in the frame of the camera. This widens the field of poses one could collect, thereby enabling more accurate calculation as well.

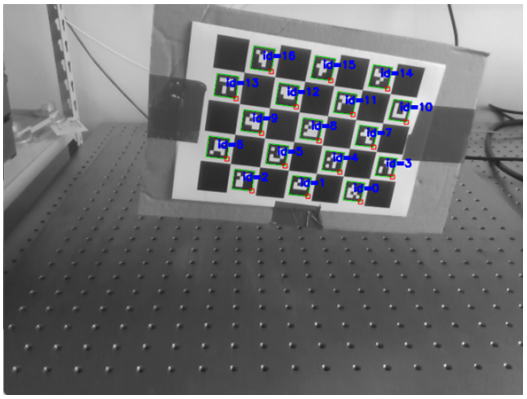


Figure 9: Example of the sytem being able to identify the Aruco markers on the ChArUco board. This enables more precise calculation of the transformation matrix.

Subsequently, to find the rotation and translation matrix from target to the camera, one needs to run the SolvePnP method from cv2. This links the calculated 3D positions of the board with the identified 2D positions in the camera. In total, 40 poses are collected. The position and translations in the robot frame (base2gripper) are easily found by consulting the robot API. Concluding, one can now find the transformation matrix.

As the camera used in this study is a depth camera, one can get the 3D coordinates of the objects in the camera frame. In order to transform these now into the robot coordinates, one can just use equation, using the transformation matrix just computed.

$$\begin{bmatrix} x_{\text{robot base}} \\ y_{\text{robot base}} \\ z_{\text{robot base}} \\ 1 \end{bmatrix} = T_{camera2base} * \begin{bmatrix} x_{\text{camera}} \\ y_{\text{camera}} \\ z_{\text{camera}} \\ 1 \end{bmatrix} \tag{5}$$

C LLM Prompt and Evaluation

C.1 LLM prompt

The goal of the LLM in this study is to summarize the results of the pipeline to the user. The LLM communicates which objects are in the scene, where they are and what their hardness is. To achieve this, the input to the LLM was a list of dictionaries. In each dictionary, the object label, the x and y position and the hardness was defined.

The model used in this study is a DeepSeek-Llama model and was accessed via Groq [34]. Normally, this model is used for reasoning tasks, however, we use it to form a correct answer prompt to the user. A low temperature (0.1) was chosen to encourage predictable and consistent responses.

The prompt given to the system is defined in 10 rules, encompassing guidance on the interpretation of ripeness, location, writing style and structure. We note that ripeness is only interpreted in case of a banana, lemon or lime. The full prompt is given on page 32.

C.2 LLM-as-a-judge

An LLM is not only used in this study to generate a response, but also to evaluate that response. While this may sound counter-intuitive at first glance, there are clear reasons to use an LLM-as-a judge setup in our use case. First, while human evaluation works, it is time-consuming and subjective. Second, in general and certainly also for an LLM, assessing properties is easier than generating. Finally, if done correctly, LLM-as-a-judge has been shown as one of the most reliable and accurate ways to evaluate LLM apps [40].

The evaluation model used in this project is a Llama model accessed by Groq [34]. The LLM has been asked to score a generated prompt based on accuracy, completeness and clarity and coherence. These criteria are outlined in the prompt given on page 33.

Besides the criteria, the model is given more context on the original LLM task: the rules of the scene location interpretation are described and the ripeness interpretation in case of the banana, lime and lemons is prompted as well (points 1 and 2 for C.1). Finally, we also give 4 example prompts, their scores and what could be improved. In this way, the model learns what is considered a good response to the user.

The evaluation is repeated 100 times for generated prompts of one till six randomly positioned fruits. The scores were noted and their mean and standard deviation are given in section 3.3.

Prompt for Scene Generator (C.1)

You are a descriptive scene generator for a robot vision system that identifies fruits and vegetables. You will receive a list of dictionaries corresponding to items, each with positions (x, y): x = front/back (negative = back, positive = front), y = left/right (negative = left, positive = right). You will also receive a hardness measurement (HA: hardness units). Your task is to describe locations, hardness values, and in some cases ripeness, concisely and naturally.

Here are 10 rules to follow:

1. Do not state exact (x, y). Use relative positions for objects close to one another (example: if 2 bananas are close to one another in the back of the scene, say: on the back, the most to the right banana has a hardness value of XX HA. Just slightly to the left of it, the second banana has a hardness level of XX HA). To get object locations out of x,y positions, follow next rules: $x < -170 \rightarrow$ back, $-170 \leq x \leq -20 \rightarrow$ center, $x > -20 \rightarrow$ front. If $y < -520 \rightarrow$ left, $-520 \leq y \leq -400 \rightarrow$ center, $y > -400 \rightarrow$ right. Make sure to check this well.
2. State the hardness value for a fruit. Do also say for banana, lime and lemon if fruit is too soft or hard by looking at the ideal ripeness ranges: for bananas this is between 60 and 75 HA, for limes and limons between 65 and 80 HA. To interpret the ripeness stage of the fruits, use following rules: (1) Above range, Say "firmer than ideal" or "unripe", (2) Below range: Say "softer than ideal" or "overripe", (3) Within ideal range, say ripe. If values are close to one another, don't be over interpretative in ripeness differences.
3. Begin always with an overview of the items that could be found.
4. You must compare the same types of fruits or vegetables in terms of ready to eat or ripeness, example given by saying if there are two bananas: the left one is the most ready to eat or the most frontis the ripest probably. If both are far outside ideal range, do state this that it is likely that no ripe item is present of that fruit. If both are within range, state that both are ripe. Focus on comparisons and not too much on ripeness individually, and state if something is really too soft for its range (overripe) or too hard (underripe).
5. Do not compare different types of fruits or vegetables in terms of ready to eat or ripeness, so skip that.
6. If you only have 1 fruit of a sort, just say hardness (and if soft or hard). Do never state that you do not have the ideal range.
7. Keep in mind location is just as support to talk about object ripeness and relative positions of similar positioned objects. It is just to refer to the user which item you are talking about. If x values are comparable, use y position to make distinction and vice versa. If x and y are center, just say center.
8. If a fruit or vegetable is present with a 0 hardness value (HA), just say to the user at end of answer that that item could not be found in the scene. Do never mention any location or hardness value of if hardness is zero.
9. Use fluent text, human-interpretable and interesting. Not over interpretative. Report in one paragraph. Be very concise. No notes and repetitive words or sayings. Use short sentences.
10. Do not output your reasoning steps or any text inside <think> tags.

Part of Prompt for Evaluator (C.2)

You are an expert evaluator of scene and hardness descriptions outputed by an LLM. You will receive:

1. A structured list of objects with positions (x,y) and hardnesses (HA)
2. A description generated by a model

Your task: Evaluate the description on the following criteria with a score from 0 (very bad) to 5 (very good):

- Accuracy: Are all objects, positions, and non-zero hardness values correctly described? Is their (relative) position correct? (0-5)
- Completeness: Are all objects mentioned? Is ripeness interpreted in the cases of lime, lemon or banana? In case of hardness 0, is the feedback given that the object was not present in the scene? (0-5)
- Clarity and coherence: Is the description understandable, concise, and fluent? Does the response flow logically and consistently from start to finish? Does it make sense? (0-5)

D Integration within a gripper

The GelSight sensor needs a dedicated linkage to mount on the gripper. Figure 10 shows the linkage developed at the end of this project. The linkage is designed in Solidworks. It consists of of 4 holes at the bottom to attach the linkage with the gripper by using screws. The GelSight can be place in the linkage from the top. The opening on the left side of the linkage enables cable connection between the GelSight and the computer.

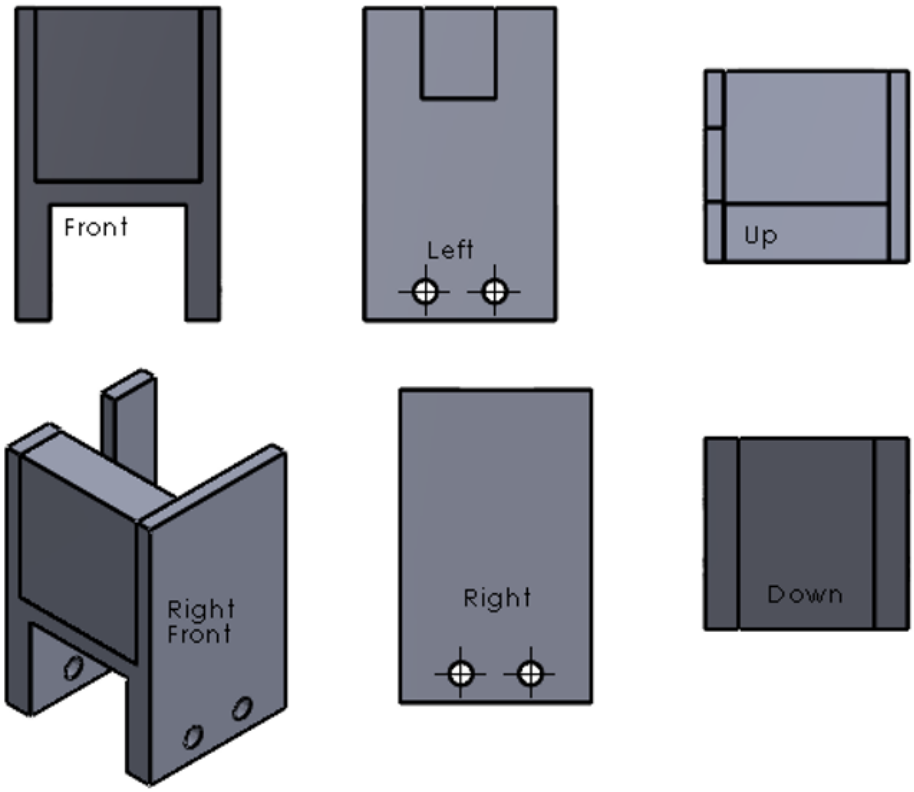


Figure 10: Linkage design in SolidWorks. The front, left, top, down, right and right front view are shown.