ELEC0145

Robotics in Medicine and Industry

Assignment 2

Submission date: 31/03/2024

Group 19

Zihao He	21093057	zihao.he.21@ucl.ac.uk
Jinyuan Zhang	21063715	jinyuan.zhang.21@ucl.ac.uk
Kun Wang	21003947	kun.wang.21@ucl.ac.uk
Yuhan Liu	21013944	zceeiuy@ucl.ac.uk

Table of Contents

Executive Summary

This report serves as a deliverable for ELEC0145 Robotics in Medicine and Industry. The tasks consist of transfer leaning on convolutional neural network (CNN) models which enables robotics to detect and classify three types of package conditions: nominal, multi-pick, and defective packaging. An innovative gripper that incorporates the training algorithm and capable of defect handling was designed and elucidated. In this report, we will provide a comprehensive analysis on the methodology used in transfer learning and detailed design procedure for the gripper.

Task 1: Image Classification

1.1 Introduction

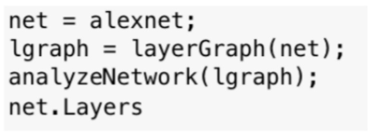
In this task, we focus on an analytical analysis on major pre-trained networks including AlexNet, GoogLeNet, ResNet50 and DenseNet201. Combined with optimization algorithms including adaptive moment estimation(Adam), stochastic gradient descent with moment(sgdm) and root mean square propagation(rmsprop), the accuracy and efficiency for each of the combinations when performing defect detection using image classification are analysed and compared[1].

The dataset tested is a derived version of ARMBench which is a large-scale benchmark dataset for perception and manipulation challenges. Images taken from Amazon warehouse comprises three package picking conditions: nominal, multi-pick, and defective packaging. The original dataset contains 144,129 high-resolution images but was downgraded to 1000 images with dimensions $277 \times 277 \times 3$.

The methodology involved modification of learnable parameters for frameworks in each of the networks such as learning rates, mini-batch sizes, weight and bias factors to align with the specific characteristics of the dataset.

The alterations made to the networks played a pivotal role in enhancing our comprehension of their flexibility and efficacy. Through a comparative study stemming from these modifications, valuable insights were gained regarding the utilization of pre-trained network models in distinct image recognition scenarios. This highlighted their adaptability and potential for optimized performance in the targeted application.

To begin with, we studied the prevailing pre-trained networks to better understand the possible modification to their layers by analyzing them using MATLAB network analysis function shown in the following code snippet executed in a separate environment.



```
net = alexnet;
lgraph = layerGraph(net);
analyzeNetwork(lgraph);
net.Layers
```

Figure 1. AlexNet network analysis

As a result, AlexNet layers are displayed sequentially and can be concluded into a equational representation written as:

$$(CNN \rightarrow RN \rightarrow MP)^2 \rightarrow (CNN^3 \rightarrow MP) \rightarrow (FC \rightarrow DO)^2 \rightarrow Linear \rightarrow Softmax$$

Where
- CNN = convolutional layer (with ReLU activation)
- RN = local response normalization
- MP = maxpooling
- FC = fully connected layer (with ReLU activation)
- Linear = fully connected layer (without activation)
- DO = dropout

One of the major advantages of AlexNet is the implementation of maxpooling layers which help alleviate the vanishing gradient problem and accelerate convergence. Additionally, AlexNet utilized dropout layers to combat overfitting and overlapping pooling to reduce the size of the network while preserving important features.

Similar analysis is conducted on GoogLeNet while we discovered that the introduction of inception modules which contains parallel convolutional layers of varying kernel sizes and pooling, all of which concatenated together to provide an efficient analysis of images. Enhancing GoogLeNet's performance can therefore be achieved by integrating residual connections to mitigate the vanishing gradient issue, allowing for deeper architectures. Additionally, adopting separable convolutions within inception modules can reduce computational costs while maintaining effectiveness. These modifications, alongside optimized hyperparameters and advanced regularization techniques, can significantly improve GoogLeNet's efficiency and accuracy in complex image recognition tasks.

Luckily, there is one network, ResNet (Residual Network), which inherits the advantages of both AlexNet and GoogLeNet, captivating our interest with its innovative approach to solving the vanishing gradient problem through the introduction of residual blocks as shown in the following figure.

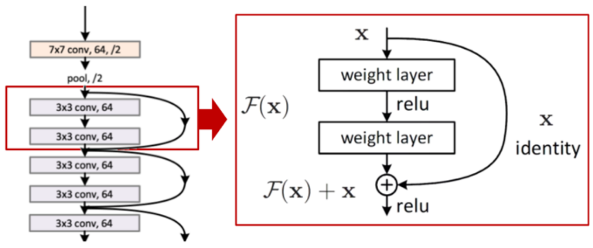


Figure 2. The removal of the layers

The feature of skipping connections allow the network to learn identity functions, enabling the seamless training of much deeper networks by effectively propagating gradients. This architecture not only leverages the depth and convolutional ability seen in AlexNet and the efficiency of GoogLeNet's inception modules but also significantly surpasses them in performance, particularly in tasks requiring deep feature extraction, making it a cornerstone in the advancement of deep learning architectures.

In addition to the above networks, DenseNet201 was also tested due to its high accuracy and depth in training, offering a unique architecture that enables feature reuse through dense connections, while leading to reduced parameters and computational efficiency. The repetitive connections ensure that each layer receives inputs from all preceding layers, strengthening feature propagation and encouraging feature reuse. However, despite these advantages, DenseNet can be more memory-intensive and time-consuming during training due to the need to store the outputs of all preceding layers, and the concatenation of feature maps might lead to a substantial increase in the number of channels, particularly in deeper layers.

After the thorough study of the networks, the transfer learning can then be conducted with modifications to specific layers and learnable parameters which are detailed in the implementation section.

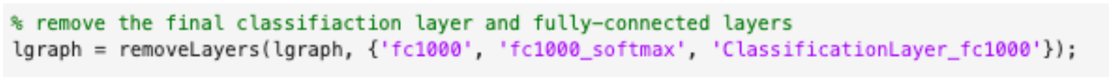
1.2 Implementation

As the transfer learning method is applied, we found there are several CNN models can be used for image classification. We found that AlexNet, GoogLeNet, ResNet50, and DenseNet201 all can be used to train the model. Also, the training algorithms: SGDM, ADAM and RMSprop are used in different CNN models to show the benefits and limitations.

In this task, we are trying to modify the fully connected layers and the classification layer due to the change in the number of classes. In order to apply transfer learning for distinguish the package type. We have done the training process using those CNN models with training coefficient 0.8 and modified the layers.

ResNet

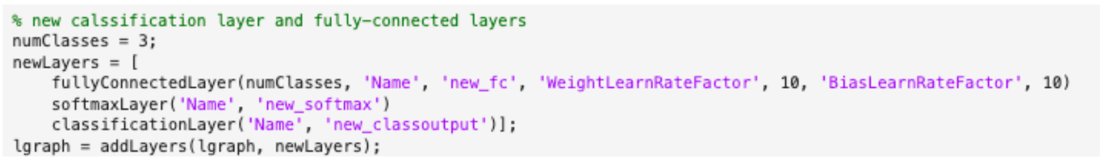
We have modified the final 3 layers suitable for the 3 types of output (nominal, multi-pick, defect package). The image below is how we modify the layers:



```
% remove the final classifiaction layer and fully-connected layers
lgraph = removeLayers(lgraph, {'fc1000', 'fc1000_softmax', 'ClassificationLayer_fc1000'});
```

Figure 3. The removal of the layers

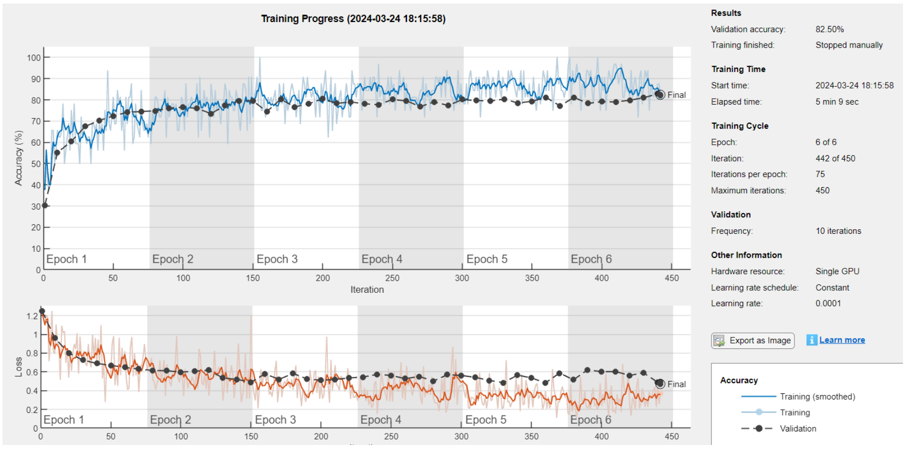
After the original layers has been removed, the new layers related to 3 outputs is created as shown in the code below:



```
% new calssification layer and fully-connected layers
numClasses = 3;
newLayers = [
    fullyConnectedLayer(numClasses, 'Name', 'new_fc', 'WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10)
    softmaxLayer('Name', 'new_softmax')
    classificationLayer('Name', 'new_classoutput')];
lgraph = addLayers(lgraph, newLayers);
```

Figure 4. New designed layers and add to original nets

In the figure above, we have created a fully connected layer, softmax activation function layer and classification layer. Here, MATLAB automatically use the entropy cross loss function. It keeps the gradient value to clearly shows how the gradient is changes during then training process. After modifying the layers and combined to the rest layers, we got the following result by using ADAM as the training algorithm:



Results
Validation accuracy: 82.50%
Training finished: Stopped manually

Training Time
Start time: 2024-03-24 18:15:58
Elapsed time: 5 min 9 sec

Training Cycle
Epoch: 6 of 6
Iteration: 442 of 450
Iterations per epoch: 75
Maximum iterations: 450

Validation
Frequency: 10 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 5. Training result by using ResNet + ADAM

Then, the validation result of this condition is shown in the confusion matrix below:

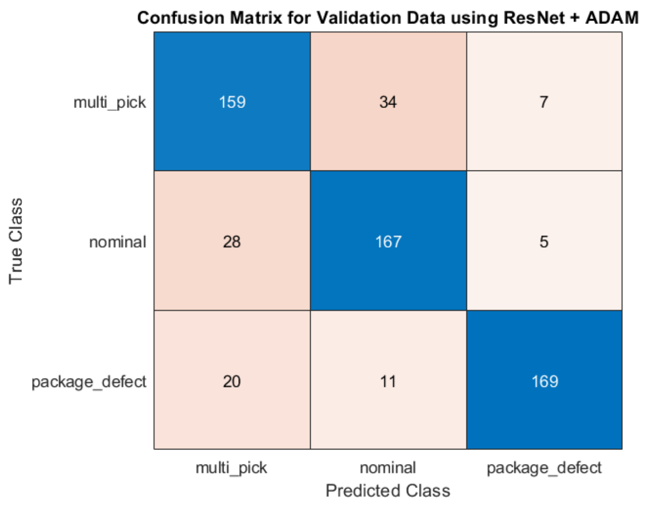


Figure 6. Confusion matrix of ResNet + ADAM

After trying the combination of ResNet and ADAM. We changed to another 2 algorithms which is SGDM and RMSProp. We got the following result as shown in the confusion matrix below the accuracy rate is also calculated:

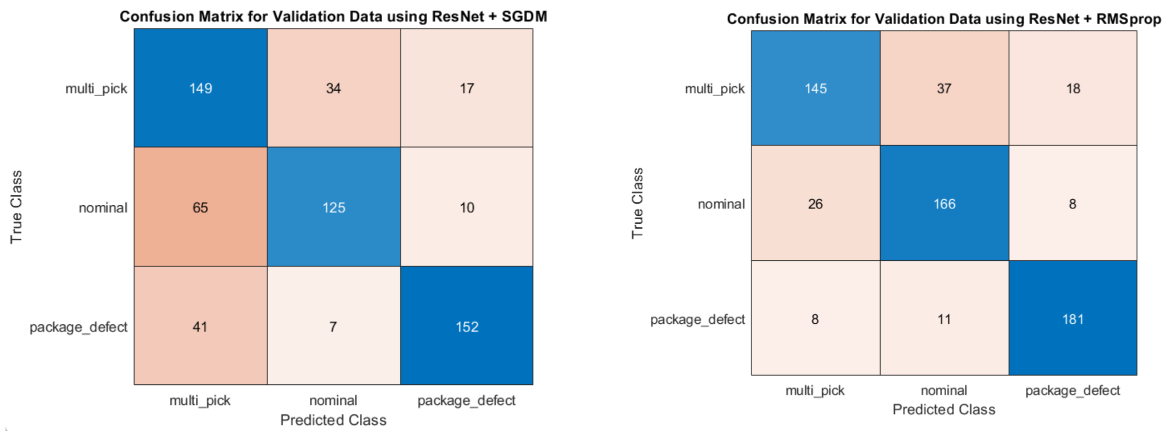


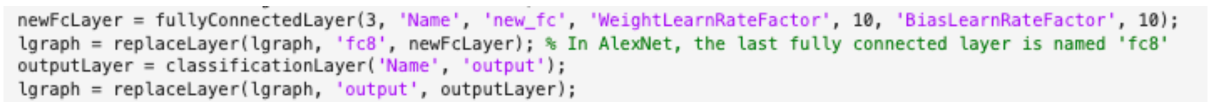
Figure 7. The confusion matrix of using SGDM & RMSprop in ResNet

Here are the successful classification rates:

- ADAM: (159+167+169)/600=82.5%
- SGDM: (149+125+152)/600=71%
- RMSprop: (145+166+181)/600=82%

AlexNet

Apply the same method by modifying the classification layer:



```
newFcLayer = fullyConnectedLayer(3, 'Name', 'new_fc', 'WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10);
lgraph = replaceLayer(lgraph, 'fc8', newFcLayer); % In AlexNet, the last fully connected layer is named 'fc8'
outputLayer = classificationLayer('Name', 'output');
lgraph = replaceLayer(lgraph, 'output', outputLayer);
```

Figure 8. Modify the layers for AlexNet

The rest parameters remain the same to make sure only the CNN model and the training algorithm affects the result. Then we did another 3 comparison experiments by changing the algorithm:

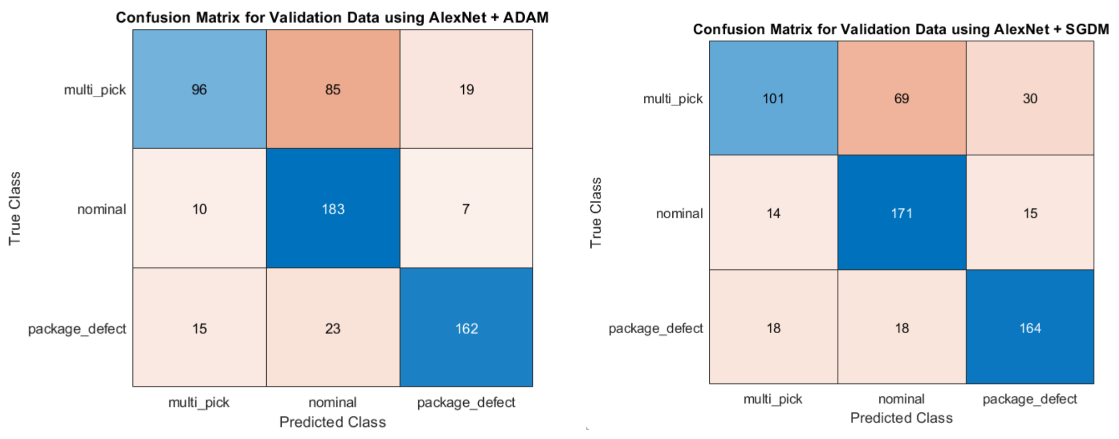


Figure 9. Validation results by using AlexNet combined with ADAM & SGDM

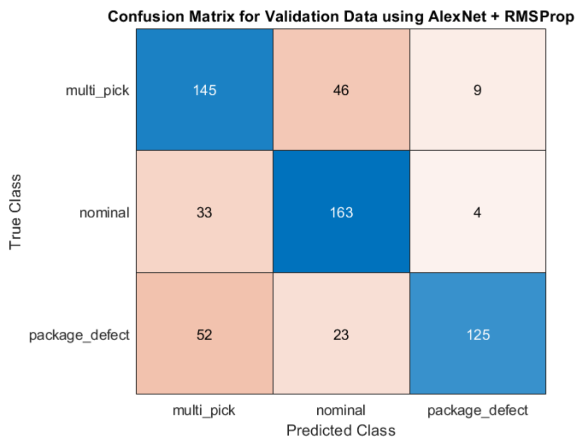


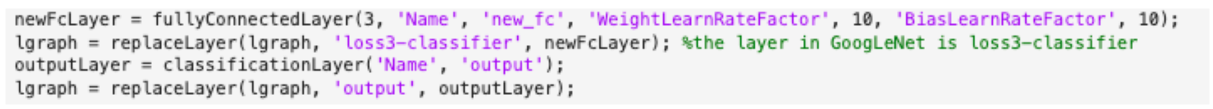
Figure 10. The validation result of AlexNet using RMSprop

Here are the successful classification rates:

- ADAM: (96+183+162)/600=73.5%
- SGDM: (101+171+164)/600=72.67%
- RMSprop: (145+163+125)/600=72.17%

GoogLeNet

Same method is applied in this CNN model by modify the layers:



```
newFcLayer = fullyConnectedLayer(3, 'Name', 'new_fc', 'WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10);
lgraph = replaceLayer(lgraph, 'loss3-classifier', newFcLayer); %the layer in GoogLeNet is loss3-classifier
outputLayer = classificationLayer('Name', 'output');
lgraph = replaceLayer(lgraph, 'output', outputLayer);
```

Figure 11. Modify the layers for GoogLeNet

The following image is the validation results by applying ADAM, SGDM, and RMSprop training algorithms:

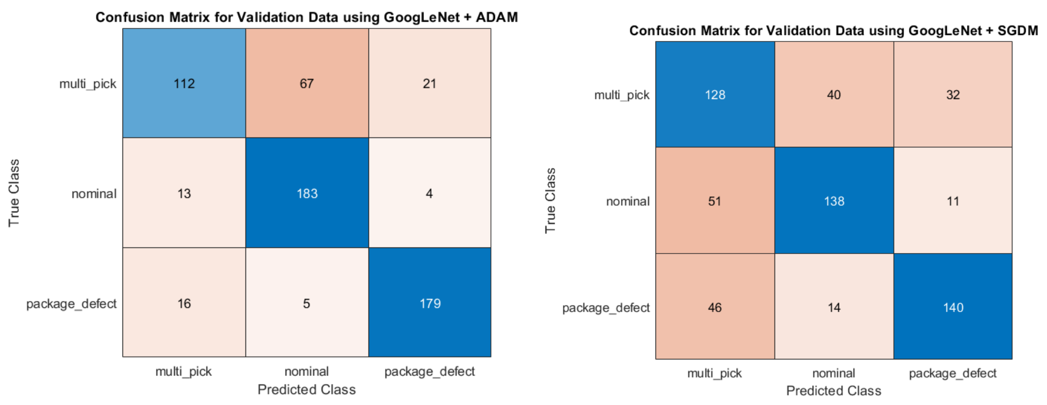


Figure 12. Validation results of using GoogLeNet with ADAM and SGDM

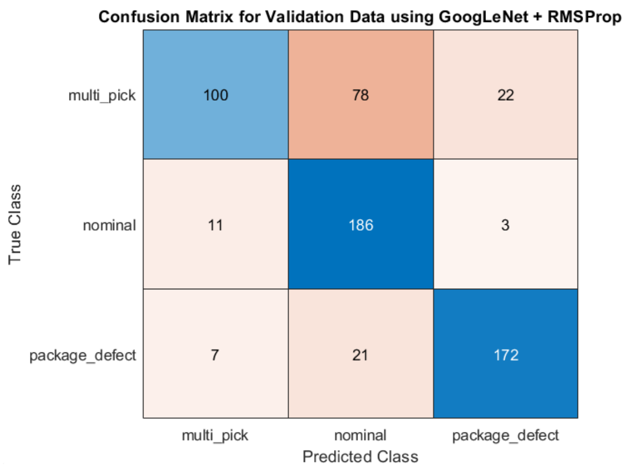


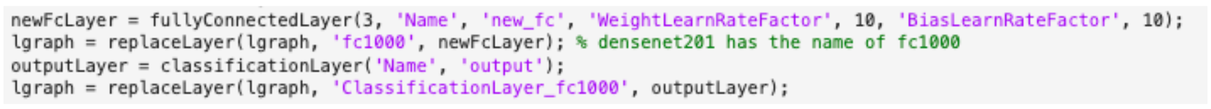
Figure 13. Validation results of using GoogLeNet and RMSProp

Here are the successful classification rates:

- ADAM: (112+183+179)/600=79.0%
- SGDM: (128+138+140)/600=67.6%
- RMSprop: (100+186+172)/600=76.33%

DenseNet

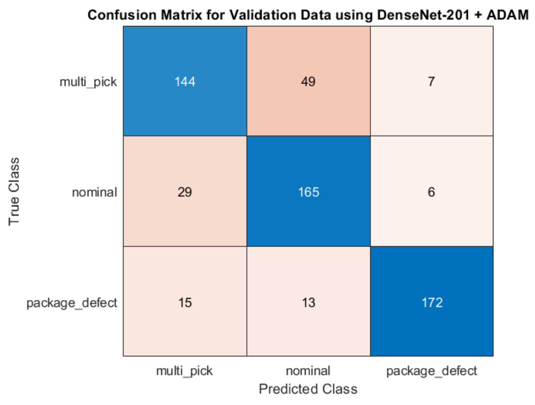
Finally, we are using DenseNet for the transfer learning. And applied the same method by changing the layers we got:



```
newFcLayer = fullyConnectedLayer(3, 'Name', 'new_fc', 'WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10);
lgraph = replaceLayer(lgraph, 'fc1000', newFcLayer); % densenet201 has the name of fc1000
outputLayer = classificationLayer('Name', 'output');
lgraph = replaceLayer(lgraph, 'ClassificationLayer_fc1000', outputLayer);
```

Figure 14. Modify the layers for DenseNet

Then, applied the 3 training algorithms, we got the following results:



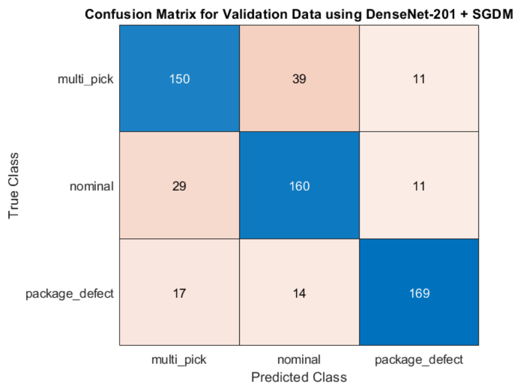


Figure 15. Validation results of using DenseNet with ADAM and SGDM

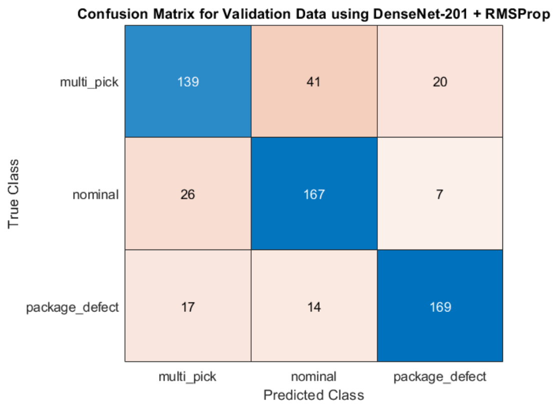


Figure 16. Validation results of using DenseNet and RMSProp

Here are the successful classification rates:

- ADAM: (144+165+172)/600=80.16%
- SGDM: (128+138+140)/600=79.83%
- RMSprop: (100+186+172)/600=79.16%
-

1.3 Results, Analysis and Conclusion

The above implementation procedures provided horizontal and vertical comparison between networks and optimizers. From the experiments, we obtained the following results as shown in the table below:

	ADAM	SGDM	RMSprop
ResNet50	82.5% (5 min 15 sec)	71.0% (4 min 43 sec)	82.0% (4 min 43 sec)
AlexNet	73.5% (1 min 7 sec)	72.7% (59 sec)	72.12% (1 min 9 sec)
GoogLeNet	79.0% (2 min 30 sec)	67.6% (2 min 23 sec)	76.3% (2 min 15 sec)
DenseNet201	80.2% (65 min 8 sec)	79.8% (62 min 15 sec)	79.2% (66 min 15 sec)

From the experiment we did in task, we found that ResNet + ADAM has the highest overall accuracy compared with other combinations of CNN models and algorithms. It is noticeable that the accuracy will not reach 100% due to the lack of convergency and the existence of blurred images as shown in Figure 17.

Figure 17. Blurred images cause low accuracy

Considering the fact that the model might be used on a larger delivering sorting project which might involve a larger number of objects to be classified, firms might consider two factors that determines the choice of combinations. A possible decision matrix is illustrated in the following table.

Criteria	Time Dominated Wt(%)	Accuracy Dominated Wt(%)	Network Alternatives							
			AlexNet		GoogLetNet		ResNet		DenseNet	
Time consumption	70	15	5		4		3		1	
Accuracy	15	70	3		4		5		5	
Resource utilization	15	15	4		4		5		2	
Total			12		12		13		8	
Weighted Total			4.55	3.45	4	4	3.6	4.7	1.75	3.95

Figure 18. Decision Matrix Table

The ratings indicated in the red boxes demonstrates the highest rated model in each decision matrices. When a firm prefers time dominated approach, they might use AlexNet+SGDM while an accuracy dominated firm might choose ResNet+ADAM instead.

Task 2: Gripper Design

Robotic manipulation has emerged as an active research area due to its broad applications across various fields, including the picking and transferring of items in express delivery package sorting. Specifically, during the process of lifting and placing packages into boxes, two primary defects induced by the robot manipulator are **multi-pick** and **package-defect**. The former refers to the instance where more than one item is picked up at a time, while the latter refers to any damage inflicted on the package.

Motivated by such a background, there are several ways to minimize the likelihood of these issues. For instance, at process level, we can manually inspect the integrity of an item's packaging and check for any damage prior to the pick and place operation. Alternatively, at the perception and planning level, given the size of the objects, we try to position the gripper on a single item as much as possible, but it's difficult to prevent packaging defects as we only focus on the gripping policy. However, these require advanced sensing algorithms and additional human resources, and are not easily achievable.

Considering the constraints of cost and implement-ability, a feasible approach is to focus on designing robotic grippers (also known as end-effectors). This should be done from a physical standpoint and in conjunction with front-end perception systems to ensure accurate gripper placement. In the following we summarized the main content and contribution of Task 2.

1) We propose a novel design for a robotic gripper that equipped with four flexible, soft fingers and a central movable suction cup. This configuration enables the gripper to adjust its grip according to the specific characteristics of the object it is handling. By modifying the shape (degree of bending) of the four soft fingers, the gripper can more precisely adapt to the object's shape and size, which minimizing the occurrence of multi-pick and package defects.
2) We conduct two sets of simulation experiments with different categories to validate the feasibility of our design.

The remaining content of this task proceeds as following sections: Section I introduces the CAD design of the robotic gripper and explains the function of each part, and detail how they work together. Section II focuses on the working flow and simulation experiments, primarily discussing how our gripper avoids defects. It also discusses an overview of the whole process, such as front-end object detection and other integrated techniques.

2.1 CAD design of gripper

The picture above shows main parts of the end-effector. From left to right displayed four parts of the end-effector, soft gripping fingers, supporting frame, servo motor for central suction cup, and the central suction cup.

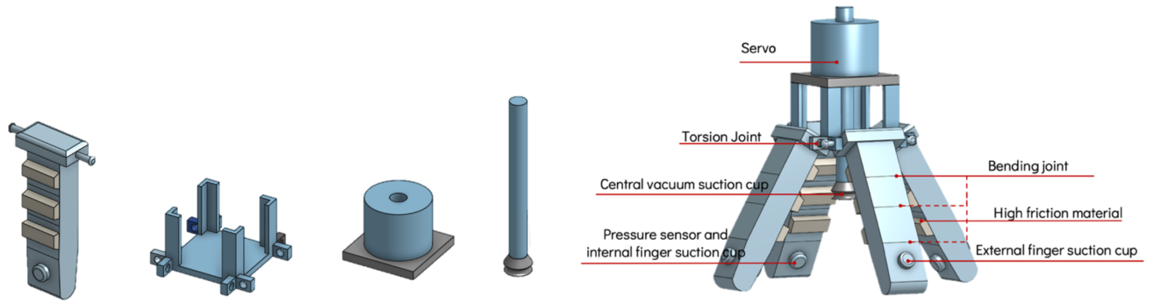

Servo
Torsion Joint
Bending joint
Central vacuum suction cup
High friction material
Pressure sensor and internal finger suction cup
External finger suction cup

Figure 19. Individual Parts of Gripper

The gripper is symmetrically designed, featuring four soft fingers equally spaced around a central vertical axis (direction of gravity). Each finger is composed of a soft inner chamber, encased by a protective layer. This protective layer is lined on the inside with high-friction material. Positioned inside the tip of each soft finger are a piezoresistive pressure sensor and an internal finger suction cup, they are equipped together. On the outer side of each finger is an external suction cup.

Here is the functional description of each part.

- **Piezoresistive pressure sensor:** The piezoresistive pressure sensor in the gripper serves two primary functions. Firstly, it measures the contact force to monitor the gripping force, ensuring that the suction cups do not apply excessive suction that may damage the object. This is important because some image dataset has shown that many defects are caused by overly strong suction. Secondly, the pressure sensor checks that all four fingers are in contact with the object, verifying the status of the four fingers, such as whether there is full contact or pickup.
- **Central movement suction cup：** It is used for the initial pickup of objects, moving vertically and driven by a servo to provide a certain amount of space underneath. This space allows the fingers to fully envelop the object. Additionally, it serves to firmly hold and stabilize the object during lifting or transferring, preventing any unwanted movement.
- **External and internal suction cup on each tip of a soft finger:** Each soft fingertip is outfitted with 2 vacuum suction cups, one on the front and one on the back. It is designed to create the necessary suction for pickup. These cups are connected by vacuum tubes to an external gas circuit for precise control. The principle behind their operation is evacuating air from the tubes to create a pressure differential that enables suction. An important note is that although the vacuum tube is independent, but they share one vacuum source to guaranteeing the stable suction forces.
- **Supporting frame:** The supporting frame is used to assemble the four fingers and also serves as a holder for essential components. This includes circuit components related to the vacuum suckers (e.g., wiring) and space for embedding systems such as microcontrollers and control modules (e.g., servo motors).
- **Servo motor:** Servo motors are used to precisely control the position, orientation, and posture of the fingers, specifically, they adjust the degree of bending in the soft fingers and also facilitate rotation around the torsion joint axis, allowing the fingers to rotate within a defined range. We will discuss more how servo motor can achieve that.

After we have introduced the functions of the essential components, we move on to discuss our soft finger, focusing on how it achieves flexibility in rotation and bend.

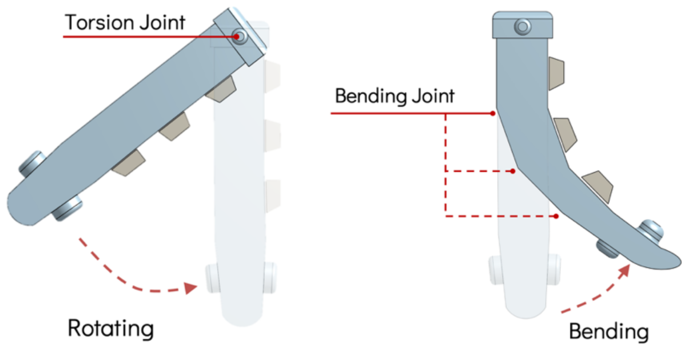


Figure 20. Operation of fingers

There are two different types of movement of the soft finger: *rotating* and *bending*. These two distinct types of motion are key to the flexibility of our gripper, and it is this characteristic that allows our finger to have various different griping modes (configuration/arrangement of four soft fingers).

Rotating: "Rotation" refers to the movement of the soft finger pivoting around the torsion joint as the center, within a limited range, the fingertip creating a trajectory that forms an arc, as shown in the image on the left. This mode of motion is made possible by a servo motor embedded within the upper part of each soft finger, which can precisely control the angle of rotation in accordance with the commands from the upper-level control system.

Bending: The bending of a soft finger is achieved and inspired by the "tendon-driven mechanism" as shown in [1] [2]. This mechanism is similar to the bending of a human finger's tendon shown in Fig.21. Specifically, the bending of four fingers is controlled by the dragging of tendon lines, which are driven by a central servo positioned at the center of the support frame. Finally, the finger is able to achieve different modes of bending deformation. Each tendon line is connected to the central servo's output axis, which is attached to an annular wheel. This wheel, in turn, evenly distributes the force to the tendons linked to each of the four fingers. In shorts, the servo motor activates the wheel, which then pulls on the tendons to achieve the bending of the fingers. Through this mechanism, the soft finger can perform different modes of bending deformation, which allowing for an adaptable and flexible gripping action.

A noteworthy feature is: Due to the material of soft finger(e.g., silicone), it has capability for both *active* and *passive* bending. The finger can actively bend through a tendon-driven mechanism and passively bend outwards or inwards when external forces are applied. This feature allows the finger to adjust its bending degree and direction upon contact with an object, conforming to its shape and hardness and enhancing stability in grip. This enhances the flexibility and adaptability of the soft finger for a range of grasping tasks.

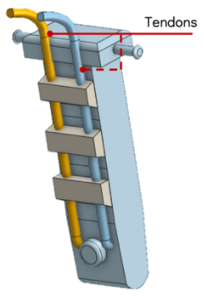


Figure 21. Internal Structure of each finger

2.2 Workflow of System

1. **Object Detection: Segmentation, Bounding, Centroid, and Orientation Identification:** To perform the gripping process, the system needs to know the object's centroid position and orientation based on visual information acquired from an external camera.

   By applying computer vision techniques, we sequentially process the following: Segmentation (to determine the shape and outline of an object), Bounding Box (to define the rectangular boundaries of the object), Centroid (the central point of the object, which aids in positioning the central sucker and facilitates gripping), and Orientation (the direction the object faces, which is critical for optimizing the gripper's orientation to improve grip success rate).
   This process can be visualized as below graph's illustration:

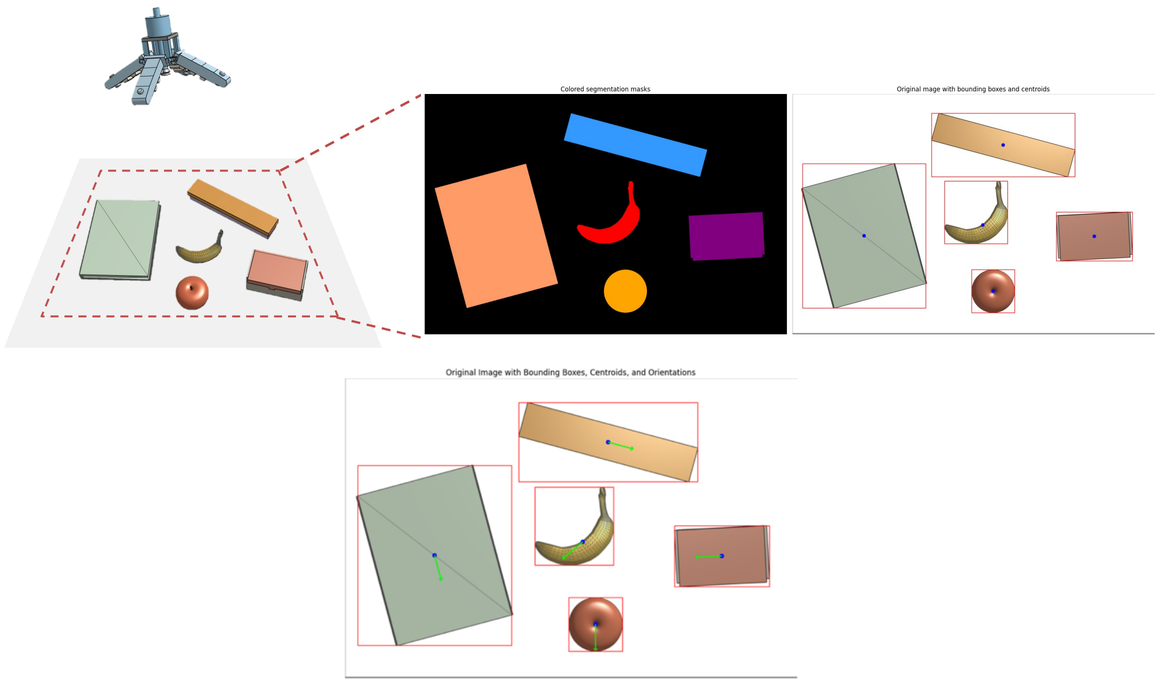

Colored segmentation masks
Original mage with bounding boxes and centroids
Original Image with Bounding Boxes, Centroids, and Orientations

Figure 22. Object detection

Here is the simple description of the method, also we adapt the external library (e.g., OpenCV):

Segmentation: Color-based thresholding and contour detection.
Bounding Box: Minimum rectangle that can enclosing object.
Centroid: Calculated via moments from contour.
Orientation: Derived from minimum area rectangle.

2. Gripping process

After obtaining the centroid position and orientation of objects, the gripping process begins:

1) The robot will first move the gripper to the centroid of the object. Assuming the object has orientation of angle θ degrees, the robot will then rotate the gripper with angle of same magnitude to align the gripper with the object, as displayed in following figure.

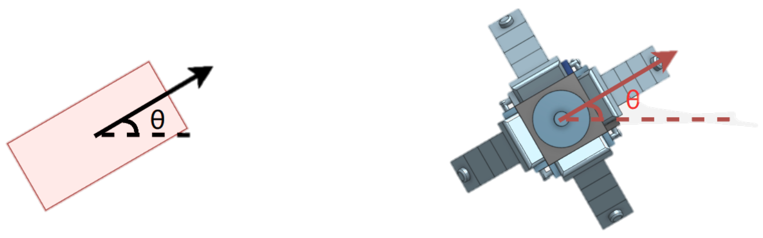


Figure 23. Gripper orientation

The red rectangle represents the object to be picked.

2) After the gripper is aligned with the object. It moves down with a linear motion approaching the object from top. At the same time, controlled by the servo motor, the central suction cup will move down. This process continues until the gripper reaches designated position, where the central suction cup touches the object.

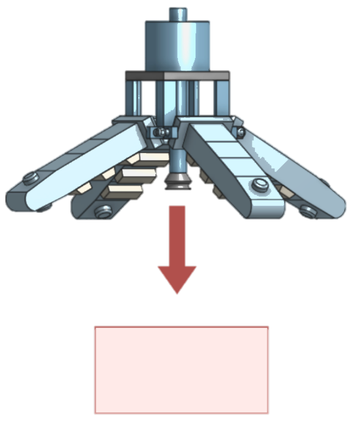


Figure 24. Gripper operation

The black and red arrow indicates the movement of the gripper and the central suction cup respectively.

3) Upon reaching the object, the central suction cup operates and lift the object slightly. At the same time, controlled by servo motors, the four gripping fingers rotates around torsion joints until the value in pressure sensor exceed the threshold (indicates the finger touching the object).

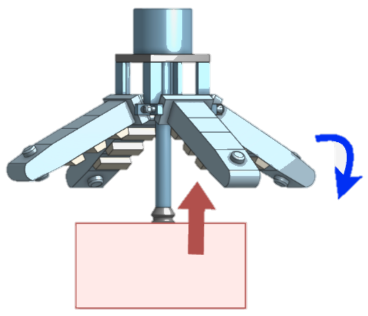


Figure 25. Gripper operation

Red arrow indicates the linear motion of central suction cup. Blue arrow indicates the rotation of gripping fingers.

4) Once the object is lifted, the four gripping fingers curl and hold the object inside. At the same time, the inner suction cups on gripping fingers activate, stabilizing the object.
Below shows the whole gripping process of gripping an apple.

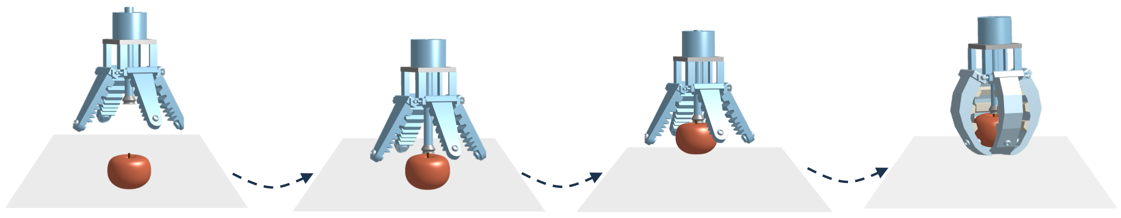


Figure 26. Full gripping process

3. Flexible gripping mode

It is noticeable that the range of size of pickable objects is strictly limited by the span of gripping fingers. In those situations, our gripper could provide alternative gripping modes which sacrifices part of functionalities on multi-pick/package-defect avoidance.

1) Alternative mode 1:

The following figure displayed alternative gripping mode 1 on three different perspectives:

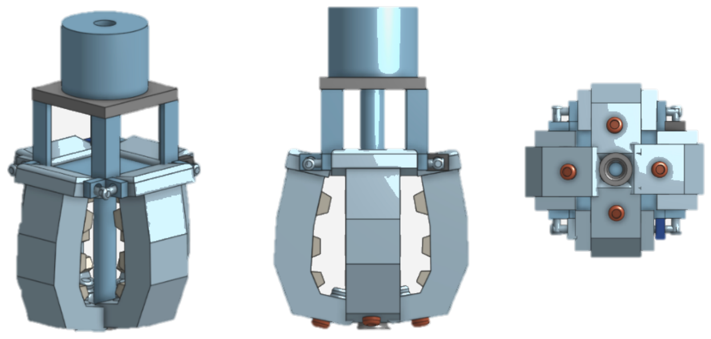


Figure 27. Alternative mode 1

The four gripping finger bends inward and stays still for whole gripping process. The external suction cups on gripping fingers are activated when gripping. The whole gripper works as regular gripping tool with 5 suction cups, similar to the Amazon robot end-effector shown in ARMBench dataset.

In this operating mode, the gripper would be able to pick up larger objects.

2) **Alternative mode 2:**

The following figure demonstrates alternative mode 2 in three different perspectives:

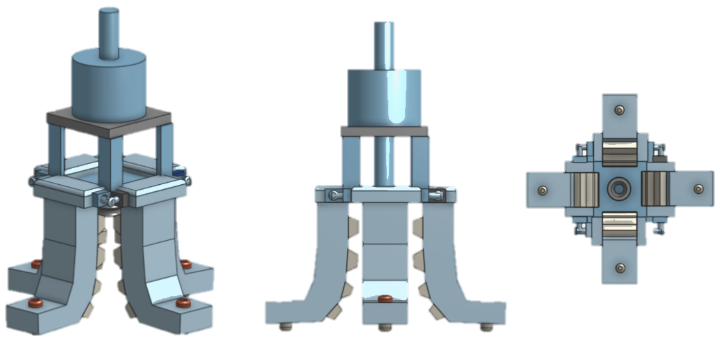


Figure 28. Alternative mode 2

This operating mode is designed for even larger objects where alternative mode 1 unable to pick up the object. In this operating mode, the four gripping fingers bends outwards and activate their internal suction cups. The gripper would be able to perform higher stability gripping for larger and heavier object. However, this would highly increase the possibility of multi-picking.

4. Demostration#1: Multi-pick avoidance

One important feature of our gripper is multi-pick avoidance. This is mainly achieved by the central suction cup. The central suction cup is designed to have a small size, which greatly reduces the chance of gripping multiple objects at the same time. With assistance of object detection algorithm, the gripper can fully avoid multi-picks.

Here shows several examples of multi-picks from ARM Bench dataset:

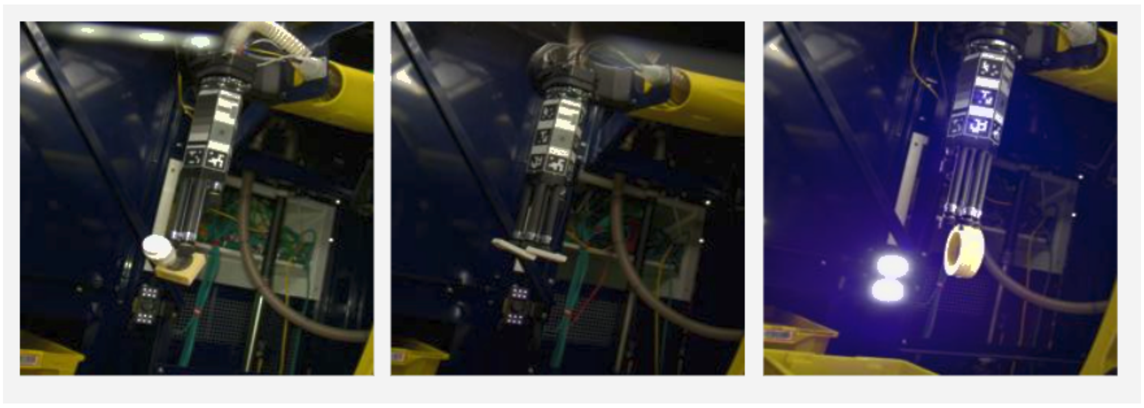


Figure 29. Multi-pick in ARMBench dataset

Their respective CAD models are made:

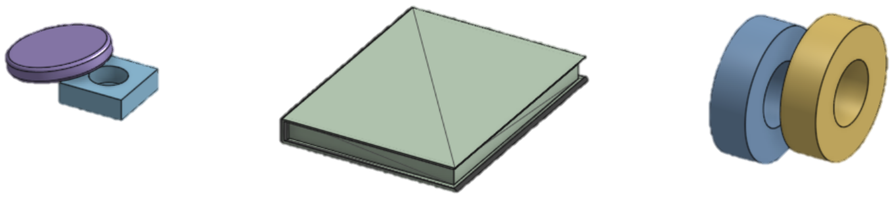


Figure 30. Multi-pick object CAD models

Below demonstrate the scenarios of how the gripper picks above object while avoiding multi-picking:

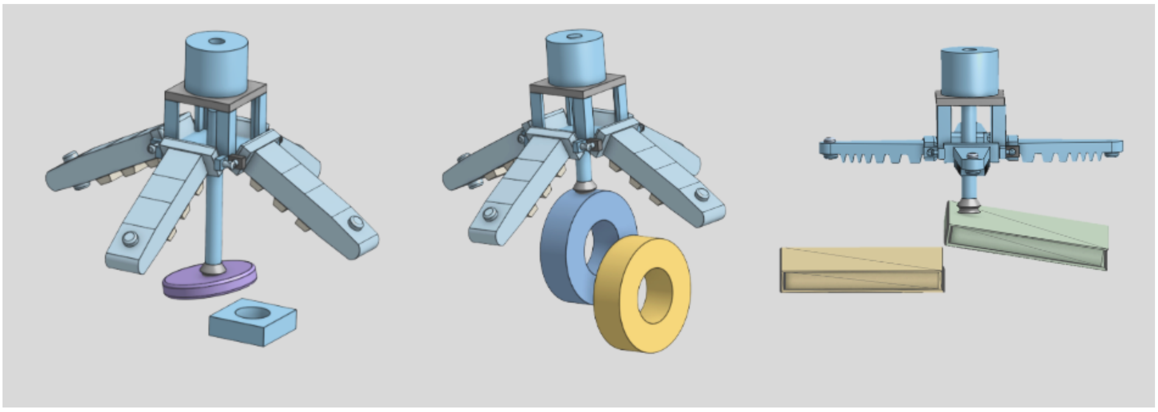


Figure 31. Multi-pick avoiding demonstration.

As shown in above figure, using a small central suction cup guided by accurate object detection algorithm, multi-picking can be eliminated.

5. Demostration#2: Package-defect avoidance

Another feature of our gripper is package-defect avoidance. The function is achieved by utilizing the flexibility of the soft gripping fingers. For defects from books and lidded boxes, the soft gripping fingers would bend inward and hold the box/book from its bottom, preventing it falling off.

Here are examples of package-defects from ARM Bench dataset:

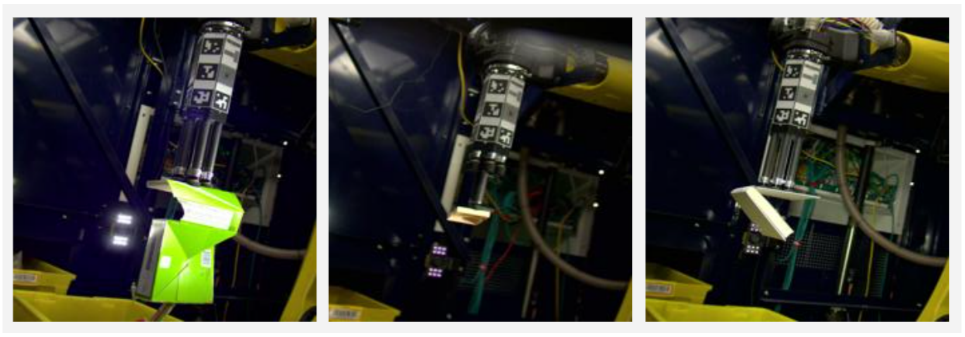


Figure 32. Package-defects in ARMBench dataset

Their respective CAD model is built:

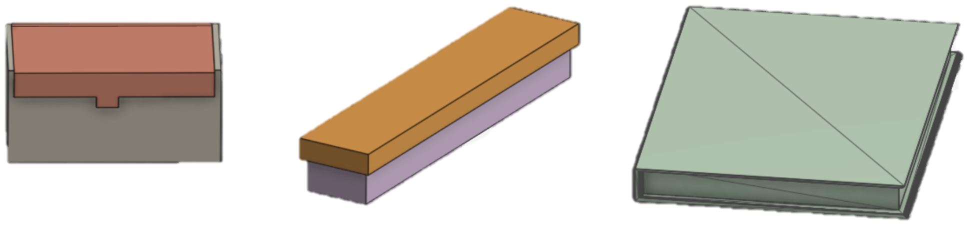


Figure 33. Package-defects objects CAD models

Below demonstrate the scenarios of how the gripper picks above object while avoiding package-defect:

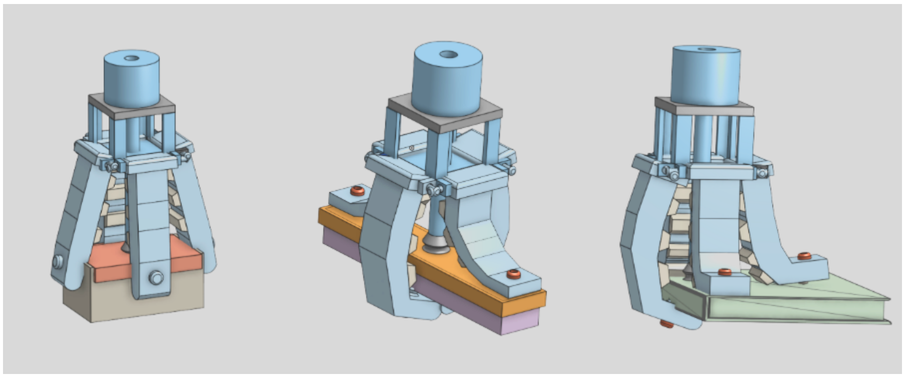


Figure 34. Package-defect avoiding demonstration.

Utilizing soft property of gripping claws, objects with various shape can be held inside the gripper. With assistance of suction cups, objects would be stabilized without any defect.

Conclusion

Based on aforementioned discussions, the image classification technique can accurately sort out the defective packages and multi-pick scenarios can be highlighted for robotics to avoid such abnormalities. The gripper design features flexible junctions and suction cups to ensure pick-up task completion. However, limitations still exist on the construction of CNN models and gripper design and are listed in the following table.

CNN Model	Gripper Design
• ResNet + ADAM, tuned for 3 classes, may not adapt to unforeseen scenarios, or create new classes. • Training on a modified ARMBench dataset with only 1000 images per class might lead to insufficient learning.	• The gripper's effectiveness is limited by its arm span, affecting the range of object sizes it can handle. • Algorithms for automatic object fitting were not developed within the given timescale, complicating design, and manufacturing tasks. • Maintenance costs and the need for regular technology upgrades to accommodate new package designs must be considered.

Consider the above limitations, firms should re-evaluate and tailor their decision matrices as per Figure 18, ensuring alignment with specific operational challenges. Integrating suitable CNN models and optimizers with image sensors in embedded systems is advised for real-time processing, enhancing system responsiveness. Furthermore, incorporating images from robotic arm cameras into the dataset for iterative training will bolster the model's adaptability. This approach ensures continuous learning, keeping the classification model updated with new data patterns, thereby improving its robustness and effectiveness in dynamic environments. This strategy of customization, integration, and continuous improvement addresses the critical need for adaptability and efficiency in automated systems, ensuring they remain effective and relevant in varying operational contexts.

Groupwork

Kun Wang & Yuhan Liu	Task 1
Zihao He & Jinyuan Zhang	Task 2

Reference

[1] B. Fang, F. Sun, L. Wu, F. Liu, X. Wang, H. Huang, W. Huang, H. Liu, and L. Wen, "Multimode Grasping Soft Gripper Achieved by Layer Jamming Structure and Tendon-Driven Mechanism," in *Soft Robot*., vol. 9, no. 2, pp. 233-249, Apr. 2022, doi: 10.1089/soro.2020.0065.

[2] F. Liu, F. Sun, B. Fang, X. Li, S. Sun and H. Liu, "Hybrid Robotic Grasping with a Soft Multimodal Gripper and a Deep Multistage Learning Scheme," in IEEE Transactions on Robotics, vol. 39, no. 3, pp. 2379-2399, June 2023, doi: 10.1109/TRO.2023.3238910.