Human Factors in Design, Engineering, and Computing, Vol. 159, 2024, 1511–1520
https://doi.org/10.54941/ahfe1005719

Advancing Vision-Based Adaptive Gripping Technology With Machine Learning: Leveraging Pre-Trained Models for Enhanced Object Classification

Diptesh Kumar Mandal[1], Kazunori Kaede[1,2], and Keiichi Watanuki[1,2]

[1]Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama-shi, Saitama, 338-8570, Japan
[2]Advanced Institute of Innovative Technology, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama-shi, Saitama, 338-8570, Japan

ABSTRACT

This paper presents a machine learning-based approach aimed at designing an adaptive robotic gripper capable of distinguishing between hard and soft objects. Using the CIFAR-100 dataset, we trained a deep learning model based on the ResNet50 architecture to classify objects into these two categories using visual data. Extensive data augmentation techniques were employed to enhance the robustness of the model, and the ResNet50 model was fine-tuned for this task. The model achieved a validation accuracy of 80.25% and demonstrated promising results in differentiating hard and soft objects, with implications for various applications in industrial and healthcare settings. Future work will focus on enhancing the robustness and accuracy of the model by utilizing the ImageNet (ILSVRC subset) dataset, applying ensemble methods, and addressing current computational limitations.

Keywords: Adaptive gripper, Deep learning, Resnet50, Object classification, Robotic manipulation, Data augmentation, AdamW optimizer

INTRODUCTION

Robotic systems are increasingly deployed in dynamic and unstructured environments where they must interact with a wide variety of objects. The ability to adaptively manipulate objects based on their physical properties, such as hardness and softness, is crucial for versatility and effectiveness of these systems. Traditional robotic grippers often rely on pre-programmed parameters or specific sensor inputs to determine how an object is to be grasped. However, these approaches can be limited in their adaptability and may not perform well in environments where the properties of objects are not known in advance.

The motivation for this study stems from the need to develop an intelligent and adaptable gripper system that can autonomously adjust the grip based on object classification. By leveraging advances in machine learning, specifically deep learning, we aim to create a gripper that can distinguish between hard and soft objects using visual data alone. This approach has the potential to simplify the design of robotic systems, reduce the need for complex sensor arrays, and broaden the range of objects that can be manipulated effectively.

The study is useful in a variety of fields where the focus lies on object handling. In everyday life, we encounter a wide array of objects with varying degrees of hardness, that require different levels of care and precision during handling. For example, in manufacturing, an assembly line might deal with delicate components such as glass or soft materials that demand a gentle grip, while also handling harder objects such as metal parts that require a firm grasp. In healthcare, robotic systems assist in surgeries and patient care, where the ability to delicately manipulate soft tissues and securely hold surgical instruments is crucial.

This paper presents the development of a deep learning model based on the ResNet50 architecture, trained on the CIFAR-100 dataset, to classify objects as either hard or soft. To improve the generalization of the model to new objects, we employed various data augmentation techniques and fine-tuned the model for this specific task. The results of this study demonstrate the potential of using visual data to generate robotic gripping strategies, laying the groundwork for future developments in this area.

BACKGROUND, VISION AND RELATED WORK

Robotic manipulation has undergone significant advancements over the past few decades, particularly in the development of adaptive grippers. These grippers are designed to handle a wide range of objects by adjusting their grip based on sensory feedback. However, most traditional approaches rely on tactile sensors, force feedback, or other specialized hardware to detect the properties of objects and adjust the grip accordingly.

For instance, Calandra et al. (2018) developed a deep learning model that used tactile data to predict grasp outcomes, enabling a robotic hand to identify objects and adjust its grip to prevent slippage or excessive forces. Similarly, the paper "Design and performance characterization of a soft robot hand with fingertip haptic feedback for teleoperation", by Li et al. (2020) focuses on designing and characterizing a soft robotic hand with fingertip haptic feedback for teleoperation emphasizing real-time tactile sensing and feedback mechanisms. Although innovative, these studies often rely on hardware that may not be feasible for all applications, particularly in scenarios where sensory inputs are restricted or unavailable.

This study differentiates itself by focusing on the use of visual data for object classification. Visual data offer several advantages over tactile sensors, including the ability to gather information from a distance, reduced hardware complexity, and potential for integration with existing computer vision systems. By training a deep learning model on a large and diverse dataset such as CIFAR-100, we aim to create a gripper that can adapt to a wide range of objects without the need for specialized sensors.

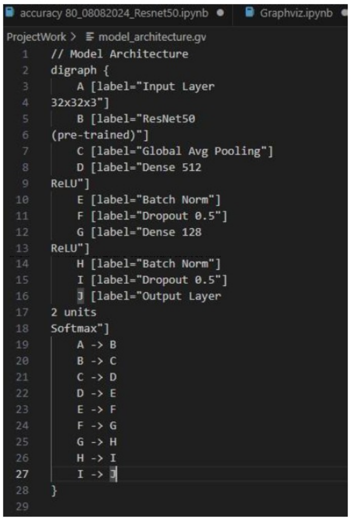
The ability of robotic systems to adaptively grip objects without constant human intervention or extensive reprogramming represents a significant advancement. By relying on visual data and machine learning algorithms, as demonstrated in this research, robotic systems can become more autonomous and versatile, reducing the burden on human operators and improving overall efficiency. This study provides a scalable solution that can be implemented across various industries, to enhance the safety, reliability, and effectiveness of robotic systems in diverse environments.

METHODOLOGY

Data Selection and Preparation - The CIFAR-100 dataset was selected for this study because of its diversity and the availability of labelled images. The dataset contains 60,000 color images across 100 object categories, with each image measuring 32×32 pixels. These categories are manually divided into two classes: hard objects and soft objects. Hard objects include items such as rocks, metals, vehicles and tools, whereas soft objects include animals, fruits, and other organic materials. The list mentioned here is not exhaustive.

To facilitate model training, the dataset was split into training and testing sets, with 50,000 images used for training and 10,000 images reserved for testing. The images were normalized to the range of [0, 1] by dividing the pixel values by 255. This normalization helped accelerate the convergence of the model during training and improved training and validation accuracy.

Model Architecture - We used the ResNet50 architecture as the backbone of our model. ResNet50, a deep convolutional neural network (CNN) with 50 layers, is well-known for its ability to handle complex image classification tasks. It was pre-trained on the ImageNet dataset, providing a robust starting point for our specific classification task. The pre-trained model was modified by unfreezing the last 200 layers and, allowing the network to learn features specific to hard and soft object classification tasks.


accuracy 80_08082024_Resnet50.ipynb
Graphviz.ipynb
ProjectWork > model_architecture.gv

```
// Model Architecture
digraph {
    A [label="Input Layer
32x32x3"]
    B [label="ResNet50
(pre-trained)"]
    C [label="Global Avg Pooling"]
    D [label="Dense 512
ReLU"]
    E [label="Batch Norm"]
    F [label="Dropout 0.5"]
    G [label="Dense 128
ReLU"]
    H [label="Batch Norm"]
    I [label="Dropout 0.5"]
    J [label="Output Layer
2 units
Softmax"]
    A -> B
    B -> C
    C -> D
    D -> E
    E -> F
    F -> G
    G -> H
    H -> I
    I -> J
}
```

(A) Input layer accepts images of size 32x32 pixels with colour channels (RGB).
(B) ResNet50 is a CNN architecture that is Pre-trained on ImageNet. It extracts features from the input images.
(C) Global Avg. Pooling reduces the outpu dimensionality from the ResNet50 model by averaging over spatial dimensions.
(D) Dense 512 with ReLU activation introduces non-linearity and learns representations from the pooled features.
(E), (H) Batch Normalisation improves training, stability and convergence.
(F), (I) Dropout 0.5 to prevent overfitting by randomly setting 50% input nodes to zero during training process.
(G) Dense 128 ReLU.
(J) Output Layer with SoftMax activation and binary classification to 2 classes.

Figure 1: Pictorial representation of the architecture of the deep learning model (Screenshot taken from Jupyter notebook).

This architecture was chosen to balance the model complexity and extract high-level features from the images, ultimately improving the classification accuracy.

Data Augmentation- Data augmentation is a crucial part of this study to ensure that the model can generalize well to new images. Given the limited size of the CIFAR-100 dataset, augmentation techniques were employed to artificially expand the dataset by generating variations of the existing images. The following augmentation techniques were used.

- Rotation (up to 30°): To simulate different orientations of objects.
- Width and Height Shifts (up to 30%): To account for objects appearing at different positions within the image frame.
- Shearing (up to 30%): Mimics the effect of perspective changes.
- Zooming (up to 30%): Simulates objects being viewed from different distances.
- Horizontal Flipping: To generate mirror images, accounting for symmetry.
- Fill Mode (nearest): Handles new pixel values generated during transformation.

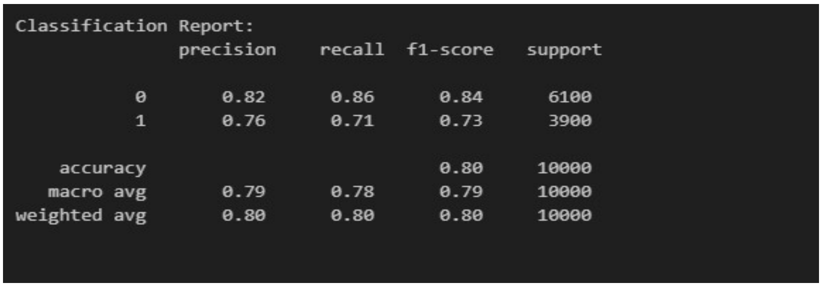
These augmentations make the model invariant to common transformations, thus improving its ability to correctly classify objects in diverse scenarios.

Training Procedure - The model was trained using the AdamW optimizer, an advanced variant of the traditional Adam optimizer that integrates weight decay regularization. This feature is particularly beneficial for preventing overfitting, as it penalizes large weight values, thereby encouraging the model to learn more generalized and robust features. The initial learning rate was set to 0.0001, with a weight decay parameter of 0.0001, ensuring a balanced approach to both learning speed and regularization.

To further refine model training, a learning rate scheduler was implemented using the ReduceLROnPlateau callback in TensorFlow. This scheduler dynamically reduces the learning rate by a factor of 0.5 whenever the validation loss fails to improve after 10 epochs. This mechanism is crucial for allowing the model to continue learning at a more granular level as it approaches convergence, thereby premature stagnation during the training process. In addition, early stopping was used to safeguard against overfitting. Training was automatically halted if the validation loss did not improve for 50 consecutive epochs, indicating that the model had reached its optimal performance on the validation set. This approach not only prevents unnecessary training but also ensures that the performance of the model on unseen data is maximized. The model was trained for over 120 epochs with a batch size of 64, striking a balance between computational efficiency and model performance. To enhance the generalization capabilities of the model, a data generator was used to apply on-the-fly data augmentations, such as rotations, shifts, and flips. This ensured that each epoch was trained on a slightly different set of images, effectively simulating a more diverse dataset and further reducing the risk of overfitting.

RESULTS

The performance of the model was evaluated using the test set, which consists of images not used during training. The results indicate that the model was able to classify hard and soft objects with a high degree of accuracy, achieving a validation accuracy of 80.25%. The classification report (Figure 2) provides additional insights into the performance of the model across the two classes.



```
Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.86      0.84      6100
           1       0.76      0.71      0.73      3900

    accuracy                           0.80     10000
   macro avg       0.79      0.78      0.79     10000
weighted avg       0.80      0.80      0.80     10000
```

Figure 2: Classification report of the program. The screenshot of the code has been taken from Jupyter notebook.

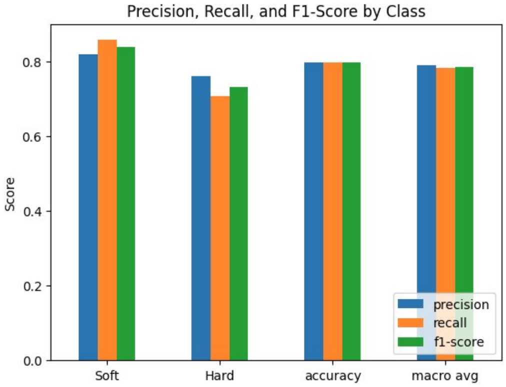


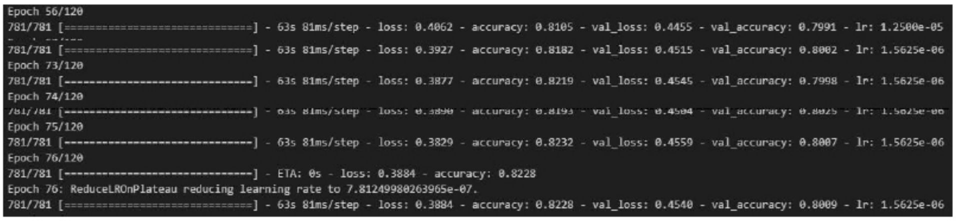
Figure 3: Precision, recall, and F1 score by class for the machine learning model.

For the soft class, precision, recall, and F1-score were all close to 0.83, indicating that the model performed well in identifying soft objects, by optimally balancing between the number of correctly identified positives (recall) and the correctness of its positive predictions (precision).

For the hard class, the precision, recall, and F1-score were slightly lower, around 0.75–0.77, showing that the model faces more difficulty correctly identifying hard objects compared to soft ones.

Finally, the macro average suggests balanced performance across both classes, with all three metrics (precision, recall, and F1-score) aligning around 0.79–0.80. This supports the observation that the model is not biased towards any particular class, but has a slightly better performance on soft objects (Figure 3).

The performance of the trained model was evaluated on the test set. The results (Figure 4) indicated, a validation accuracy of 80.25% with a minimum validation loss of 0.4455. The model demonstrated a strong ability to distinguish between hard and soft objects, as evidenced by its overall accuracy and classification metrics.



```
Epoch 56/120
781/781 [==============================] - 63s 81ms/step - loss: 0.4062 - accuracy: 0.8105 - val_loss: 0.4455 - val_accuracy: 0.7991 - lr: 1.2500e-05
781/781 [==============================] - 63s 81ms/step - loss: 0.3927 - accuracy: 0.8182 - val_loss: 0.4515 - val_accuracy: 0.8002 - lr: 1.5625e-06
Epoch 73/120
781/781 [------------------------------] - 63s 81ms/step - loss: 0.3877 - accuracy: 0.8219 - val_loss: 0.4545 - val_accuracy: 0.7998 - lr: 1.5625e-06
Epoch 74/120
781/781 [------------------------------] - 63s 81ms/step - loss: 0.3890 - accuracy: 0.8193 - val_loss: 0.4504 - val_accuracy: 0.8025 - lr: 1.5625e-06
Epoch 75/120
781/781 [------------------------------] - 63s 81ms/step - loss: 0.3829 - accuracy: 0.8232 - val_loss: 0.4559 - val_accuracy: 0.8007 - lr: 1.5625e-06
Epoch 76/120
781/781 [------------------------------] - ETA: 0s - loss: 0.3884 - accuracy: 0.8228
Epoch 76: ReduceLROnPlateau reducing learning rate to 7.81249980263965e-07.
781/781 [==============================] - 63s 81ms/step - loss: 0.3884 - accuracy: 0.8228 - val_loss: 0.4540 - val_accuracy: 0.8009 - lr: 1.5625e-06
```

Figure 4: Screenshot of a part of the epoch score during model training. Minimum val_loss achieved is 0.4455 at epoch 56 and maximum val_accuracy achieved is 0.8025 at epoch 74.

To gain deeper insights into the model's performance, a confusion matrix was plotted (Figure 5). It provided a detailed breakdown of the model's predictions against actual labels. The confusion matrix was particularly useful for understanding the types of errors made by the model, such as whether it was more prone to falsely classifying hard objects as soft or vice versa.

The confusion matrix revealed the following results:

- True Positives (TP): The model correctly identified 5235 soft objects, reflecting high precision for this class.
- True Negatives (TN): 2756 hard objects were correctly classified, showing that the model was also effective in identifying hard objects.
- False Positives (FP): Approximately 865 soft objects were incorrectly classified as hard, suggesting that the model occasionally confuses the two classes when they share similar visual features.
- False Negatives (FN): Around 1144 hard objects were misclassified as soft, indicating that while the model performed well overall, there was room for improvement in minimizing these errors.

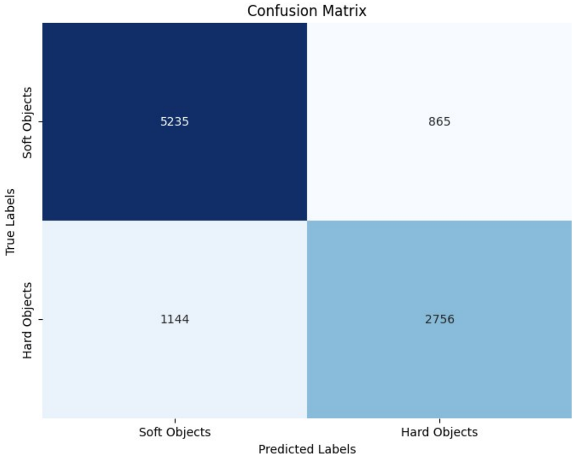


Figure 5: Confusion matrix of the program duly plotting the predicted and true labels. The screenshot has been taken from Jupyter notebook.

Overall, the confusion matrix provided critical insights into the strengths and limitations of the model, highlighting areas where further tuning and data augmentation could potentially lead to performance improvements.

The training process showed a steady decrease in both the training and validation losses, with the best epoch occurring at epoch 74, where the validation accuracy peaked at 80.25%. This suggests that the model has learned to generalize well from the training data, although there is still room for improvement in classifying hard objects.

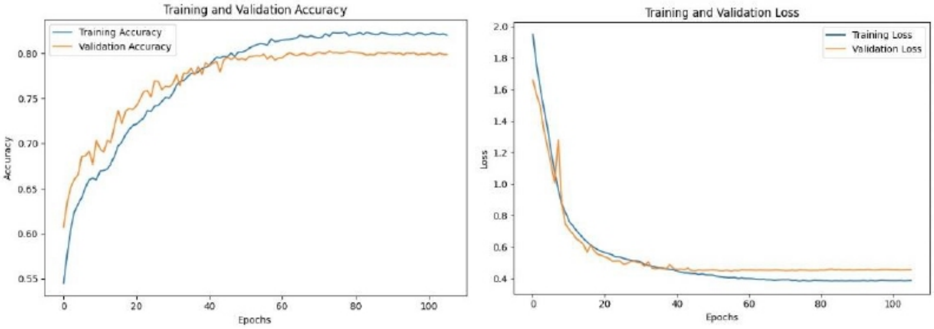


Figure 6: Training and validation accuracy and loss plotted against the number of epochs during model training.

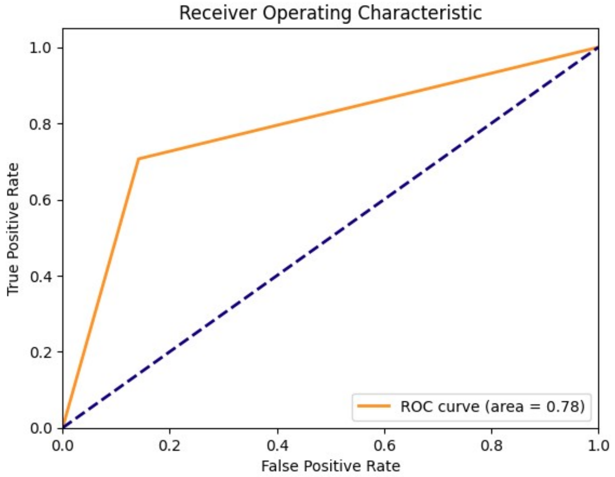


Figure 7: Receiver operating characteristic (ROC) curve with an AUC of 0.78.

The Receiver Operating Characteristic (ROC) curve depicted in Figure 7 shows the performance of the machine learning model in distinguishing between the two classes hard and soft objects. The Area Under Curve (AUC) is 0.78 which indicates that the model performs fairly when distinguishing between hard and soft objects using visual data alone, although there remains scope for further improvement.

DISCUSSIONS

The results of this study demonstrated the effectiveness of using deep learning models, (herein) ResNet50, to classify objects based on visual data. The accuracy and F1-scores obtained suggest that the model is well-suited for integration into adaptive gripper systems where sensory inputs may be limited or unavailable.

However, this study also highlights certain challenges, particularly in the classification of hard objects. The lower precision and recall values for the hard object class suggest that the model may struggle to distinguish between hard and soft objects when the visual differences are subtle. This could be due to the smaller number of hard object examples in the training set, as well as the inherent difficulty of the task resulting from the type of images encountered by the model during the training process. In some cases the model might come across very simple images like a metal rod kept on a plain background but on other instances it might encounter a similar metal rod being placed on top of a soft toy making the image inherently complex and the model may confuse itself.

Data augmentation plays a crucial role in improving the robustness of the model. By simulating real-world variations in the training data, we were able to train a model that was more resilient to changes in object orientation, position, and scale. This is important for applications where the objects being manipulated may vary significantly in appearance.

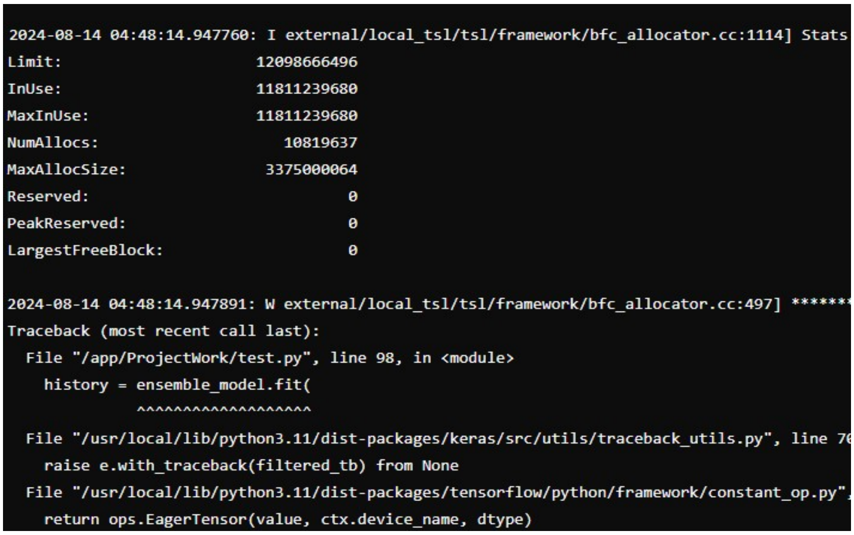
FUTURE WORK AND LIMITATIONS

Although the current model performed well on the CIFAR-100 dataset, there are several areas for improvement that will be addressed in future. One of the most significant steps will be to train and validate the model using the ImageNet (ILSVRC subset) dataset. This dataset is much larger and more diverse than CIFAR-100, containing over 1.2 million images across 1000 categories. By increasing the sampling set, overfitting can be reduced and accuracy of the model can be boosted. Additional diversity in the ImageNet dataset will also help the model to learn more nuanced features that can better distinguish between hard and soft objects.

In addition to expanding the dataset, we plan to experiment with ensemble methods to improve the performance of the model further. Ensemble methods involve training multiple models and combining their predictions to produce the final output. This approach can often lead to better generalization, because it reduces the likelihood of any single model's biases affecting the final prediction. Techniques, such as bagging, boosting, and stacking will be explored in future experiments.

Further data augmentation techniques will also be employed to make the model even more robust. For example, we plan to use techniques such as CutMix, which combines multiple images to create new training examples, and MixUp, which blends the pixels of two images to generate new training examples. These advanced augmentation methods can help the model learn more complex patterns and improve its ability to generalize to new data.

We tried using ensemble methods to train the model and enhance the system accuracy. For this purpose, we used ResNet50, InceptionV3 and Xception as the base model combinations. InceptionV3 and Xception accept image inputs of size $75 \times 75 \times 3$ pixels or more. However, the current limitation of this study is the limited computational power available in our laboratory. The NVIDIA GeForce GTX TITAN X GPU (Compute Capability 5.2) used in this study struggles to handle larger image sizes of 75 × 75 × 3, leading to memory allocation errors during training. This limitation hindered our ability to experiment with larger models and more complex architectures, as evidenced by the error (Figure 8) encountered during training.



```
2024-08-14 04:48:14.947760: I external/local_tsl/tsl/framework/bfc_allocator.cc:1114] Stats
Limit:                    12098666496
InUse:                    11811239680
MaxInUse:                 11811239680
NumAllocs:                   10819637
MaxAllocSize:              3375000064
Reserved:                           0
PeakReserved:                       0
LargestFreeBlock:                   0

2024-08-14 04:48:14.947891: W external/local_tsl/tsl/framework/bfc_allocator.cc:497] ******
Traceback (most recent call last):
  File "/app/ProjectWork/test.py", line 98, in <module>
    history = ensemble_model.fit(
              ^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/keras/src/utils/traceback_utils.py", line 7
    raise e.with_traceback(filtered_tb) from None
  File "/usr/local/lib/python3.11/dist-packages/tensorflow/python/framework/constant_op.py",
    return ops.EagerTensor(value, ctx.device_name, dtype)
```

Figure 8: Extract from the python terminal. Memory allocation error encountered when training the deep learning model with ensemble methods involving image sizes of 75 x 75 x 3.

Efforts are underway to secure more powerful computational resources to facilitate future experiments using larger and more complex datasets. Upgrading to a more modern GPU with higher memory capacity will allow us to train larger models, process higher resolution images, and explore more advanced deep learning techniques. Additionally, the use of cloud computing resources or high-performance computing (HPC) clusters may be considered to overcome these limitations.

CONCLUSION

This paper presents a novel approach for adaptive robotic gripping by leveraging deep learning models trained on visual data to classify objects based on their hardness or softness. The results obtained from the model make it a promising candidate for deployment in various industrial and healthcare applications after due deliberation and further improvement in research.

The novelty of this research lies in its focus on visual data alone for object classification in robotic gripping, which contrasts with previous research that predominantly relied on tactile feedback or specialized hardware. This approach significantly broadens the applicability of adaptive gripping technologies to industries where advanced tactile sensors might not be available or feasible. The use of standard deep learning techniques, combined with data augmentation and fine-tuning of pre-trained models, showcases a method that is both accessible and scalable, offering practical solutions for real-world applications in manufacturing, healthcare, and beyond. The proposed methodology not only enhances operational efficiency but also paves the way for more versatile and human-aware robotic systems. In addition, this approach can lead to cost savings by reducing the need for specialized hardware, such as tactile sensors, which are often expensive and complex to integrate. By utilizing existing visual data and open-source datasets like CIFAR-100 and ImageNet, the method proposed in this study offers an affordable and accessible alternative solution to the problem.

Future work will focus on expanding the dataset to include more diverse examples, employing ensemble methods to improve generalization, and addressing current computational limitations. The goal is to develop a highly adaptable gripper system that can handle a wide variety of objects in dynamic and unstructured environments, ultimately enhancing the capabilities of robotic systems in both industrial and everyday applications such as manufacturing, logistics, healthcare, and beyond.

REFERENCES

AboZaid, Y.A., Aboelrayat, M. T., Fahim, I. S., Radwan, A. G. (2024). Soft Robotic Grippers: A Review on Technologies, Materials, and Applications. Sensors and Actuators A: Physical. 372. 115380. 10.1016/j.sna.2024.115380.

Calandra, R., et al. (2018). "The Feeling of Success: Does Touch Sensing Help Predict Grasp Outcomes?" IEEE Transactions on Robotics, 34(6), 1496–1505.

Chi, Y., Zhao, Y., Hong, Y., Li, Y., & Yin, J. (2023). "A Perspective on Miniature Soft Robotics: Actuation, Fabrication, Control, and Applications." Advanced Intelligent Systems, 6(2), 2300063. https://doi.org/10.1002/aisy.202300063

Gjerstad, T. B., Lien, T. K., & Buljo, J. O. (2006). "Handle of Non-rigid Products Using a Compact Needle Gripper," in Proceedings of the 39th CIRP International Seminar on Manufacturing Systems, 145–151.

Liu, Y., Hou, J., Li, C., & Wang, X. (2023). "Intelligent Soft Robotic Grippers for Agricultural and Food Product Handling: A Brief Review with a Focus on Design and Control." Advanced Intelligent Systems, 5(12), 2300233. https://doi.org/10.1002/aisy.202300233

Li, M., Zhuo, Y., Chen, J., He, B., Xu, G., Xie, J., Zhao, X., & Yao, W. (2020). Design and performance characterization of a soft robot hand with fingertip haptic feedback for teleoperation. *Advanced Robotics*, *34*(23), 1491–1505. https://doi.org/10.1080/01691864.2020.1822913

Yao J, Fang Y, Li L. (2023) Research on effects of different internal structures on the grasping performance of Fin Ray soft grippers. *Robotica*. 41(6), 1762–1777. DOI: 10.1017/S0263574723000139.