2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

A Self-Attention Multi-Task Learning Model for Garment Segmentation and Parts Recognition

Yilin Zhang[1], Alberto Petrilli[2], Naoya Chiba[1], and Koichi Hashimoto[1]

Abstract— **The integration of robotics in the garment industry remains relatively limited, primarily due to the challenges in the highly deformable nature of garments. This study thus explores a vision-based garment and garment parts recognition model to facilitate the application of robots in garment manipulation. The main objective is to detect and segment each garment piece from a random table and provide multi-dimensional information on it, as well as recognize garment parts such as collar to facilitate proposing grasping points for various robotic tasks. In order to achieve this goal, an MTL (Multi-Task Learning) model based on YOLOv8 and HyCTAS's self-attention head was processed. Transfer learning was applied and the model was fine-tuned and tested on a self-collected dataset as well as an open-source garment dataset Fashionpedia. Experiment results demonstrate that this MTL model is able to substantially improve the processing speed while having a minimal decrease in mask average precision for each integrated vision task. And while this performance preservation is mainly attributed to the HyCTAS implementation, further enhancements can be achieved by adding auxiliary tasks and loading weights from single tasks.**

I. Introduction

The garment processing industry, despite witnessing widespread adoption of robotics across various sectors, has been slow to embrace this technological revolution. This reluctance can be attributed primarily to the inherent challenges posed by the highly deformable nature of garments, which frequently present unprecedented shapes during processing, making it difficult for robots to comprehend, generalize, and respond effectively.

While some research endeavors [1], [2], [3], [4], [5] have ventured into the domain of robotic garment handling and manipulation of deformable objects, they are often limited in scope, addressing specific operations within controlled environments [1], [2], or constrained to single slightly crumpled garment [2], [3], [4] or square-shape cloth [5]. However, in a garment producing factory, more versatile garment handling tasks and more diverse initial garment states are usually observed. Consequently, a more generalized model is needed that can recognize garment pieces and provide multiple information on them, e.g., color, states, material, etc., and subsequently propose grasping points based on the robot task and garment's state. This study thus explores a vision-based model that could provide a possible solution. The envisioned model is designed to segment t-shirts from a possible mix-up pile and recognize them by 6 defined states, predict whether the t-shirt is on top layer or being occluded by other t-shirts, and recognize t-shirt components after obtaining the individual t-shirt pieces in real-time. Additionally, the model should be able to adapt easily when a new vision task is added.

The goal of segmenting t-shirts or t-shirt parts can be achieved by instance segmentation, which can be seen as per-pixel classification of an image. While Segment Anything Model (SAM) [6] had shown superior ability on this task, it was hampered by inference speed, lack of semantic meaning, and the difficulty in training. YOLO series have been famous for their real-time inference speed, high level accuracy, as well as the integration ability. YOLOv8 [7], as one of the most recent developments, has demonstrated very good performance on vision tasks including instance segmentation in many fields. And it provides rich publicly available weights that can be used as a base for transfer learning. However, using traditional instance segmentation methods like YOLOv8 for multiple tasks in the same scene would require multiple models – one for each task. And this deployment would substantially increase the inference time.

In addressing models that predict multi-dimensional results from single-modality inputs, multi-label neural networks [8] or Multi-Task Learning (MTL) architectures [9], [10], [11], [12], [13], [14] are typically considered. However, the former is mostly constrained to classification tasks and would demand an overall change on the trained model and the dataset when new tasks are introduced. MTL models are promising for these needs because of its inherent inductive transfer ability, which allows knowledge learned from one task to improve the performance of another related task by sharing useful features. However, current MTL models that efficiently balance accuracy and speed are usually limited to bounding box detection tasks and single-class segmentation tasks [13], [14], as multi-class instance segmentation presents more difficulty in achieving comparably high performance while maintaining processing speed.

Self-attention [15] is a learning mechanism whose ability to focus on relevant information allows improved accuracy. [16] proposes a Hybrid Convolutional-Transformer Architecture Search (HyCATS) framework that uses a combination of self-attention module and convolution module, and search for the best combination in the search space so that it reduces the

*This work was partially supported by the Innovation and Technology Commission of the HKSAR Government under the InnoHK initiative, and by JSPS KAKENHI Grant Number 21H05298.

[1]These authors are with Department of System Information Sciences, Graduate School of Information Sciences, Tohoku University, Aoba-ku, Sendai 980-8579, Japan zhang.yilin.r8@dc.tohoku.ac.jp; chiba@nchiba.net; koichi.hashimoto.a8@tohoku.ac.jp

[2]Alberto Petrilli is with Department of Robotics, Graduate School of Engineering, Tohoku University, Aoba-ku, Sendai 980-8579, Japan petrilli.barcelo.alberto.elias.c3@tohoku.ac.jp

computation cost while preserving self-attention's abundant information. This mechanism could potentially assist the MTL model by letting each task branch focus on their own most relevant features.

To realize our vision, this research adopts an MTL approach based on YOLOv8 and HyCTAS. The model takes in RGB images as input, and processes the extracted features by multiple self-attention heads for different vision tasks. To facilitate recognizing t-shirt states, layers, and parts, a dataset of t-shirts on tables was collected. Experiments show that this MTL model is able to achieve a similar result as the independent tasks trained by YOLOv8 while executing with a higher speed than running multiple single models. Additionally, loading single models' weights without training MTL model can potentially achieve a comparable performance as well. To summarize, the main contributions of our work are as follows:

- We designed a framework to facilitate garment handling robots under industrial settings, which recognize and differentiate individual t-shirt from a random pile of t-shirts and identifies their component parts. To support such models, we created a t-shirt dataset.
- We developed an MTL model based on YOLOv8 with integration of the self-attention mechanism of HyCTAS.
- We adopted this MTL model to enable the simultaneous execution of multiple garment related vision tasks, e.g., t-shirt state recognition and layer estimation. On those tasks, the proposed MTL model was able to achieve similar accuracies with faster inference speed compared to executing single task models.

II. Related works

A. Robot vision for garment handling

Robot vision for garment handling continues to attract attention as a challenging topic, yet developments remain limited. [1] employs an edge detection-based method, which detects wrinkles represented by curvilinear structures and evaluate graspability along the line, but the procedure may repeat several times until a garment is fully open, which is time-consuming for an industrial setup; [2] utilizes an encoder-decoder network trained on human labeled dataset, directly predicting two grasping points for each defined action from an RGBD image, but it is constrained to overhead view and limited to single t-shirt handling; [3] uses Auto Encoder-Decoder Network to propose grasping points for garments, [4] matches point clouds to the garment image and learns the grasping points from simulation, but they are only able to handle slightly crumpled single garments; [5] learns a mesh correspondence and infers the 3D geometry from images, but it's constrained to single layer fabrics.

B. Garment dataset

Garment dataset, several of which are publicly available, have been proposed [17], [18], [19]. They offer a variety of garment types and sizes. However, those garments are worn on humans, presenting a large difference in appearance with the ones being processed in a factory, which are mostly on an operating table or plate. [20] proposes a garment dataset featuring garments during different household tasks, and it includes images of garments on table. However, this dataset is yet to be disclosed. Moreover, when it comes to multiple garments, they either are placed with considerable gaps or have notably different colors when touching boundaries. Thus, dataset for garments under complex industrial settings remain unsolved.

C. Multi-Task Learning models

Multi-Task Learning models stand out for their generalizability and high efficiency. And these are important for the garment recognition task. The most common way of implementing MTL models is through hard parameter sharing [9], [12], [13], [14]. These models share the hidden layers between all tasks and keep a few task-specific layers as their respective output heads. Hard parameter sharing MTL models are usually more robust to overfitting. Some other MTL models [10], [11] preserve a separate feature extraction network for each task and allow information flow between parallel layers of these networks. These methods are known as soft parameter sharing. [10] introduces the Sluice network that applies linear combination of the outputs of each task's previous parallel layer so that each layer has one more choice to focus on – the integrated features. [11] uses a 1 x 1 convolution instead of linear combination to integrate the shared features. These soft parameter sharing methods usually consume more execution time. Recently, more works using MTL models have emerged in the fields of autonomous driving [12], [13], [14]. However, their tasks are limited to object detection and single-class semantic segmentation. In addition, the specialized loss function for each task increases the difficulty of adding new tasks.

III. Methods

In this section, the details of this work are presented, which includes the architecture of the MTL model, the single tasks that are combined into the multi-task and their corresponding datasets.

A. MTL model

In this work, we employed a hard parameter sharing manner. As shown in Fig.1, our MTL model shares one feature extraction backbone as the encoder, and utilizes several task-specific neck and head as the decoder. The number of the neck and head depends on the number of tasks incorporated into the MTL model, and the available task types are object detection, instance segmentation, and pose estimation. We followed YOLOv8's modular coding structure, allowing easy and independent adjustments or updates of the model's backbone, neck, and head. Currently for our t-shirt recognizing work, we followed the backbone and neck structure of YOLOv8 and integrates the combined self-attention head of HyCTAS.

1) Backbone and neck

We use the same backbone and neck as YOLOv8 so that transfer learning can be applied from YOLOv8's rich

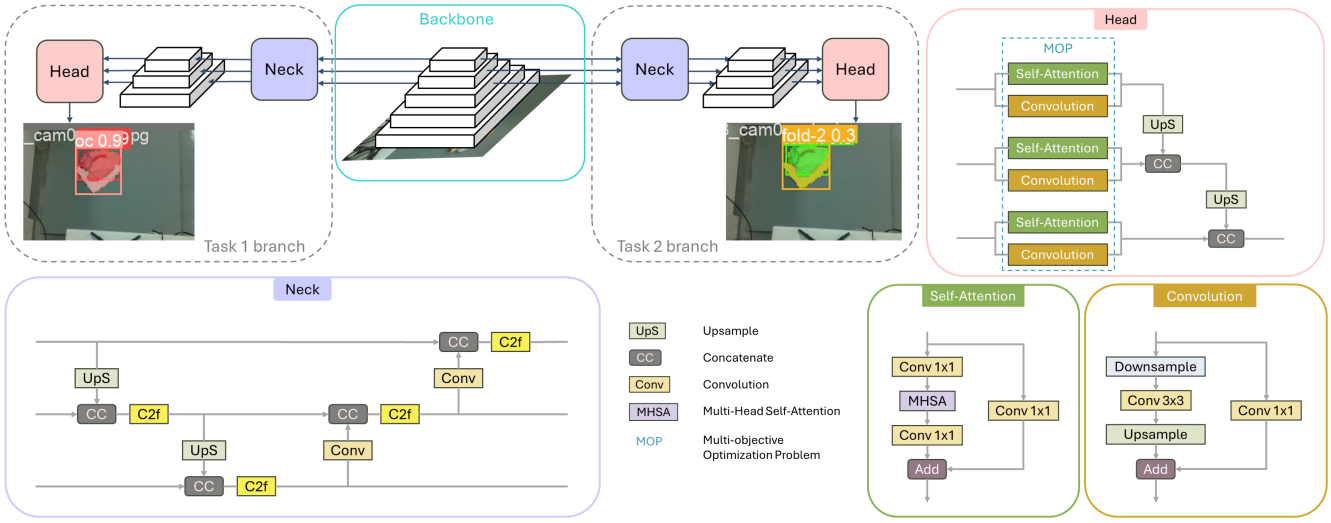


Fig. 1. Structure of the proposed MTL model with two tasks. It shares one feature extraction backbone and utilizes two task-specific necks and heads. Backbone uses YOLOv8's CSPDarknet53; neck utilizes the same top-down and bottom-up structure of YOLOv8; and head adopts combined self-attention and convolution mechanism from HyCTAS.

pretrained models on large object datasets. YOLOv8 uses a custom CSPDarknet53 as the backbone, introducing a Cross-Stage Partial (CSP) connection that improves gradient flow between different layers. Additionally, the Spatial Pyramid Pooling Fast (SPPF) module captures features at various scales, allowing enhanced accuracy.

The neck is responsible for merging feature maps from different backbone layers. Instead of the traditional top-down Feature Pyramid Network (FPN), YOLOv8 uses a Path Aggregation Network (PANet) that includes both a top-down and a bottom-up FPN. This further improves the model's ability to capture multi-scale features effectively.

The C2f module used in both backbone and neck utilizes a bottleneck structure. It enables effective combination of high-level semantic features and low-level spatial details, leading to reduced computational complexity with boosted accuracy especially for small objects.

2) Head

Convolutions allow extraction of region-sensitive features, but have limited receptive fields. Self-attention mechanisms used in transformers that track long-range dependencies and capture better global context, however, are demanding in computational and memory consumptions. HyCTAS thus combined a memory-efficient self-attention module and a light-weight convolution module both with residual structures to achieve high segmentation accuracy with efficiency. The combination, which can be seen as weight optimization, is solved as a multi-objective optimization problem (MOP). In this work, we apply this combined self-attention head to the 1/8, 1/16 and 1/32 scale output layers from the neck.

3) Loss

We employ an end-to-end training for the MTL model with a directly combined multi-task loss:

$$Loss = \sum Loss_{task} \tag{1}$$

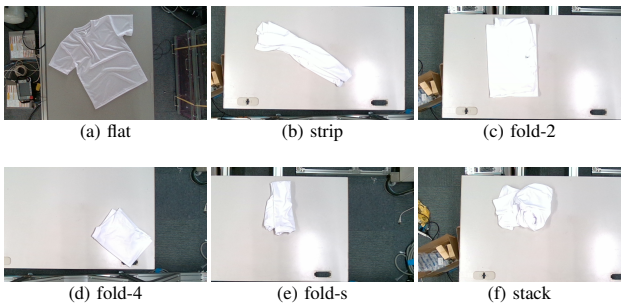
The components of loss functions for different tasks $Loss_{task}$ fully depend on the task type:

$$\begin{aligned} Loss_{det} =& w_{BCE}Loss_{BCE} + w_{VF}Loss_{VF} + \\ & w_{CIoU}Loss_{CIoU} + w_{DFL}Loss_{DFL} \\ Loss_{seg} =& w_{BCE}Loss_{BCE} + w_{VF}Loss_{VF} + \\ & w_{CIoU}Loss_{CIoU} + w_{mIoU}Loss_{mIoU} + \\ & w_{DFL}Loss_{DFL} \\ Loss_{pose} =& w_{BCE}Loss_{BCE} + w_{VF}Loss_{VF} + \\ & w_{CIoU}Loss_{CIoU} + w_{DFL}Loss_{DFL} + \\ & w_{kpt}Loss_{kpt} + w_{kobj}Loss_{kobj} \end{aligned} \tag{2}$$

where CIoU (Complete Intersection over Union) loss evaluates intersections of bounding boxes while mIoU evaluates intersections of masks; DFL (Distributed Focal Loss) is used to deal with class imbalance; BCE (Binary Cross-Entropy) loss and Varifocal loss are both used for classification; $Loss_{kpt}$ evaluates predicted keypoints by Euclidean Distance; and $Loss_{kobj}$ is used to reduce proposed keypoints with low objectness; and ws are the coefficients.

B. T-shirt state recognition

One primary vision objective of the t-shirt handling project is to identify and segment t-shirts in the environment and understand the conditions they are in. T-shirt states is one of those conditions. Based on industrial needs and empirical everyday-life actions, taking into account the most frequent t-shirt states it might assume, we defined six classes as illustrated in Fig.2. (a) **Flat** represents a t-shirt laid out largely flat on a surface, maintaining its recognizable shape, while by chance exhibiting slight crumples or minor folds at the corner or on the sleeves. (b) **Strip** is a t-shirt folded vertically but spread horizontally, assuming a strip-like shape. This can occur either by casually picking up the garment around the



(a) flat (b) strip (c) fold-2

(d) fold-4 (e) fold-s (f) stack

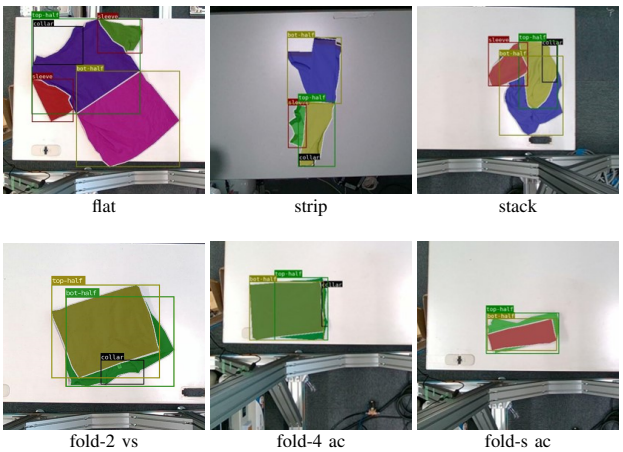
Fig. 2. Examples of the six t-shirt states.

collar or both shoulders and then placing and dragging it on the table, or by carefully folding it from the flat state. (c) **Fold-2** means a t-shirt being folded once horizontally from the flat state, with the positions of the sleeves being random. (d) **Fold-4** is a t-shirt folded into a square-like shape, which can be achieved either by folding one more time vertically from the 2-fold state, or by folding in the style of a "dress shirt fold" or "military fold". (e) **Fold-s** represents a t-shirt folded once more horizontally from the strip state, mimicking the method typically employed when one wishes to fold a t-shirt swiftly and casually. (f) **Stack** occurs when the t-shirt is picked up randomly at one or two points and then casually dropped on the table, or represents any t-shirt that cannot fall into the other state categories.

To address the lack of available public open-source dataset on diverse on-the-table t-shirts, we collected and annotated a dataset of 1250 RGB images, 80% of which was used as trainset. The orientation of the t-shirt in those images, whether it is front-up or back-up, is not predetermined but random. The dataset includes a wide range of t-shirt colors as well as both short-sleeve and long-sleeve t-shirts. Additionally, the t-shirts were placed in various environments, including different tables and lighting conditions such as natural daylight and artificial light. All the images in this dataset have been manually annotated by outlining each t-shirt segment and assigning a state class. Our dataset encompasses several challenging scenarios that are not presented in any other t-shirt or garment datasets. As illustrated in Fig.4, besides the normal setting of single t-shirt, multiple t-shirts with gaps, and multiple different colored t-shirts with no gap, our dataset also includes (1) those scenarios from viewpoints other than top-view; (2) multiple same-colored t-shirts with no gap; and (3) multiple t-shirts mixed up or laid over each other with considerable occlusions. Additionally, in the images not from top-view, operating tables are introduced as the 7th class.

C. T-shirt layer estimation

Another important t-shirt condition to understand is the layers they are in – whether they are on the top or being occluded by other t-shirts. For this task we use the same dataset as in the t-shirt state recognition but change the labels to two classes – top or occluded, as illustrated in the first column of Fig.5. This task, however, can be a simple object



flat strip stack

fold-2 vs fold-4 ac fold-s ac

Fig. 3. Examples of the four t-shirt parts for each state, where ac means annotated by actual top or bottom and vs means visual top or bottom.

detection task without the masks if combined with the t-shirt state recognition task.

D. T-shirt parts detection

Understanding t-shirt parts is an important intermediate stage for a versatile garment handling project. Many works [2], [3], [4] that directly estimate grasping points or propose robot movements either require very large dataset or can only handle specific operations or initial garment states. However, knowing the parts or components of a garment would enable the proposals of grasping points without further deep learning process and can be easily adjusted according to the needs of different operations.

We defined 4 t-shirt parts – collar, sleeve, top-half, and bottom-half – for each one of the 6 t-shirt states respectively, as illustrated in the first column of Fig.5. Specially for states fold-2, fold-4, and fold-s, there are two ways to define top-half and bottom-half: one is to define by the actual top or bottom of a t-shirt – the half that includes the collar is the top-half; the other is to define by the visual top or bottom – whichever half is on the top layer without occlusions is the top-half. For the stack state, top-half is defined as the part that is not occluded and that includes the points a human wants to grasp on by instinct. The remaining parts of the t-shirt would then be classified as the bottom-half.

For the training of this task, we did not include the images that have multiple mixed-up and occluded t-shirts for a better instance balance of the dataset and the fact that recognizing parts from single t-shirt under certain states presents enough challenge. Results and explanations will be detailed in the experiments section.

IV. Experiments

In this section, we introduce the experiments carried out on the single tasks and the MTL model. mAP (mean Average Precision) and inference time are evaluated for model performances. mAP is calculated by the area under precision-recall curve, where mAP_{50} is the mAP at an IoU threshold of 50% and mAP_{50-95} is the average mAP at IoU thresholds every

TABLE I

SINGLE TASKS RESULTS

Model	State task		Layer task	
	mAP_{50} (%)	mAP_{50-95} (%)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv8-seg	**98.0**	91.5	**98.3**	88.2
w/ HyCTAS	97.6	**92.4**	95.2	**89.1**
YOLOv8-det			**98.5**	**90.2**
w/ HyCTAS			97.0	88.0

TABLE II

mAP_{50-95} RESULTS UNDER DIFFERENT SETTINGS

			all	flat	fold-2	fold-4	fold-s	strip	stack
separate	original	ac-top		75.6	36.6	25.9	12.5	32.5	20.7
		vs-top		75.6	37.5	28.0	16.2	32.5	20.7
	cropped	ac-top		**77.4**	39.3	34.8	14.1	38.2	17.7
		vs-top		**77.4**	40.9	32.1	16.6	38.2	17.7
mixed	original	ac-top	36.6	73.1	38.3	47.0	22.6	40.4	22.4
		vs-top	39.9	71.4	**43.9**	48.4	28.9	39.0	**25.5**
	cropped	ac-top	40.5	76.4	37.9	43.5	25.3	41.0	22.2
		vs-top	**44.0**	76.2	41.7	**51.5**	**40.9**	**43.4**	20.3

TABLE III

INFERENCE SPEEDS (FPS) BETWEEN THE SINGLE AND MTL MODELS

Tasks	Single tasks total	MTL model
state-seg + layer-det	86.21	128.39
state-seg + layer-seg	95.24	140.05
fpg1-seg + fpg2-seg	98.04	210.47
state-seg + layer-seg + fpg1-seg	64.10	231.19
state-seg + layer-seg + parts-seg + fpg1-seg	45.87	256.41

5% from 50% to 95%. mAP is a widely recognized metric for detection and segmentation tasks for its comprehensive evaluation ability. Inference time is evaluated by FPS (frames per second). All the experiments are carried out with an RTX 3080 Laptop GPU.

A. T-shirt states and layers

We trained and tested t-shirt states task and t-shirt layer task respectively as single tasks using YOLOv8 with the combined self-attention head of HyCTAS, and compared the results with using original YOLOv8. Each model was trained around 200 epochs until convergence with a stepped learning rate starting from 0.001 and a batch size of 1. Comparison results are as shown in Table I. YOLOv8 with HyCTAS outperforms YOLOv8 slightly on the stricter metric mAP_{50-95} in segmentation tasks but not on mAP_{50}. This demonstrates that YOLOv8 with HyCTAS is more precise with the details and proposes less inaccurate masks. The original YOLOv8 has better results in detection tasks, which we believe is because HyCTAS were optimized specifically for segmentation tasks with finer per-pixel focus. This suggests that YOLOv8 with HyCTAS could be better at complex, fine-grained tasks.

B. T-shirt parts

We employed different strategies for the t-shirt parts recognition task. Annotation-wise the top-layer can be the actual top (referred to as ac-top) or the visual top (vs-top) as in 3; augmentation-wise the images can be cropped and focused on the t-shirts as in 3 or not as in 2; training-wise each state can be either trained separately resulting in 8 different models or mixed up and trained together. We compared the overall and per-state performance, as shown in Table II. The results showed that overall cropped images annotated by visual-top trained with mixed states has better performance in terms of mAP_{50-95}. We analyze that in mixed-states training, the differences between each state in fact improve the model's generalizability and focus more on the common and effective features. Along with the size of the dataset, this could explain the improved performance on the difficult states (fold-2, fold-4, fold-s, strip, stack) and sacrificed flat-state performance.

C. MTL models

We trained and tested various MTL models with different garment tasks. For two-task-MTL models, we tried the combination of instance segmentation of t-shirt states (state-seg) plus instance segmentation of t-shirt layer (layer-seg), and state-seg plus object detection of t-shirt layer (layer-det). We also trained and tested their corresponding versions of MTL model with the original YOLOv8 head structure for comparison. Besides our own dataset, we also utilized Fashionpedia, which is an open garment dataset that presents 46 types of garments worn on humans. We chose 4 classes (t-shirt and sweatshirt, sweater, jacket, and dress) as one group (fpg1) and another 4 (hair accessory, tie, bag and wallet, and umbrella) as a second group (fpg2) with each group containing more than 4000 images, and formed a MTL model with two instance segmentation tasks (fpg1-seg and fpg2-seg). Additionally, we tested this MTL model structure's ability with more than two tasks, specifically, with 3 tasks: state-seg + layer-seg + fpg1-seg, and with 4 tasks: state-seg + layer-seg + fpg1-seg + parts-seg, where parts-seg refers to the t-shirt parts segmentation task with all the six states mixed up.

We evaluated these models by inference speed and performance with comparison to the corresponding single tasks. In each single comparison, the irrelevant (e.g., batch size for inference speed comparison) parameters were set to the same though among different comparisons they might be different. The models' hyperparameters were adjusted for better performances. The results and details will be discussed in the following paragraphs with a focus on inference speed, performance boosted by HyCTAS head, and possible improving methods – loading weights and adding unrelated tasks.

(1) Inference speed: We evaluated each MTL model's inference speed in terms of fps and compared it with the speed of running all their tasks using single task models. The results are shown in Table III, where the numbers are given by an average of multiple test results. It demonstrates that this MTL structure can significantly increase inference speed when handling multiple tasks. And the more tasks there are, the bigger the improvements.

(2) Performance: For the two-task MTL models, we compared the results from our proposed MTL structure (w/ HyCTAS) with the ones from an MTL with the original

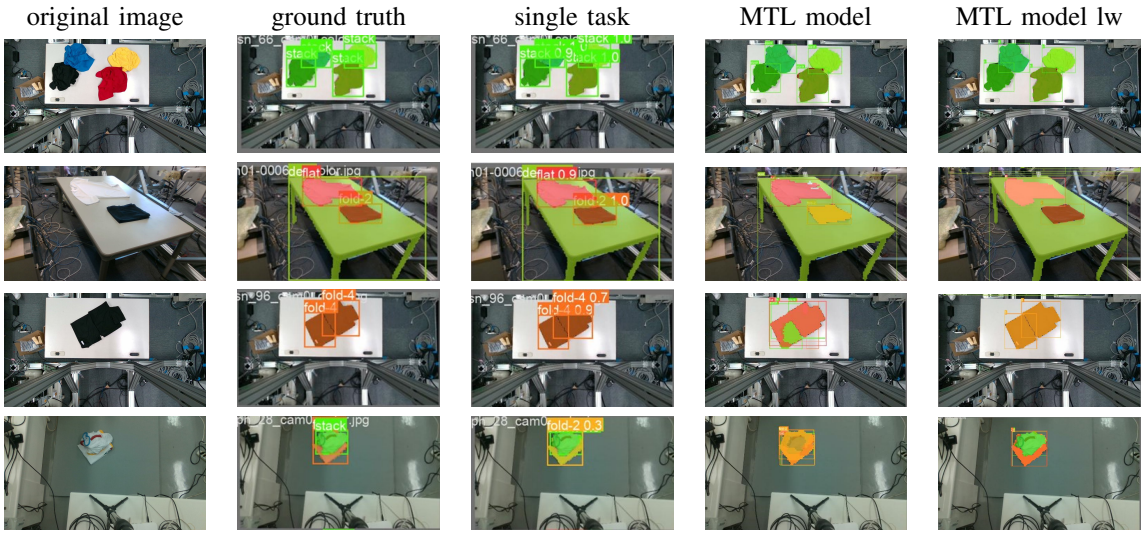


Fig. 4. Examples of the t-shirt state segmentation task (state-seg) and predictions from different models. lw refers to the MTL model directly loading weights from single task models without further training. The MTL model tends to over-detect compared to single task models in the difficult scenes as in the third and fourth rows, which can be mediated by loading weights.

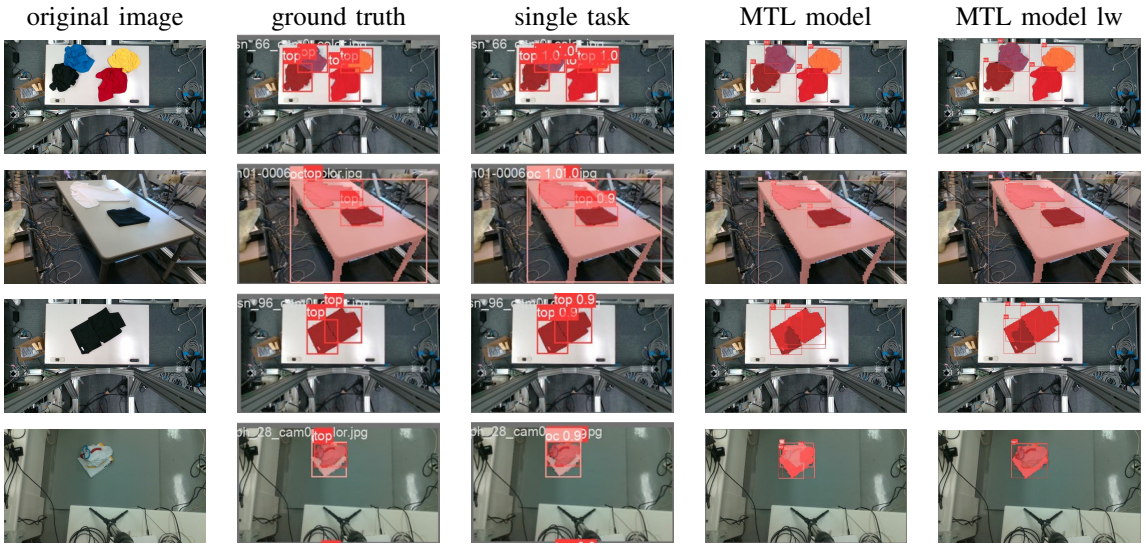


Fig. 5. Examples of the t-shirt layer segmentation task (layer-seg) and predictions from different models. lw refers to the MTL model directly loading weights from single task models without further training. The MTL model tends to over-detect compared to single task models in the difficult scenes as in the third and fourth rows, which can be mediated by loading weights.

YOLOv8 structure, as presented in Table IV, Fig.4 and Fig.5. We also compared these results to the results from corresponding single task models, in terms of differences by single task results subtracting MTL model results (DfST). Although the single task models still mostly outperform our proposed MTL models by a small amount, the added HyCTAS structure significantly boosted the performance of MTL models. This is likely because of the combined self-attention and convolution head structure of HyCTAS and the optimization of their weights, which could better utilize the general features of MTL model's single encoder.

(3) Loading weights: During the experiments we found that loading single task model's weights to the MTL model without further training the MTL model could achieve similar results as single tasks and better ones than the trained MTL models, as shown in Table IV. However, this phenomenon does not apply to all the MTL models as it largely depends on the similarity between the integrated tasks. And the choice of backbone's weights also makes a big difference. The results shown in Table IV are from an MTL model with single state-seg task's backbone weights, which explains why the performance of the layer-seg branch has bigger differences from the single task.

(4) Adding unrelated task: Another promising way to improve the performance of the MTL model is to add an extra task. As shown in Table V, fpg1-seg is unrelated to the target of providing information on t-shirts in the industrial setting, but adding it as one branch of the MTL model

TABLE IV

COMPARISON OF MTL MODELS WITH ORIGINAL YOLOV8 STRUCTURE AND WITH HYCTAS HEAD. DFST REFERS TO THE DIFFERENCE FROM SINGLE TASK'S RESULTS. LW REFERS TO THE MTL MODEL THAT LOADS SINGLE TASKS' WEIGHTS WITHOUT TRAINING. THE TABLE HIGHLIGHTS THE PERFORMANCE ENHANCEMENTS ATTRIBUTED TO THE HYCTAS HEAD, AS OBSERVED IN THE 'W/ HYCTAS' ROWS IN COMPARISON WITH THE 'YOLOV8' ROWS, AND THE POTENTIAL FOR FURTHER IMPROVEMENT THROUGH LOADING WEIGHTS, AS DEMONSTRATED IN THE 'LW' ROWS.

	state segmentation + layer detection								state segmentation + layer segmentation							
	state segmentation mask				layer detection bbox				state segmentation mask				layer segmentation mask			
	mAP_{50}	DfST	mAP_{50-95}	DfST	mAP_{50}	DfST	mAP_{50-95}	DfST	mAP_{50}	DfST	mAP_{50-95}	DfST	mAP_{50}	DfST	mAP_{50-95}	DfST
YOLOv8	90.3	7.7	77.8	13.7	94.6	3.9	78.0	12.2	89.6	8.4	79.2	12.3	88.3	10.0	72.9	15.3
w/ HyCTAS	**94.2**	**3.4**	**86.8**	**5.6**	**98.1**	**-1.1**	**85.0**	**3.0**	**95.9**	**1.7**	**89.7**	**2.7**	**97.1**	**-1.9**	**83.7**	**5.4**
YOLOv8 lw									96.6	1.4	89.7	1.8	94.2	4.1	**84.8**	**3.4**
w/ HyCTAS lw									**97.6**	**0**	**92.5**	**-0.1**	**91.3**	**3.9**	84.2	4.9

TABLE V

PERFORMANCE BOOST FOR MTL MODEL WITH AN UNRELATED TASK FPG1-SEG. WITH FPG1-SEG AS A BRANCH OF THE MTL MODEL, STATE-SEG AND LAYER-SEG ACHIEVE RESULTS THAT ARE MUCH BETTER THAN THE MTL MODEL OF ONLY THOSE TWO AND THAT ARE MORE SIMILAR TO SINGLE TASK MODELS.

	state segmentation		layer segmentation		fpg1 segmentation	
Models	mAP_{50}	mAP_{50-95}	mAP_{50}	mAP_{50-95}	mAP_{50}	mAP_{50-95}
Single task	97.1	91.0	96.6	88.7	80.8	64.0
MTL: state-seg + layer-seg	89.6	79.2	88.3	72.9		
MTL: fpg1-seg + fpg2-seg					78.2	63.3
MTL: state-seg + layer-seg + fpg1-seg	97.4	89.1	93.8	83.7	65.5	52.6

increases the performance of the desired task branches state-seg and layer-seg. However, this only happens when the extra task uses a dataset big enough and is with certain difficulty. It suggests that even through hard parameter sharing, the MTL model is able to leverage the general or task-specific information in the extra task and shift its preference to more commonly useful features and thus possibly improve model's generality [21].

V. CONCLUSIONS AND FUTURE WORK

In this work, we proposed an MTL model based on YOLOv8 with HyCTAS combined self-attention head for complex garment recognition tasks. It is designed to fulfill the requirements of providing multi-dimensional information on garments during industrial process. Currently we focused on segmenting t-shirts and identify their states and layers, as well as recognizing t-shirt parts. Using our own dataset and transfer learning technique, experiments were carried out on this MTL model in comparison with the single models and the MTL model without the HyCTAS head. Results demonstrated that our MTL model was able to run with a significantly higher speed than executing multiple single models. With the HyCTAS head the performance difference from the single task models were largely reduced. And by loading single task models' weights or adding an extra auxiliary task, this difference could be further diminished and our MTL model could achieve a similar result as the independent single tasks.

Nevertheless, there are still opportunities for enhancements. Firstly, the performance of each single task can be improved, especially the segmentation of t-shirts from a complex same-color mix-up pile and the t-shirt parts recognition of the folded states. These can be exploited by upgrading the dataset, leveraging better data augmentations and investigating different backbone, neck or heads as the current MTL model allows independent changes to those modules. Secondly, the pattern of efficient auxiliary task can be further explored. It would be beneficial to discover a common but more detailed rule for selecting an efficient auxiliary task. Achieving this would require a large number of experiments.

The next phase of our work involves proposing task-specific grasping points for various robot tasks based on the information of state and layer of each t-shirt segment and recognized t-shirt parts. It would be also important to expand this method to other garment types and include more aspects of information to provide. By setting these goals, we aim to not only advance the garment processing industry but also benefit the households and laundries.

REFERENCES

[1] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[2] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1–8.

[3] Y. Deng and D. Hsu, "Generalizable clothes manipulation with large language model," 2024 ICRA Workshop on Representing and Manipulating Deformable Objects.

[4] R. Wu, H. Lu, Y. Wang, Y. Wang, and H. Dong, "Unigarmentmanip: A unified framework for category-level garment manipulation via dense visual correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 340–16 350.

[5] W. Wang, G. Li, M. Zamora, and S. Coros, "Trtm: Template-based reconstruction and target-oriented manipulation of crumpled cloths," *arXiv preprint arXiv:2308.04670*, 2023.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[7] R. Varghese and M. Sambath, "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. IEEE, 2024, pp. 1–6.

[8] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The emerging trends of multi-label learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 11, pp. 7955–7974, 2021.

[9] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer, 2014, pp. 94–108.

[10] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4822–4829.

[11] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, "Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3205–3214.

[12] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE intelligent vehicles symposium (IV)*. IEEE, 2018, pp. 1013–1020.

[13] D. Wu, M.-W. Liao, W.-T. Zhang, X.-G. Wang, X. Bai, W.-Q. Cheng, and W.-Y. Liu, "Yolop: You only look once for panoptic driving perception," *Machine Intelligence Research*, vol. 19, no. 6, pp. 550–562, 2022.

[14] J. Wang, Q. J. Wu, and N. Zhang, "You only look at once for real-time and generic multi-task," *IEEE Transactions on Vehicular Technology*, 2024.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[16] H. Yu, C. Wan, M. Liu, D. Chen, B. Xiao, and X. Dai, "Real-time image segmentation via hybrid convolutional-transformer architecture search," *arXiv preprint arXiv:2403.10413*, 2024.

[17] M. Jia, M. Shi, M. Sirotenko, Y. Cui, C. Cardie, B. Hariharan, H. Adam, and S. Belongie, "Fashionpedia: Ontology, segmentation, and an attribute localization dataset," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 316–332.

[18] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1096–1104.

[19] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5337–5345.

[20] H. Lu, Y. Li, R. Wu, C. Ning, Y. Shen, and H. Dong, "Unigarment: A unified simulation and benchmark for garment manipulation," in *ICRA Workshop on Deformable Object Manipulation*, 2024.

[21] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997.