CuttingEdgeAI

RL for Unique 2D Cutting Stock Problems

Greg Bateham
Department of Computer Science
Florida Polytechnic University
Lakeland, Florida, USA

gbateham9459@floridapoly.edu

Sunim Acharya
Department of Computer Science
Florida Polytechnic University
Lakeland, Florida, USA

sacharya9762@floridapoly.edu

Satyam Thakur
Department of Computer Science
Florida Polytechnic University
Lakeland, Florida, USA

sthakur0112@floridapoly.edu

Joyce Champie
Department of Computer Science
Florida Polytechnic University
Lakeland, Florida, USA

cjoycemaptuetagne00@floridapoly.edu

Anastasiya Dmytryk
Department of Computer Science
Florida Polytechnic University
Lakeland, Florida, USA

admytryk9341@floridapoly.edu

Dr. Bayazit Karaman
Department of Computer Science
Florida Polytechnic University
Lakeland, Florida, USA

bkaraman@floridapoly.edu

Abstract— **Fashion manufacturing, in the case of small businesses and solo seamstresses, typically remains muddled in labor-consuming, time-wasting techniques that suppress creativity and encourage material waste. This project considers the cutting stock problem in pattern layout optimization of items within irregular boundaries—a problem whose solution, if achieved, would significantly reduce garment manufacturing to a simple task. Our motivation stems from the desire to automate a previously analog field, hence reducing the technical burden on designers and seamstresses and making room for more creative design efforts. To address this issue, we propose an integrative machine learning solution that automatically determines the optimal combination of unique pattern types on garments of various geometries. Our approach applies a broad database of SVG and PNG images, each labeled with specific parameters, to train models that learn and make predictions for ideal layout compositions. We utilize cutting-edge techniques such as Long Short-Term Memory networks for pattern recognition, ResNet for dimension mapping, and tree search and hierarchical reinforcement learning algorithms for dimension extraction and cutting optimization. These are quantified in terms of computation time, waste percentage, and material usage. In the end, our research will not only contribute an innovative machine learning methodology to a lackluster researched domain of fashion manufacturing but can also potentially revolutionize how pattern layouts are developed, increasing sustainability and efficiency industry-wide.**

I. Introduction

Fabric waste is a persistent drag on efficiency in apparel manufacturing. Industry surveys have estimated the waste per roll of fabric at anywhere from 10 percent to 25 percent, which adds to both material expense and environmental footprint. Although there has been vast research on the two-dimensional cutting-stock problem (2DCSP) for several decades, the simplifying assumptions of traditional models are seldom fulfilled in actual manufacturing contexts. There are spots or tears on the irregular textile remnants, pattern pieces of irregular shapes, and directionality constraints like grain direction or print alignment that eliminate most rotations. These factors narrow down an already NP-hard combinatorial problem to a search space where classical heuristics always get stuck at obviously sub-optimal arrangements by visual inspection. The majority of shop owners circumvent this by purchasing additional fabric or by spending valuable time manually placing pattern pieces, neither of which is feasible for their bottom line or the environment. This paper describes a start-to-finish system that takes an image of any fabric scrap along with its pattern pieces and returns a cutting plan that maximizes usable material while accommodating defects, grain direction, and print repeats.

The proposed workflow integrates two main innovations. First, a lightweight computer-vision module detects defects and infers directional constraints, converting a raw image into a precise placement mask. Second, a benchmarking suite evaluates greedy heuristics, evolutionary search, and recent reinforcement-learning algorithms to identify the approach that delivers the highest material utilization under realistic constraints.

The proposed workflow integrates two main innovations. First, a lightweight computer-vision module infers directional constraints and detects defects, converting a raw image to a precise placement mask. Second, we employ a benchmarking suite to compare greedy heuristics, evolutionary search, and state-of-the-art reinforcement-learning algorithms for determining the most promising approach to realizing the optimum material utilization within practical limits. The system, by virtue of the integration of automatic defect detection and data-driven layout optimization, aims for significant savings in fabric waste, man-hours, and production costs, thus bringing sophisticated optimization technology within the reach of small garment producers.

II. Literature Review

The challenge of optimizing two-dimensional cutting stock problems and garment pattern recognition has been approached using a variety of machine learning techniques.

A. *Shape recognition and corner detection in 2D drawings*

Shape recognition and corner detection in 2D drawings have traditionally relied on rule-based approaches and handcrafted feature extraction. The first paper introduces an LSTM-based method that improves shape recognition by learning temporal dependencies in stroke sequences, making the approach more robust against variations in drawing styles.

B. Packing irregular objects efficiently

Packing irregular objects efficiently is another key optimization problem in manufacturing and logistics. Traditional heuristic-based methods struggle with handling complex shapes and constraints. The second paper proposes hierarchical reinforcement learning (HRL), which breaks down the problem into high-level decision-making and low-level execution, making it more efficient.

C. Deep learning for garment classification and pattern recognition

Deep learning for garment classification and pattern recognition has shown promise in automating fashion-related tasks. Convolutional neural networks (CNNs) and generative adversarial networks (GANs) have been utilized to improve classification accuracy. The third paper highlights the effectiveness of these models in extracting deep features from textile images and garment designs.

D. Tree search reinforcement learning (TSRL) for cutting stock problems

Tree search reinforcement learning (TSRL) for cutting stock problems is a more structured approach to optimizing cutting patterns. Traditional mathematical optimization techniques struggle with complex industrial constraints. The fourth paper introduces a reinforcement learning model that integrates tree search, ensuring feasible and efficient cutting solutions.

E. Optimized 2D ball trees for shape layout applications

Optimized 2D ball trees for shape layout applications are an optimization for shape recognition and distance recognition problems. While the study doesn't focus on fashion garments, the process of selecting a shape with ball trees can be reused for any other 2D pattern recognition problem. This paper goes into the details of how to minimize the number of balls needed, thus reducing the computational complexity of shape recognition. One of the significant developments in deep learning techniques is the NeuralTailor architecture [4], which proposes a novel architecture specifically designed for analysis and classification of garments. These papers collectively contribute to the broader field of machine learning for optimization, pattern recognition, and decision-making, forming the foundation for our project.

F. Research Gaps and Justification

Existing methods struggle with accurately detecting intersection points and handling variations in image quality. The proposed LSTM-based approach aims to fill this gap by introducing angle and curvature features to enhance model performance. Hyperparameter tuning further optimizes the model, making it more robust against noise and varying image sizes. However, comparisons with CNN-based models remain unexplored. One major limitation of existing methods is their inability to capture fine-grained geometric details of irregular objects, leading to inefficient space utilization. Prior approaches primarily rely on volumetric representations or simplified bounding box approximations, which fail to fully exploit an object's true shape. To address this, multi-view heightmaps and a hierarchical reinforcement learning framework have been introduced to improve geometric understanding and placement decisions. However, these methods still struggle with computational bottlenecks and sequence optimization inefficiencies.

The integration of multiple data modalities remains a significant gap. While deep learning techniques have improved feature extraction, current approaches largely focus on geometric features while underutilizing measurement and pattern data. Additionally, models Models trained on synthetic data exhibit performance degradation when applied to noisy real-world inputs, reducing the model's effectiveness in real-world classification tasks. Reinforcement learning-based cutting stock optimization has made strides in constraint handling, yet current research still relies on linear function approximations for value functions. The integration of deep neural networks could enhance solution accuracy and generalization across different domains. Additionally, the experimental paradigm relies on well-crafted industrial data, but robustness to noise and uncertainty in real-world problems remains untested. Future research should explore adaptability to real-world variability, improvements in search efficiency, and robustness against uncertainties to make these models more scalable and practical.

Ball tree pattern recognition is not used in the fashion industry. Currently, it is not the most popular technique in shape recognition either. However, balls trees are extremely flexible data structures; with them, it is easier to recognize any curved shape; they can differ in diameter to fill any space [8]. The biggest downside of using ball trees is that the research done with them is very faint, with small data sets used. Trying to recreate the experiment from the paper on a different data set may lead to unknown results, as it is not promised to be a universal solution. Collision detection is also one of the possible faults that needs to be considered. While the algorithm looks out for collisions, it is not always accurate. For our experiment, collision detection needs to be flipped, where if collision is detected, then the shape is within the fabric. Overall, while this method isn't proven to be the best, it can aid in recognizing the shape of the textile itself without training the model on specific textiles, reducing the computational complexity, as fabric will be slightly different for every test.

III. Methodology

This section details the methodology employed in our "Cutting Edge" system, which focuses on the analysis of garment patterns and cloth materials for optimal pattern placement. The methodology integrates computer vision (CV) techniques, deep learning models, and reinforcement learning (RL) to achieve pattern recognition, cloth analysis, and efficient material utilization. All implementation was performed using the Python programming language, leveraging libraries such as PyTorch, OpenCV, Stable Baselines3, Gymnasium, and NumPy.

A. Data Acquisition and Preprocessing

The system utilizes the GarmentCodeData dataset. The DatasetLoader class handles data loading.

Image preprocessing prepares data for the deep learning models. A standardized pipeline, implemented in the preprocess_image_for_model utility function, includes resizing, color space conversion, and tensor transformation. Crucially, it applies normalization using ImageNet statistics. For an input image tensor X' each channel X_{ch} is normalized as:

$$X'_{ch} = \frac{X_{ch} - \mu_{ch}}{\sigma_{ch}}$$

where μ_{ch} and σ_{ch} are the pre-computed mean and standard deviation for that channel from the ImageNet dataset.

The PatternDataset class manages data loading for training, applying these transformations and data augmentation.

B. Pattern Recognition

The PatternRecognitionModule analyzes pattern images. Its core is a PatternCNN model, using a ResNet50 backbone for feature extraction, followed by a final fully connected layer for classification.

Key functionalities include:

1. **Contour Extraction:** Standard OpenCV algorithms are used via the extract_contours utility to find pattern outlines.
2. **Dimension Estimation:** A dimension_predictor network predicts the pattern width (w) and height (h) from extracted features f . This involves minimizing a loss function, typically Mean Squared Error (MSE), between the predicted dimensions $\dot{d} = (\dot{w}, \dot{h})$and the ground truth dimensions $d = (w, h)$:

$$L_{dim} = \frac{1}{N}\sum_{i=1}^{N} \left|\left|\dot{d}_i - d_i\right|\right|^2$$

where N is the batch size.

3. **Classification:** The CNN classifies the pattern type. Training minimizes the Cross-Entropy (CE) loss between the predicted probability distribution $\dot{y}$ over C classes and the one-hot encoded true label y :

$$L_{class} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \log(\dot{y}_{ij})$$

4. **Training:** The module's train method optimizes the model parameters by minimizing a combined loss, typically $L_{total} = L_{class} + \lambda L_{dim}$, using backpropagation and an optimizer like Adam.

C. Cloth Analysis

The ClothRecognitionModule analyzes cloth images.

1. **Segmentation and Feature Extraction:** It uses a U-Net model for semantic segmentation and an EfficientNet for feature extraction. U-Net training typically involves minimizing a pixel-wise loss function, such as CE loss or Dice loss, over the segmentation mask.
2. **Contour and Mask Generation:** OpenCV functions extract the primary cloth contour, generating a binary mask representing the usable area.
3. **Dimension Mapping:** A dim_mapper network predicts overall dimensions, likely trained similarly to the pattern dimension predictor using an MSE loss.

D. Pattern Fitting using Reinforcement Learning

Optimal placement uses RL via a PatternFittingModule. The problem is modeled as a Markov Decision Process (MDP), defined by a tuple (S, A, P, R, γ), where S is the state space (cloth layout, current pattern), A is the action space (position (x, y), rotation θ), P is the state transition probability (deterministic in this placement task), R is the reward function, and γ is the discount factor.

1. **Environment:** The PackingEnvironment implements the MDP logic. The state $s \in S$ includes the cloth state grid and current pattern information. An action $a = (x, y, \theta) \in A$ transitions the environment to state s'.
2. **RL Agent:** A PPO agent learns a policy $\pi(a \parallel s)$ that maximizes the expected discounted future reward. PPO optimizes a clipped surrogate objective function to ensure stable policy updates.
3. **Reward Function:** The reward $R(s, a, s')$ is crucial. In this system, it is designed to encourage valid placements and high material utilization U:

$$R(s, a, s') = w_1 \cdot \mathrm{II}(success) + w_2 \cdot \Delta U + w_3 \cdot efficiency - w_4 \cdot \mathrm{II}(failure)$$

where $\mathrm{II}(\cdot)$ is the indicator function, ΔU is the change in utilization, efficiency might reward adjacent placements, and w_i are weighting factors. The utilization U is calculated as:

$$U = \frac{Area_{patterns}}{Area_{cloth}}$$

where areas are measured within the valid cloth mask.

4. **Training and Inference:** The train method optimizes the policy π using the PPO algorithm. The fit_patterns method uses the learned policy π to select actions greedily or stochastically during Inference to place patterns.

E. System Integration and Execution

The main.py script integrates these modules. It parses arguments, initializes modules, manages data flow, calls the respective processing functions (process_pattern, process_cloth, fit_patterns), and handles visualization using Matplotlib. Configuration parameters from config.py are used throughout.

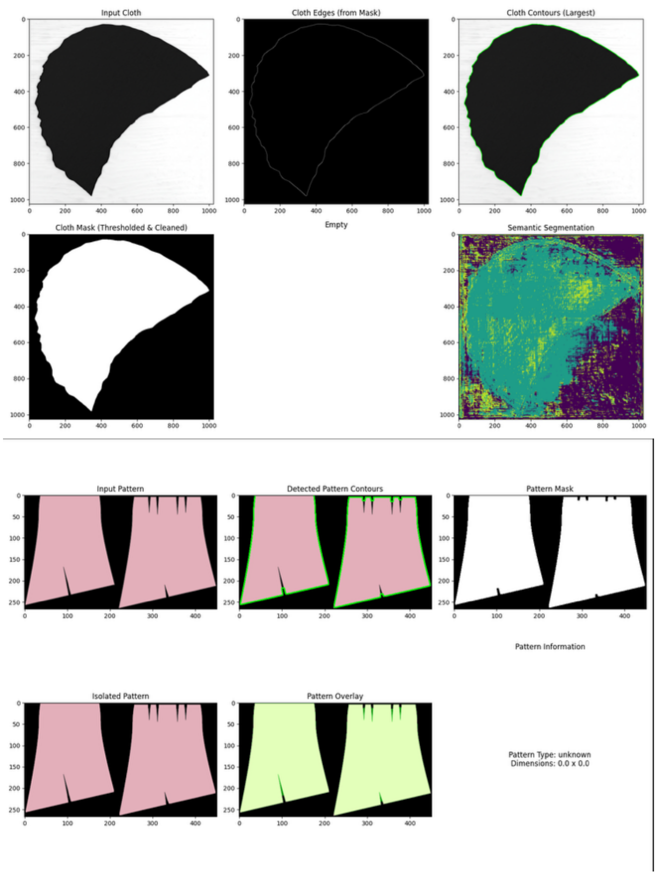
IV. Implementation and Results

A. Implementation overview

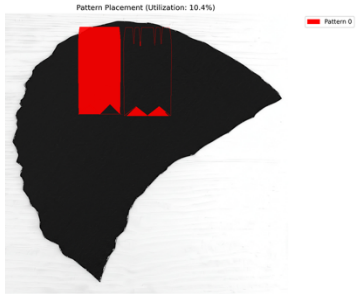
The multi-stag pipeline system is implemented on a standard MacOS computing platform and on Linux with a CUDA optimized GPU. The core system is built using Python 3.x with the OpenCV, Py-torch and U-net libraries. OpenCV is used for image processing, contour detection and morphological operations. For image processing the torchvision and PyTorch were applied specifically ResNet and EfficientNet. There are three pipeline stages: Cloth Parameter extraction, Pattern Parameter extraction, Optimized Pattern fitting. After the data is fed through the pipeline the results are analyzed to see if any parameter tunning is required.

B. Results Overview

The system was evaluated on both MacOS and Linux environments. Figure 1 shows the main stages of pattern placement such as input, cloth detection, pattern contour and the optimized placement.



From figure 2 it can be observed that the system achieved a 10.4% pattern placement rate, that percentage indicates how much material was optimized for an efficient pattern placement on an irregular shaped cloth.



It was observed that the Linux system with CUDA had a higher utilization rate compared to the one on macOS with the ration of 10.38% to 1.01% as follows. It is a great indicator that with a GPU acceleration the system can achieve better results. Both systems were tested with the cloth dimensions of 977x982 pixels, however macOS system was placing three patterns using advanced Manual placement methods with manual fallback for edge cases and Linux system implemented a single pattern placement.

The results established give a promising baseline for an emerging field of Fashion Technology. Traditional optimization problems deal with rectangular shapes and have a lower success rate when implemented for irregular shapes or require a lot of human intervention to achieve significant results. The ability to achieve almost 57% material utilization of an irregular shape can help save vast manufacturing costs. Moreover, the ability to adapt the algorithm for different frameworks while maintaining consistent pattern recognition indicates that with mor training and refinement even higher optimization rates are achievable. This research represents an important step towards saving textiles from laying in the land fields and providing companies with more usage out of their materials.

V. Discussion And Conclusion

We successfully developed an automated system integrating Computer Vision and Reinforcement Learning to optimize garment pattern layout directly from cloth and pattern images, addressing a critical need for waste reduction in fashion manufacturing. Our methodology effectively employs robust computer vision techniques like thresholding, morphological operations, and contour analysis for cloth boundary detection, coupled with Deep Learning (CNNs) for pattern feature extraction and dimension estimation. Intelligent placement decisions are driven by a Reinforcement Learning agent (PPO via Stable Baselines3), with placements validated using precise geometric checks (Shapely). The system's strengths lie in its use of modern ML/RL techniques, its modular design allowing for future improvements, and the inclusion of heuristic fallbacks for increased robustness.

Despite its success, the system's performance can be sensitive to image quality and contrast, and handling complex fabric properties like defects or directionality requires further enhancement. Optimizing the RL agent may demand significant computational resources and tuning, while generalization across highly varied inputs remains an ongoing challenge. Nevertheless, this work demonstrates a viable AI-driven approach to significantly reduce material waste, offering potential cost savings and improved sustainability for the industry. It establishes a strong foundation for future advancements in automated textile processing systems.

VI. Future Directions

Future directions aim at three aspects: (1) include a fabric-aware vision module that can detect defects and nap, grain, and pattern repeats so that erroneous regions and orientations are already eliminated early; (2) replace PPO with more generalizable algorithms like Soft Actor Critic or cooperative multi-agent control and combine them with a reward that encourages utilization, cutting-path simplicity, and material cost; and (3) increase robustness with synthetic-real mixed training and an interactive interface where operators can edit layouts and feed their corrections to continuous learning. These steps altogether will transform the prototype into a responsive, cost-effective, production-ready cloth-planning assistant.

References

[1] S. C. Turan, "A Machine Learning Approach to Detecting Corner Points in Sketches," Eur. J. Artif. Intell., vol. 2, no. 1, pp. 1-12, 2024. [Online]. Available: https://www.ej-ai.ejece.org/index.php/ejai/article/view/34.

[2] S. Huang, Z. Wang, J. Zhou, and J. Lu, “Planning Irregular Object Packing via Hierarchical Reinforcement Learning,” IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 81–88, Jan. 2023, doi: 10.1109/LRA.2022.3222996.

[3] M. Korosteleva, F. Kemper, S. Wenninger, J. Koller, Y. Zhang, M. Botsch, and O. Sorkine-Hornung, “GarmentCodeData (v2): A dataset of 3D made-to-measure garments with sewing patterns,” ETH Library, 2024.

[4] M. Korosteleva and S. H. Lee, “NeuralTailor: Reconstructing sewing pattern structures from 3D point clouds of garments,” ACM Transactions on Graphics, vol. 41, no. 4, article 1, 2022.

[5] M. Korosteleva and O. Sorkine-Hornung, “GarmentCode: Programming parametric sewing patterns,” ACM Transactions on Graphics, vol. 42, no. 6, article 197, 2023.

[6] F. Shi, Y. Meng, and L. Tang, “Tree Search Reinforcement Learning for Two-Dimensional Cutting Stock Problem With Complex Constraints,” IEEE Transactions on Automation Science and Engineering, 2024.

[7] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, “Neural combinatorial optimization with reinforcement learning,” in Proceedings of the 5th International Conference on Learning Representations (ICLR), 2017.

[8] Retondaro, Luis C.S.C., and Claudio Esperança. “Optimized 2D Ball Trees for Shape Layout Applications.” Computers & Graphics, vol. 103, 14 Feb. 2022, pp. 129–139, www.sciencedirect.com/science/article/pii/S0097849322000231, https://doi.org/10.1016/j.cag.2022.02.006.