# TOWARDS A NEVER-ENDING LEARNING MODEL OF RECYCLABLE MEDICAL WASTE

**FENG Jiaqi**
**20231947**
**jiaqifeng077@gmail.com**

**SHAO Qichen**
**20232764**
**shaoqichen2000@gmail.com**

## ABSTRACT

Medical Waste classification is a critical step in achieving efficient processing and resource utilization. However, traditional techniques face significant limitations in classification efficiency, generalization capability, and adaptability to new waste categories. Moreover, even if certain categories of medical waste are predefined a priori, they may appear in different forms in practical applications, further increasing the complexity for classification models. To address these challenges, this study proposes an intelligent medical waste classification model based on an Image Analysis Technology and Continual Learning framework, using the *Medical Waste 4.0* dataset as the experimental foundation. To systematically explore the impact of different training strategies on model performance, five progressively designed experiments were conducted : (1) full-batch training, (2) multi-task learning, (3) independent task training, (4) continual learning, and (5) hierarchical task clustering for continual learning. These experiments aim to gradually uncover the model's potential and challenges in classification performance, convergence efficiency, and task adaptability.

The experimental results show that full-batch training provides a reliable baseline for optimizing the initial performance of the model; multi-task learning significantly improves the model's learning efficiency and shared feature extraction capability; independent task training aims to explore whether there is any interference between tasks in continual learning and to determine the appropriate number of training epochs for each task; continual learning excels in enhancing adaptability to new categories. In addition, the study further explores the impact of task embeddings and task transfer optimization on continual learning performance by incorporating task2vec and LEEP scores, providing a new perspective and practical approach for efficient continual learning in complex task scenarios.

Through the progressive exploration of these five experiments, this study comprehensively investigates the design and optimization strategies for intelligent medical waste classification models and provides theoretical and practical support for achieving the goal of **Never-Ending Learning**. The significance of this research lies in advancing the technological development of intelligent medical waste management, reducing treatment costs, and minimizing the threats posed by medical waste to the environment and human health.

**Key Words :** Medical Waste, Multi Task Learning, Continual Learning

**Encadré par: Monsieur Massinissa Hamidi**

CONTENTS

# 1 INTRODUCTION

Medical waste management is a major global challenge faced by the healthcare industry. According to statistics from the World Health Organization (WHO) in 2024 (Organization (2024)), approximately 15% of medical waste is infectious, toxic, or radioactive, presenting significant complexity, particularly during the classification phase. Although medical waste classification standards vary across countries, textile materials (such as gauze and bandages) and plastic products (such as infusion bags and syringes) typically account for the majority. In addition, sharp objects (such as needles and scalpels) make up about 12% of medical waste and are a primary source of injury and infection for healthcare workers and waste handlers. If improperly managed, these hazardous wastes not only pose a direct threat to human health but also contaminate soil, water, and air, causing severe ecological damage and exacerbating the spread of diseases.

Current medical waste management systems (Zhou et al. (2022)) show significant shortcomings in the classification phase. As the foundational step in waste management and disposal, inefficient and inaccurate waste classification often leads to increased complexity in subsequent disposal processes and may even result in secondary pollution or infection risks. Furthermore, with the continuous advancement of medical technologies, new types of medical waste are emerging, making traditional static classification methods increasingly inadequate to adapt to these dynamic changes. These issues hinder the standardization and scientific improvement of medical waste management and limit the industry's progression toward greater efficiency and sustainability. Effectively addressing the challenges of medical waste classification could significantly improve overall management efficiency and have far-reaching implications for environmental protection and public health.

To address these challenges, this study proposes a medical waste classification model that integrates Image Analysis Techniques with Continual Learning methods, featuring a "Never-Ending Learning" capability. The model is designed to continuously accumulate knowledge, dynamically update features, and adapt to new categories while maintaining classification performance for existing categories, meeting the long-term demands of complex medical waste classification scenarios. Through image analysis techniques, the system achieves automated waste classification, significantly reducing manual intervention and improving classification efficiency and accuracy. Meanwhile, the continual learning approach enables the system to dynamically adapt to new categories, alleviating the problem of "catastrophic forgetting" and ensuring stable classification performance.

To validate the performance of the model and its feasibility in achieving never-ending learning, this study designed five progressive experiments : Full-Batch Training, Independent Task Training, Multi-Task Learning, Continual Learning, and Clustering-Based Task Classification for Continual Learning. These experiments progressively explore the potential and challenges of different training strategies in terms of classification efficiency, feature extraction optimization, and task adaptability.

In the fifth experiment, task embeddings were represented using Task2vec, and tasks were reclassified through hierarchical clustering to form more representative and structured task groups. Subsequently, the LEEP score was utilized to determine the optimal task transfer sequence, and the newly generated tasks were sequentially introduced into the model for continual learning. This experiment further investigates the impact of task embeddings and transfer optimization on continual learning performance, providing a new perspective and practical approach for efficient continual learning in complex task scenarios.

## 1.1 THE CHANGING LANDSCAPE OF MEDICAL WASTE

### 1.1.1 SCENARIO 1 : CATEGORY VARIATION

The types and characteristics of medical waste have undergone significant transformations alongside advancements in medical technology and material science. This evolution not only reflects progress in healthcare but also highlights the growing complexity of waste management and the pressing need for effective classification systems.

*Early Stage : Basic Protective Items*

In the early stages of medical practice, medical waste primarily consisted of simple protective items, such as cloth-based gauze, bandages, fabric masks, and natural rubber gloves. These items were

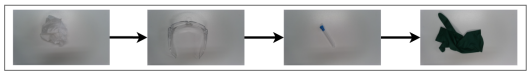


Figure 1: Expansion of item categories

designed to meet basic protective needs at minimal cost. However, their limited durability and susceptibility to contamination posed challenges for infection control and waste management.

*Transition Phase : Specialized Materials*

With the advancement of surgical and diagnostic technologies, medical waste began to include more specialized materials. Synthetic rubber gloves, disposable plastic covers, and impermeable protective suits emerged as essential items. These materials offered improved durability and resistance to chemical exposure, but their increased complexity introduced challenges in classification and disposal.

*Modern Stage : Complex Diagnostic and Therapeutic Tools*

In the modern era, the scope of medical waste has significantly expanded to include diagnostic and therapeutic tools such as syringes, needles, glass test tubes, catheters, and single-use plastic containers. The design and functionality of these items have become more specialized, featuring sterile packaging, antimicrobial coatings, and advanced materials. These innovations have enhanced infection control but simultaneously increased the diversity and volume of medical waste.

*Emerging Challenges : Dynamic and Evolving Waste*

As medical technologies continue to advance, new types of waste, such as materials from high-tech diagnostic devices, radioactive isotopes, and biodegradable implants, have emerged. Traditional static classification methods are increasingly inadequate to address these dynamic changes, complicating waste management processes and amplifying environmental risks.

The evolving landscape of medical waste underscores the necessity for adaptive and efficient classification systems. This evolution demands solutions capable of addressing the challenges of diversity, complexity, and sustainability, laying the foundation for a more effective and environmentally responsible approach to medical waste management.

### 1.1.2 Scenario 2 : Form variation

As shown in the figure 2, the shape of the gauze continuously changes, transitioning from a flat state to becoming increasingly crumpled and complex. This highlights the model's ability to adapt to different newly shaped items.

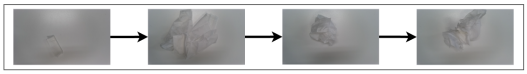


Figure 2: Expansion of item forms

## 2 Continual Learning Setup

In continual learning (Chaudhry et al. (2020)), a learner encounters a stream of data triplets $(x_i, y_i, t_i)$, where $x_i$ is an input, $y_i$ is the corresponding target, and $t_i \in \mathcal{T} = \{1, \ldots, T\}$ is the task identifier. For each task $t_i$, the input-target pair $(x_i, y_i) \in \mathcal{X}_{t_i} \times \mathcal{Y}_{t_i}$ is drawn independently and identically from an unknown distribution $P_{t_i}(\mathcal{X}, \mathcal{Y})$. This distribution characterizes the $t_i$-th learning task.

We assume that tasks are performed sequentially, meaning $t_i \leq t_j$ for all $i \leq j$, and the total number of tasks $T$ is not predetermined.

Under this setup, our goal is to estimate a predictor $f_\theta = (w \circ \phi) : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$, parameterized by $\theta \in \mathbb{R}^D$, where $D$ denotes the dimensionality of the model's parameter space. The predictor is

composed of a feature extractor $\phi : \mathcal{X} \to \mathcal{H}$ and a classifier $w : \mathcal{H} \to \mathcal{Y}$, aiming to minimize the multi-task error.

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{(x,y)\sim P_t}\left[\ell(f(x,t),y)\right] \tag{1}$$

The objective is to minimize the average error across multiple tasks, where $\ell(f(x,t),y)$ represents the loss function measuring the discrepancy between the model's prediction and the true label. First, the average loss is computed for each task $t$ under its data distribution $P_t$ using the expectation operator. Then, the overall average error across all tasks is calculated to evaluate the model's performance in a multi-task setting.

The average accuracy of a predictor is defined as

$$Accuracy_t = \frac{1}{t}\sum_{j=1}^{t} a_{t,j} \quad t \in \{1,\ldots,T\} \tag{2}$$

where $a_{i,j}$ denotes the test accuracy on task $j$ after the model has finished experiencing task $i$. The average accuracy for each task is calculated by averaging the test accuracies over all tasks $j$ such that $j \leq i$.

The final maximum forgetting is defined as:

$$Forgetting = \frac{1}{T-1}\sum_{j=1}^{T-1}\max_{l\in\{1,\ldots,T-1\}}\left(a_{l,j} - a_{T,j}\right) \tag{3}$$

The formula calculates the average performance degradation for each task by comparing the peak performance during training to the final performance after all tasks have been learned. A higher forgetting value indicates a more severe forgetting problem in continual learning.

## 3 Datasets, Task Generation, and Task Learning Order Determination

### 3.1 Medical Waste 4.0

The dataset **Medical-Waste-4.0-Dataset** (Bruno et al. (2023a)) was created by Bruno Antonio, Massimo Martinelli, and Davide Moroni following a selection process of the most commonly used types of medical items in hospitals. The data was collected using new medical equipment to simulate medical waste. This dataset (Bruno et al. (2023b)) was collected as part of the "Medical Waste Treating 4.0" project funded by the Tuscany Region, aimed at supporting research related to medical waste management, particularly in the fields of machine learning and computer vision.

The dataset includes the following categories : gauze, latex gloves (single and pairs), nitrile gloves (single and pairs), surgical gloves (single and pairs), medical caps, medical goggles, shoe covers (single and pairs), test tubes, and urine bags.

| Classification | Sample size |
|---|---|
| Gauze | 393 |
| Glove pair latex | 330 |
| Glove pair nitrile | 330 |
| Glove pair surgery | 300 |
| Glove single latex | 303 |
| Glove single nitrile | 333 |
| Glove single surgery | 306 |
| Medical cap | 306 |
| Medical glasses | 318 |
| Shoe cover pair | 351 |
| Shoe cover single | 312 |
| Test tube | 363 |
| Urine bag | 300 |
| **Total dataset size** | **4,245** |

Table 1: Dataset category and corresponding sample size

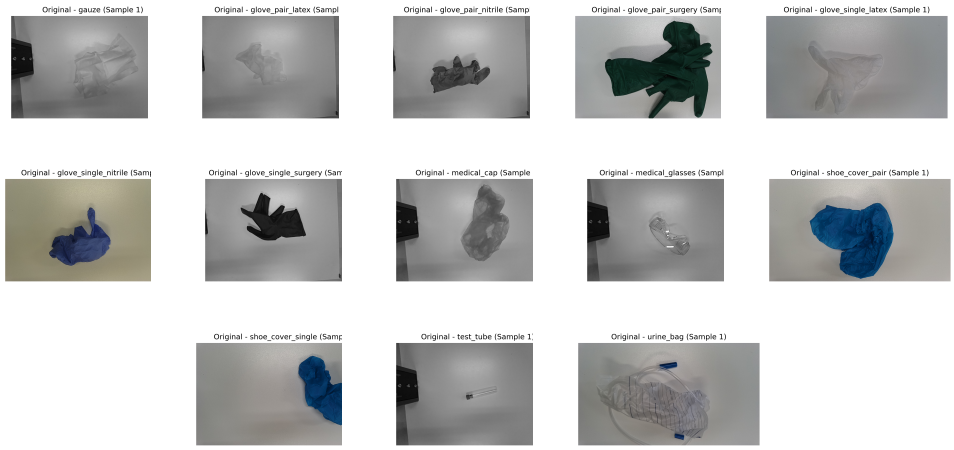



Figure 3: Original Dataset Samples

Each sample in the dataset consists of three images : a high-resolution RGB image (1920 x 1080) and a stereo pair of grayscale images, each with a resolution of 640 x 400. The naming conventions for the files are as follows : timestamp.jpg for the RGB image, timestamp_r.png for the right view, and timestamp_l.png for the left view in the stereo pair. A public dataset containing over 1400 image triplets has already been released, while a more structured dataset with over 2100 image triplets will be published in the near future.

This dataset is designed to be a valuable resource for devising and testing computer vision methods for the primary sorting of medical waste.

### 3.2 Semantic-based Task Generation

To simulate the scenario of a continual learning model when deployed in the real world, we classified the categories into three distinct tasks based on their functionality, material characteristics, and usage state : gloves (Task 1), shoe covers (Task 2), and other medical items (Task 3). Other task organizations are also possible, depending on the specific application or experimental requirements.

**Task 1** includes medical gloves, further divided by material type (latex, nitrile, and surgical-specific) and usage state (pairs or singles). This classification is based on the material properties and the form

in which they are discarded, facilitating sorting and processing. This category includes the following items :

- `glove_pair_latex`
- `glove_pair_nitrile`
- `glove_pair_surgery`
- `glove_single_latex`
- `glove_single_nitrile`
- `glove_single_surgery`

By organizing gloves based on material, this task enables a more granular understanding of glove waste, such as distinguishing disposable latex gloves from the more robust surgical ones. This classification is crucial for optimizing recycling or disposal processes based on material types.

**Task 2** consists of medical shoe covers, categorized by whether they are paired or single. This straightforward classification reflects the basic form differences of shoe covers during usage or disposal. This category includes the following items :

- `shoe_cover_pair`
- `shoe_cover_single`

The separation between single shoe covers and pairs mirrors their practical use in hospitals and clinics, where single covers might be used as spares, while pairs are standard for operations. Grouping shoe covers as a standalone task highlights their unique role and simplifies the identification of footwear-related waste.

**Task 3** encompasses other medical items such as urine bags, gauze, medical caps, medical glasses, and test tubes. These are categorized based on their unique functionalities (e.g., liquid collection, protection, experimental purposes) and item-specific characteristics, addressing different processing needs. This category includes the following items :

- `urine_bag`
- `gauze`
- `medical_cap`
- `medical_glasses`
- `test_tube`

This grouping reflects the functional diversity of these items while ensuring their waste characteristics are considered together. These items are not as frequently disposed of as gloves or shoe covers but still constitute a significant portion of medical waste, particularly in diagnostic and procedural settings.

Overall, this separation of tasks ensures a logical organization of the dataset, improving the clarity and usability for downstream machine learning tasks. Each task aligns with specific medical waste management processes, facilitating targeted analysis and model development for different categories of medical waste.

### 3.3 Hierarchical Clustering-based Task Generation

To improve task classification performance and transcend traditional semantic-based methods, we adopted two approaches :

1. Euclidean Distance Analysis : Leveraging Euclidean distance to measure the proximity between tasks.
2. Task2Vec Embedding and Similarity Measurement : Generating task embeddings using Task2Vec, followed by assessing the relationships between these embeddings through cosine similarity.

These methodologies facilitated the construction of distance matrices and similarity matrices that effectively represent relationships between categories or embeddings. Based on these matrices, hierarchical clustering was employed to group 13 categories. To ensure the quality of the clustering results, we conducted a comprehensive evaluation with the Silhouette score as the primary criterion. The number of clusters was determined by selecting the configuration with the most appropriate Silhouette score, ensuring a balance between intra-cluster cohesion and inter-cluster separation.

### 3.3.1 EUCLIDEAN DISTANCE

In the initial stage, the ResNet34 model was pre-trained on the entire dataset to learn task-specific feature representations. After pre-training, the classification head of the model was removed and the remaining layers were utilized as a feature extractor. The feature extractor was used to extract high-dimensional features from each sample in the dataset. By aggregating the features of samples belonging to the same class, the class centroids were computed to represent the central position of each class in the feature space. Subsequently, the Euclidean distances between class centroids were calculated to quantify the similarity between classes. Based on the resulting distance matrix, hierarchical clustering was applied to group the classes, uncovering the latent structure within the dataset.

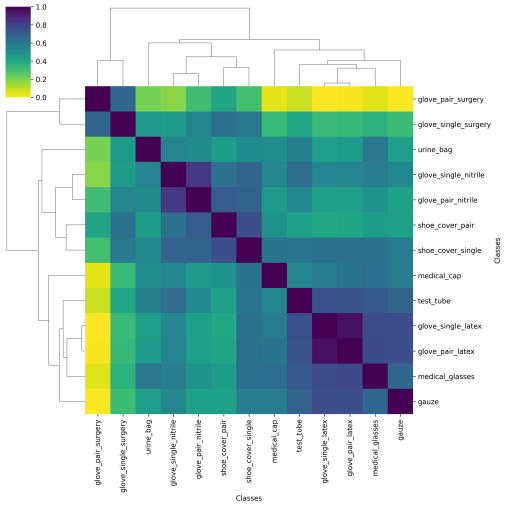



Figure 4: Similarity matrix obtained using Euclidean Distance

Figure 4 illustrates the similarity matrix . . . The color values in the matrix represent the similarity between categories. Darker colors (closer to purple) indicate higher similarity, while brighter colors (closer to yellow) represent lower similarity. This matrix reflects the directional or representational consistency between categories.

The similarity matrix measures the proportional relationship of distances between categories, reflecting the directional consistency in the feature space. Categories with higher similarity usually have closer feature representations and are therefore more likely to be grouped together. From the heatmap and the hierarchical clustering dendrogram, the classification results are consistent with those from the distance matrix. This indicates that the feature space achieves stable classification results under both distance and similarity measures.

From the heatmap and the hierarchical clustering dendrogram, it can be observed that "glove_pair_latex" and "glove_single_latex" are grouped together due to their small distance. Overall, categories with smaller distances are grouped into the same cluster. This demonstrates that the feature extractor effectively captures the geometric relationships between categories and achieves clear category differentiation in the feature space.

### 3.3.2 TASK2VEC-BASED DISTANCE

**Task2vec (Achille et al., 2019).** Task2vec is a task embedding method that captures the statistical properties and feature distributions of task datasets to generate high-dimensional vector representations. These embeddings not only quantify the similarity between tasks but also intuitively reflect the semantic and classification relationships among tasks.

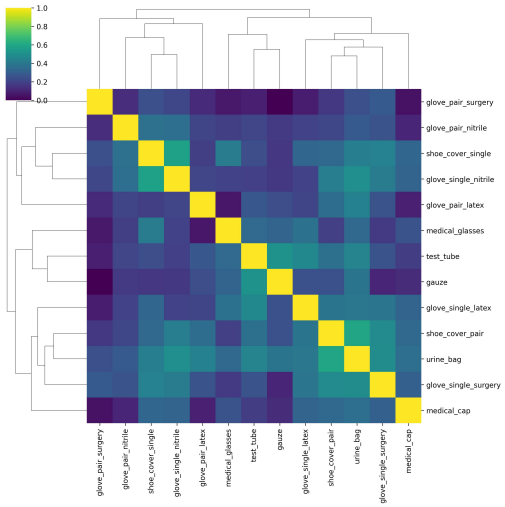



Figure 5: Similarity matrix ordered by hierarchical clustering obtained using the Cosine Distance between the Task embeddings produced by Task2vec.

In our implementation, Task2Vec utilizes a pretrained ResNet34 as a probe network, generating task embeddings by computing the diagonal of the Fisher Information Matrix.

Specifically, in the *Medical Waste 4.0* dataset, we employed the Task2vec method to generate embeddings for each category (treating each category as an independent task). These embeddings effectively quantify the similarity between tasks and almost align with human intuition regarding the semantic and classification relationships among tasks.

For example, classification tasks for gloves made of different materials, such as "glove_pair_latex", "glove_pair_nitrile", and "glove_pair_surgery", exhibit higher similarity in the embedding space due to shared visual features like texture, shape, and color. As shown in the similarity matrix, these categories form closely related clusters, reflecting the difficulty in distinguishing between them based solely on visual features. However, two glove-related categories, "glove_single_latex" and "glove_single_surgery", were grouped into a separate cluster despite their visual similarities to other glove types. This indicates that the clustering was influenced not solely by visual features but potentially by additional task-specific factors or subtle differences in their embedding representations.

To further analyze the relationships between these task embeddings more intuitively, we constructed distance matrices and similarity matrices based on the generated embeddings, as illustrated in the figure 5 above.

Figure 5 illustrates the similarity matrix ... The color values in the matrix represent the cosine similarity between categories. Darker colors (closer to purple) indicate lower similarity, while lighter colors (closer to yellow) indicate higher similarity.

Cosine similarity directly measures the directional consistency between categories. Categories with higher similarity are more likely to be grouped together. From the heatmap and the hierarchical clustering dendrogram, it can be observed that categories with higher similarity are grouped together. For instance, "shoe_cover_pair" and "urine_bag", as well as "shoe_cover_single" and "glove_single_nitrile", have higher similarity and are therefore clustered into the same group. The classification results are consistent with the categories with smaller cosine distances, which verifies the stability of the feature extractor under different calculation methods.

### 3.3.3 T-SNE DIMENSIONALITY REDUCTION

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a nonlinear dimensionality reduction algorithm commonly used for visualizing high-dimensional data. To intuitively reveal the internal structure of the data and assist with clustering tasks, we use t-SNE to reduce the high-dimensional data to two dimensions for visualization. Clustering labels are generated based on the hierarchical clustering linkage matrix and the specified number of clusters, with each category assigned a corresponding cluster label. The hessian attributes of the embeddings (containing key information) for each category are flattened and stored as feature vectors. Finally, t-SNE is applied to reduce these feature vectors to a two-dimensional space, enabling a clear and intuitive visualization of the clustering results.

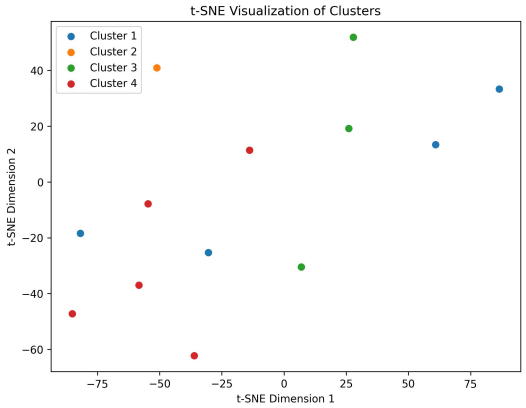
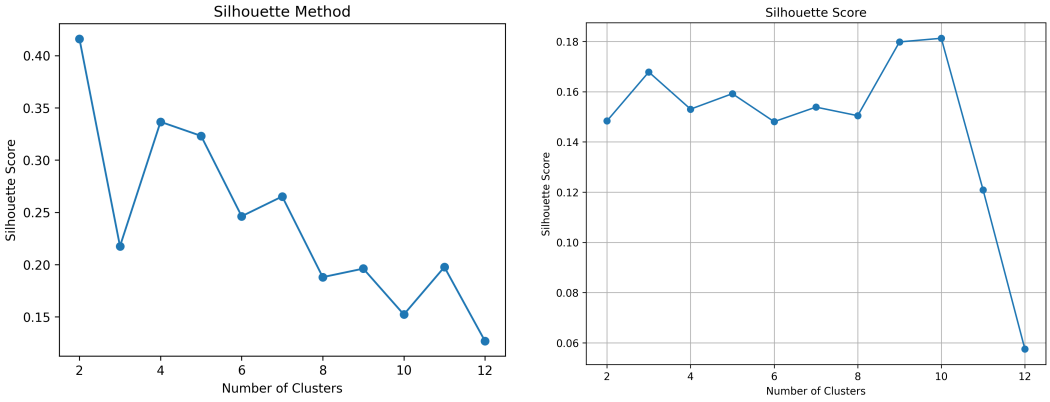



Figure 6: T-SNE visualisation of the embedding of tasks

As shown in the figure above 6, it is evident that there is significant spacing between the data points, with large gaps between the clusters and minimal overlap, demonstrating excellent class separability. The clusters are relatively dispersed in the two-dimensional space, yet the overall distribution is uniform. The dimensionality reduction process effectively avoids excessive compression while preserving the integrity of global information. Specifically, the points for Task 4 are primarily distributed in the lower-left region, Task 2's points are concentrated in the upper-left, Task 3's points are located in the center, while Task 1's points are relatively scattered, with some overlapping into the clusters of Task 4 and Task 2.

### 3.3.4 COMPARISON




(a) Plot silhouette-Euclid

(b) Plot silhouette-Task2vec

Figure 7: Plot silhouette

The left image 7a illustrates the Silhouette scores calculated based on Euclidean distance as the number of clusters increases from 2 to 12. The Silhouette score exhibits a decreasing trend as the number of clusters increases. At 2 clusters, the Silhouette score reaches its peak, close to 0.4, indicating strong intra-cluster consistency and good inter-cluster separation at this point. However, as the number of clusters increases, the scores gradually decline, suggesting that a larger number of clusters may reduce inter-cluster separation and degrade clustering quality.

The right image 7b shows the Silhouette scores based on Task2Vec embeddings as the number of clusters increases. Compared to the Euclidean method, the overall Silhouette scores of Task2Vec are lower, with the highest score being around 0.18 (at 10 clusters). Between 2 and 10 clusters, the scores exhibit relatively small fluctuations, indicating that Task2Vec embeddings are less sensitive to changes in the number of clusters. This also implies that the similarity among tasks represented by the embeddings remains more stable across varying cluster numbers. However, when the number of clusters reaches 12, the Silhouette score drops sharply to 0.06, indicating that too many clusters significantly diminish inter-cluster separation and affect clustering validity.

Ultimately, the optimal number of clusters obtained using the Euclidean method is 2, where inter-cluster separation and intra-cluster consistency are maximized. For the Task2Vec method, the optimal number of clusters is 10. Although the Silhouette score for this point is lower, the embeddings provide more balanced cluster distributions. Given the complex feature distribution of the tasks, Task2Vec embeddings offer more stable clustering support. Despite their lower Silhouette scores, they can better capture the relationships between tasks, making the Task2Vec results more favorable in our analysis.

After weighing the Silhouette score trends from both methods, we decided to use 4 clusters for subsequent continual learning experiments. This decision was made because using 10 clusters would result in a large number of tasks containing only a single class, which would undermine the credibility of the experimental results. On the other hand, splitting the dataset into only 2 clusters in a continual learning context would fail to reflect the characteristics of continual learning. Additionally, the Silhouette score for 3 clusters is not high when using the Euclidean method.

In future work, we will further verify the effectiveness of other cluster numbers and conduct more in-depth tests on their clustering performance.

#### 3.3.5 Results of Task Generation

**Task 1** includes additional medical waste items grouped to streamline disposal and management processes. The included classes are:

- `glove_pair_latex`
- `glove_pair_nitrile`
- `glove_single_nitrile`
- `shoe_cover_single`

**Training set:** 1044 instances
**Validation set:** 261 instances

**Task 2** focuses on surgical gloves specifically designed for medical surgeries, aiding in the efficient handling and categorization of these items. The included class is:

- `glove_pair_surgery`

**Training set:** 240 instances
**Validation set:** 60 instances

**Task 3** includes general medical supplies categorized by their material and functional use, designed to streamline processing and sorting. The included classes are:

- `gauze`

- medical_glasses
- test_tube

**Training set:** 859 instances
**Validation set:** 215 instances

**Task 4** includes a variety of medical items classified based on their usage and functional purpose. This category facilitates efficient sorting and processing. The included classes are:

- glove_single_latex
- glove_single_surgery
- shoe_cover_pair
- urine_bag
- medical_cap

**Training set:** 1252 instances
**Validation set:** 314 instances

### 3.4 Generation of the Learning Tasks Order

In the previous part, we saw that using an appropriate task embedding generator and a distance metric on these embeddings, one can generate tasks that better capture the intrinsic empirical relationships across the categories. However, the order in which the resulting tasks should be learned in a continual fashion is not straightforward. To further leverage the transferability between tasks, we employed the Log Expected Empirical Prediction (LEEP) (Nguyen et al., 2020) score to quantify the knowledge transfer potential between source and target tasks. This metric enables us to prioritize tasks with higher transfer potential, thereby optimizing the training sequence and laying a foundation for applications in continual learning scenarios.

**LEEP (Nguyen et al., 2020).** LEEP (Log Expected Empirical Prediction) is a lightweight evaluation metric primarily used to quantify the adaptability of pre-trained models to target tasks. Its core idea is to calculate the log expectation of target task labels under the predictions of the pre-trained model, thereby reflecting the transfer potential of the model. Compared to traditional methods, LEEP does not require fine-tuning the pre-trained model, making it computationally more efficient.

#### 3.4.1 Learning Order Generation Algorithm

In continual learning, reasonable task division and learning order are critical for improving the adaptability of models. We propose a task learning order generation algorithm based on a greedy strategy. This algorithm uses LEEP scores as a basis, prioritizing task pairs with higher scores and gradually selecting locally optimal solutions to construct a coherent task chain. This approach ultimately generates a task learning order that achieves near-global optimal results with low computational cost.

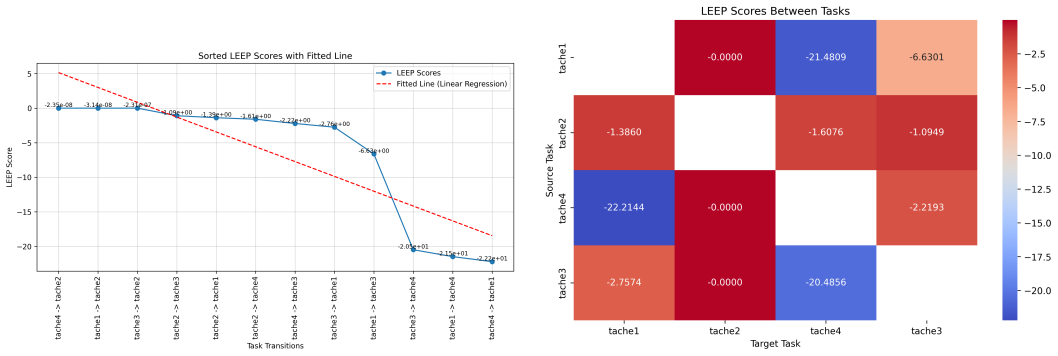
Algorithm steps are as follows :

1. Sort all task pairs in descending order based on their LEEP scores.
2. Initialize an empty chain and maintain sets for used source tasks and target tasks.
3. Starting from the highest-scoring task pair, iteratively check whether it can be added to the chain (unused tasks or those meeting source/target conditions).
4. Stop the selection process when the chain length reaches a predefined limit.
5. Extract the starting task from the chain and generate the final task learning order based on the chain relationships.

### 3.4.2 RESULTS OF TASK ORDERING

To showcase the proposed process, we selected 4 as the number of clusters and applied hierarchical clustering to assign each class to a corresponding cluster label. Subsequently, categories sharing the same cluster label were grouped into the same task list, thus completing the process of task reorganization and generation.

To determine the optimal task transfer sequence suitable for continual learning scenarios, we conducted pairwise analyses of the newly generated 4 tasks and calculated the LEEP (Log Expected Empirical Prediction) score between each pair. This metric effectively evaluates the knowledge transfer potential between tasks, allowing us to design a more reasonable training sequence and improve the overall efficiency of continual learning.




(a) Leep Score Line

(b) Leep Score Matrix

Figure 8: Plot silhouette

The final LEEP scores are shown in the figure, where the scores between each pair of tasks are visualized using a line plot and a heatmap. The line plot demonstrates the trend of task transfer scores, while the heatmap provides an intuitive representation of the transferability distribution among tasks.

**LEEP Score Line Plot :**

In the line plot 8a on the left, the x-axis represents task transfer pairs (e.g., "tache4 - tache2" indicates transferring from task 4 to task 2), while the y-axis corresponds to the respective LEEP scores.

To more clearly display the relative magnitude of the LEEP scores between task pairs, task transfers with higher scores (e.g., "tache4 - tache2") are placed on the left, and those with lower scores are positioned on the right. This arrangement highlights that the initial task pairs have higher knowledge transfer potential, making them suitable as priority tasks in training. Conversely, task pairs with lower scores (e.g., "tache4 - tache3") suggest that transferring between these tasks may negatively impact model performance.

Additionally, the red dashed line represents a linear regression fit, which clearly shows that the overall transfer potential between tasks diminishes as the transfer sequence progresses.

**LEEP Score Heatmap Matrix :**

In the heatmap matrix 8b on the right, the rows represent source tasks, while the columns represent target tasks. The color gradient in the heatmap indicates the LEEP score for each task pair, ranging from red (low scores) to blue (high scores).

From the figure, it can be observed that certain task pairs (e.g., "tache4 - tache2" and "tache4 - tache1") exhibit significant negative scores (dark blue), indicating poor transfer performance when transferring from these source tasks to target tasks. The diagonal elements are all zero, as self-transfer scores are meaningless.

The heatmap provides a clear visualization of the knowledge transfer potential between tasks, revealing considerable differences in transferability across task pairs. For instance, “tache4” as a source task consistently shows low scores, suggesting that transfers originating from task 4 are generally ineffective. In contrast, the LEEP scores between “tache2” and “tache3” are closer to zero, indicating relatively stable transfer potential, albeit with limited overall transfer benefits.

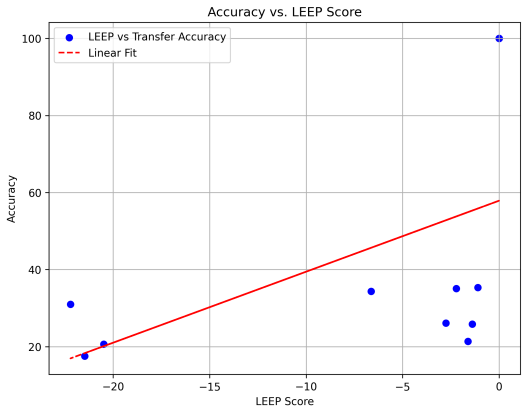



Figure 9: LEEP scores vs. test accuracies

Figure 9 illustrates the relationship between LEEP scores and test accuracies, based on task transfer experiments conducted on the Medical Waste 4.0 dataset. We used ResNet34 as the base model, dividing the dataset into four tasks (taches) and computing directed LEEP scores for all task pairs, resulting in 12 data points. The blue points represent the correspondence between LEEP scores and transfer accuracies, while the red line indicates the linear fit.

The experimental results show that there is a certain degree of positive correlation between LEEP scores and transfer accuracies, with higher LEEP scores corresponding to better target task model performance in some cases. This suggests that LEEP scores hold some value as a predictive metric for transfer learning performance in the medical waste classification domain. However, some deviations from the linear fit are observed, which may be attributed to task-specific characteristics or differences in inter-task feature distributions, indicating that further research and optimization are still needed.

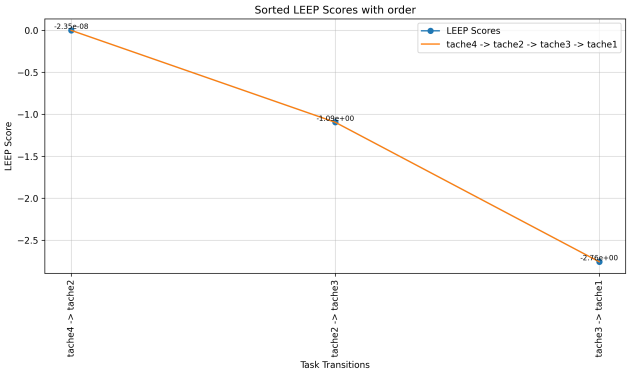



Figure 10: Final task order

Finally, based on the comprehensive analysis results and the evaluation of knowledge transfer potential between tasks, we selected the top three task pairs with the highest overall scores and determined the final task transfer order. As shown in Figure 10, the final task transfer order is **tache4 - tache2 - tache3 - tache1**.

The figure 10 reflects the knowledge transfer potential between tasks. The orange line illustrates the task transfer sequence and the trend of LEEP scores. It can be observed that the LEEP scores gradually decrease as the transfer progresses, indicating that the transfer potential between tasks diminishes over the sequence. Therefore, we prioritized task pairs with higher LEEP scores as the starting point of the transfer to maximize transfer efficiency and optimize training outcomes.

By integrating the results of hierarchical clustering with transferability analysis, we refined the task generation process, determined a new task training sequence, and provided a more structured solution for task handling.

# 4 EXPERIMENTS

In the process of building and optimizing a model, selecting an appropriate training method is crucial. Different training strategies significantly impact the model's performance, convergence speed, and computational resource requirements. To achieve the final goal, we designed and conducted a series of experiments to explore the characteristics and applicability of various strategies, including ***full-batch training***, ***independent task training***, ***multi-task learning*** and ***continual learning***. The following sections will provide a detailed explanation of each method's design approach, experimental process, model architecture, and corresponding result analysis.

## 4.1 EXPERIMENTAL DETAILS

**Data preprocessing.** In convolutional neural networks, input images are required to have a fixed size, as convolutional operations and fully connected layers rely on consistent input dimensions. Mismatched image sizes can result in dimensional incompatibility errors. Furthermore, when models are trained in batches, each batch must consist of uniformly sized samples to facilitate efficient processing.

However, image datasets often originate from diverse sources with varying resolutions and aspect ratios, ranging from high-resolution images (e.g., 1920×1080) to low-resolution ones (e.g., 640×400). By applying the Resize operation during data loading, all images can be adjusted to a consistent size, thereby simplifying data preprocessing, ensuring uniformity across samples, and improving the overall efficiency of training and processing pipelines (see Figure **??** in Appendix for examples).

Additionally, while high-resolution images capture more details, they significantly increase memory usage and computational costs. For many tasks, retaining the full detail of the original resolution is unnecessary. Thus, resizing images to an appropriate size effectively reduces computational overhead and optimizes model performance.

**Architecture and training details.** *Seed* : To ensure reproducibility and stability across experiments, a fixed random seed of 0 is used. The seed is consistently applied across all components, including NumPy, Python's built-in random module, and PyTorch (both CPU and GPU operations). Additionally, PyTorch's CUDNN backend is configured with deterministic=True and benchmark=False to eliminate non-deterministic behaviors during computation. By setting the environment variable PYTHONHASHSEED, we ensure that hash-based operations are also reproducible. These settings guarantee consistent dataset splits, data loading, and model initialization across runs, providing a stable basis for analyzing experimental results.

*Hyper-parameter* : The model is a 3-layer convolutional neural network, with 80% of the data used for training and 20% for validation. The first convolutional layer outputs 16 feature channels (a total of 37,696 neurons), and the second layer outputs 32 feature channels (a total of 69,120 neurons). The activation function is ReLU, and no Dropout regularization is applied. The learning rate is set to 0.001, and the Adam optimizer is used to update the weights efficiently and stably. These configurations balance the model's representation capability and training efficiency.

| Hyper-parameter | Value |
|---|---|
| Validation split | 0.2 |
| Number of layers | 3 |
| Number of Neurons in Conv1 | $16 \times 38 \times 62 = 37{,}696$ |
| Number of Neurons in Conv2 | $32 \times 36 \times 60 = 69{,}120$ |
| Activation function | ReLU |
| Dropout | Not used |
| Learning rate | 0.001 |
| Optimizer | Adam |
| Batch size | 16 |

Table 2: Hyper-parameter of model

The hyper-parameter values used in 5 experiments are shown in the table 2.

### 4.2 Experiment 1 : Full-Batch Training (Flatten)

In the full-batch training method, the entire dataset is treated as a single entity without task partitioning. During the training process, data batches are iteratively drawn from the entire dataset and fed into the model until the training is complete. This approach offers several advantages: it enables a comprehensive evaluation of the model's learning ability across the entire dataset, provides a clear and intuitive view of its overall performance during global training, and establishes a reliable baseline for comparison. This baseline serves as a critical reference point for assessing the effectiveness and improvements introduced by other training methods.

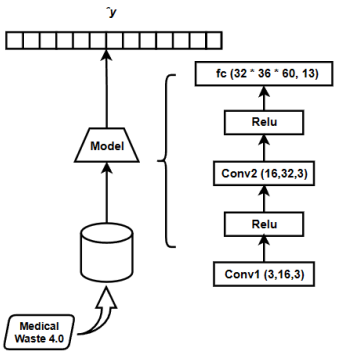



Figure 11: Flatten Model Architecture

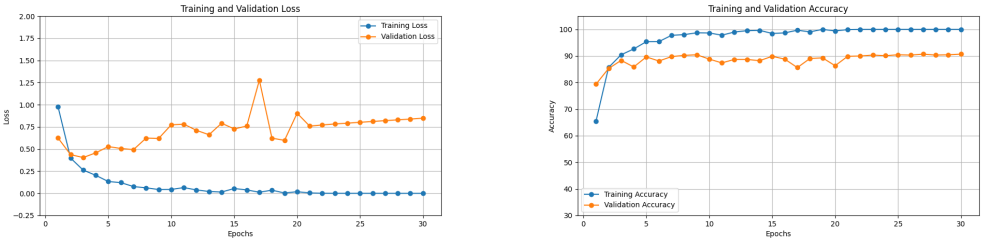
To better illustrate the training process under this strategy, we have drawn the model structure shown in the figure 11. This figure illustrates the architecture of the model applied to the "Medical Waste 4.0" dataset, with a focus on the flattening process within the network. The raw input data is processed through multiple layers, including convolutional layers (Conv1, Conv2) and activation functions (ReLU), before being flattened into a one-dimensional vector. This vector is then passed into fully connected (fc) layers for classification, ultimately producing the output y, which represents the predicted categories of medical waste.

| Hyper-parameter | Value |
|---|---|
| Number of Neurons in FC | 13 |

Table 3: Hyper-parameter of model flatten

The hyper-parameter values used in the experiment are shown in the table 3.

### 4.2.1 Analysis of Model Training Methods




(a) Loss-flatten (b) Accuracy-flatten

Figure 12: Comparison of Loss and Accuracy

**Accuracy and Loss Rate Analysis**

From these two graphs 12, which describe the evolution of **accuracy** and **loss** during training and validation, the following conclusions can be drawn :

*Training and Validation Loss (Loss-flatten [12a])*

- Training loss decreases rapidly at the beginning, showing that the model effectively learns the features of the training data.
- Validation loss follows a similar downward trend initially but stagnates between epochs 8 and 15, then starts to fluctuate and even increase (notably around epoch 25).

*Training and Validation Accuracy (Accuracy-flatten [12b])*

- Training accuracy increases rapidly and eventually reaches nearly 100%, indicating a strong ability of the model to fit the training data.
- Validation accuracy improves steadily during the first 8 epochs but starts to fluctuate between epochs 8 and 15, failing to improve further. It remains lower than training accuracy, stabilizing around 90%.

**Phenomenon Analysis**

*Model Performance*

- The model performs excellently on the training set (*training accuracy close to 100%, training loss close to 0*), demonstrating a strong ability to fit the training data.
- However, performance on the validation set is less satisfactory (*validation accuracy remains below 90% with fluctuations, and validation loss increases*), revealing limited generalization capability on unseen data.

*Overfitting Phenomenon*

- Starting from epoch 8, the model shows signs of overfitting (Li et al. (2024)) : training loss continues to decrease, while validation loss ceases to decrease and begins to fluctuate or even increase.
- Training accuracy is significantly higher than validation accuracy, and validation performance stabilizes or deteriorates, confirming that the model overfits the training data in the later epochs.

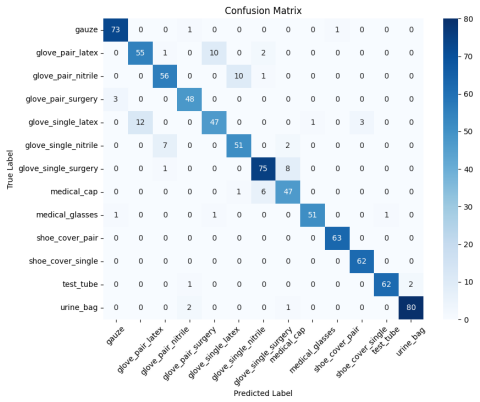



Figure 13: Confusion matrix-flatten

**Confusion Matrix Analysis**

From the confusion matrix, it is evident that the model performs well in most categories, but there are some misclassification issues in certain easily confused categories. Notably, there is significant confusion between glove-related categories, such as glove_pair_latex and glove_single_latex, as well as glove_single_nitrile and glove_pair_nitrile, likely due to their similar material or physical characteristics, with noticeable confusion in distinguishing the number of gloves. Additionally, there is some misclassification between medical_cap and medical_glasses, indicating that the model struggles to differentiate the features of these categories effectively.

### 4.3 Experiment 2 : Independent Task Training

Consistent with the data partitioning approach used in multi-task learning, we also divided the dataset into three tasks. However, in this method, each task is trained independently. We trained a separate model for each task and evaluated the impact of task separation on model performance by recording the accuracy and loss values for each task. This experiment also serves as a baseline for comparison with other methods.

This method aims to evaluate the impact of task separation on model training performance and to identify optimization opportunities in scenarios where tasks operate independently without interference. To ensure clarity and facilitate comparisons, a corresponding model diagram is included, providing a visual representation of the independent task training process.

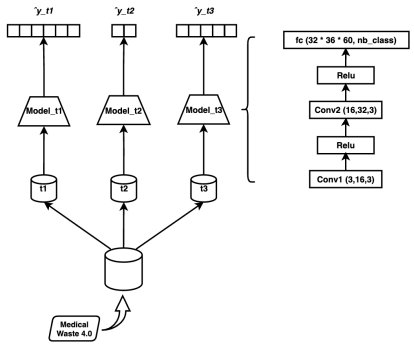



Figure 14: Independent Task Model Architecture

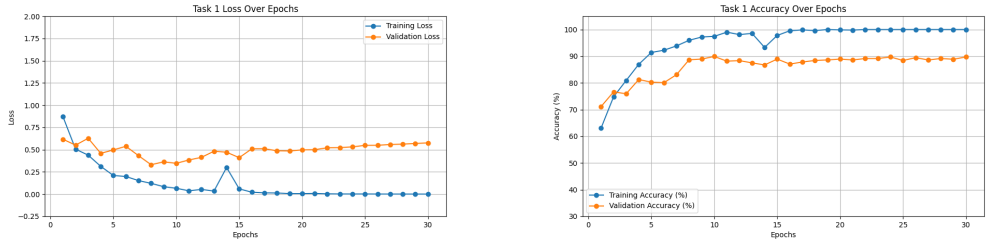
The diagram 14 shows how the dataset is divided into separate tasks, with each task assigned to an individual model. This design ensures that tasks operate independently, allowing for a systematic assessment of task separation and its influence on overall model performance.

| Hyper-parameter | Value |
|---|---|
| Number of Neurons in FC_T1 | 6 |
| Number of Neurons in FC_T2 | 2 |
| Number of Neurons in FC_T3 | 5 |

Table 4: Hyper-parameter of model specific

The hyper-parameter values used in the experiment are shown in the table 4.

#### 4.3.1 Analysis of Model Training Methods




(a) Loss-task_1 (b) Accuracy-task_1

Figure 15: Loss and Accuracy of task_1

**Accuracy Rate Analysis**

Training accuracy rises rapidly and plateaus near 100% after the 15th epoch, indicating excellent performance on the training dataset.

Validation accuracy stabilizes around 90% after the 10th epoch but remains lower than training accuracy, suggesting possible overfitting.

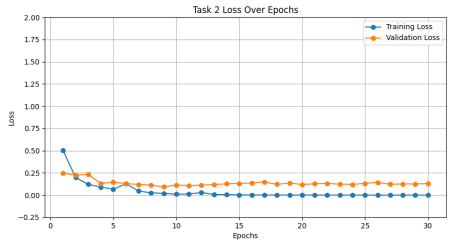
**Loss Rate Analysis**

Training loss decreases rapidly and approaches zero, indicating minimal error on the training dataset.

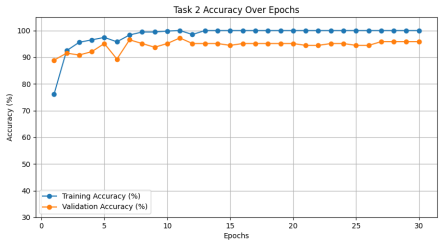
Validation loss stabilizes after an initial decline but remains significantly higher than training loss, with some fluctuations, further highlighting limited generalization ability.

**Phenomenon Analysis**

The model performs well on the training dataset but shows signs of overfitting on the validation dataset.




(a) Loss-task_2




(b) Accuracy-task_2

Figure 16: Loss and Accuracy of task_2

**Accuracy Rate Analysis**

Training accuracy rises quickly in the first few epochs and stabilizes near 100% after the 5th epoch, indicating excellent performance on the training dataset.

Validation accuracy increases rapidly in the initial epochs and stabilizes around 90%-92%, slightly lower than training accuracy, suggesting mild overfitting.

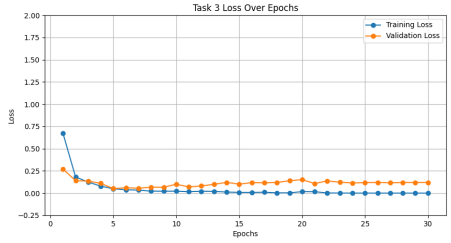
**Loss Rate Analysis**

Training loss decreases sharply during the first few epochs and approaches zero, showing minimal error on the training data.

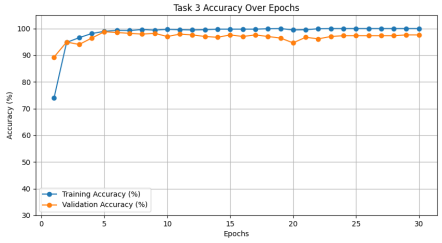
Validation loss drops initially and stabilizes at a low level (around 0.1 to 0.2), with a small gap compared to training loss, indicating good generalization without significant overfitting.

**Phenomenon Analysis**

The model demonstrates excellent training performance and good generalization on the validation set.




(a) Loss-task_3




(b) Accuracy-task_3

Figure 17: Loss and Accuracy of task_3

**Accuracy Rate Analysis**

Training accuracy increases rapidly and stabilizes near 100% after the 5th epoch, demonstrating excellent learning on the training set.

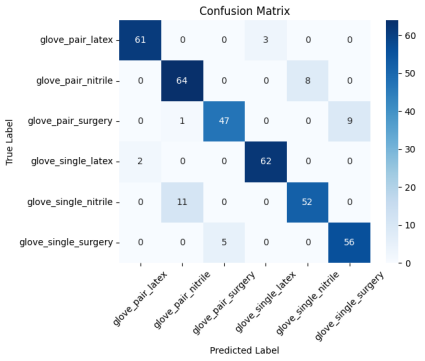
Validation accuracy stabilizes around 90%-92%, slightly lower than training accuracy, indicating mild overfitting but overall good generalization.

**Loss Rate Analysis**

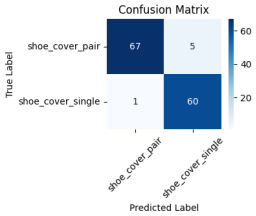
Training loss drops sharply in the early epochs and approaches zero, reflecting minimal error on the training data. Validation loss stabilizes at a low level (around 0.1 to 0.2), with a small gap compared to training loss, suggesting the model generalizes well.

**Phenomenon Analysis**

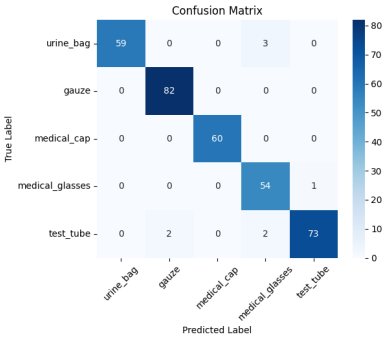
The model achieves excellent training performance and good validation accuracy.




(a) Confusion matrix-task_1




(b) Confusion matrix-task_2




(c) Confusion matrix-task_3

Figure 18: Confusion matrix

**Confusion Matrix Analysis**

**Task 1**

*Main Confusion :* There is significant confusion between glove_single_nitrile and glove_pair_nitrile, glove_single_latex and glove_pair_latex, as well as glove_single_surgery and glove_pair_surgery.

*Reason :* The confusion is primarily caused by the difficulty in distinguishing between single and paired gloves made of the same material.

**Task 2**

*Main Confusion :* There is minor confusion between shoe_cover_pair and shoe_cover_single, where 5 samples of shoe_cover_pair were misclassified as shoe_cover_single, and 1 sample of shoe_cover_single was misclassified as shoe_cover_pair.

*Reason :* The similar appearance of single and paired shoe covers, especially when limited image information is available, makes them difficult to distinguish.

**Task 3**

*Main Confusion :* urine_bag has 3 samples misclassified as medical_glasses. test_tube has 2 samples misclassified as gauze and medical_glasses.

*Reason :* These items may share similarities in certain angles or material properties, such as transparency or similar texture features.

### 4.4 Experiment 3 : Multi-Task Learning with Dynamic Batch size

Based on the multi-task learning framework, we divided the data into three related tasks. In each training iteration, a batch of data is dynamically sampled from each task. During training, the data first passes through the shared backbone for feature extraction and is then routed to different head models according to the task type for task-specific learning.

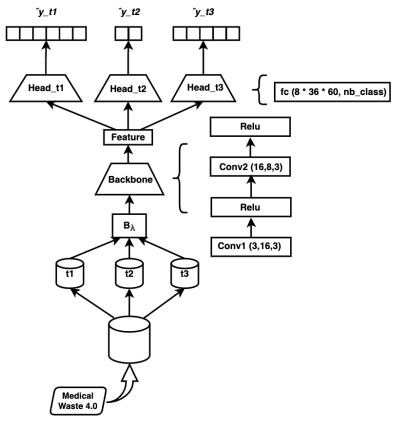



Figure 19: Multi-Task Learning Model Architecture. The model uses a shared backbone for feature extraction and task-specific heads (Head_t1, Head_t2, Head_t3) to produce outputs (y_t1, y_t2, y_t3). Input data (Medical Waste 4.0) is processed through convolutional and fully connected layers to optimize multi-task performance.

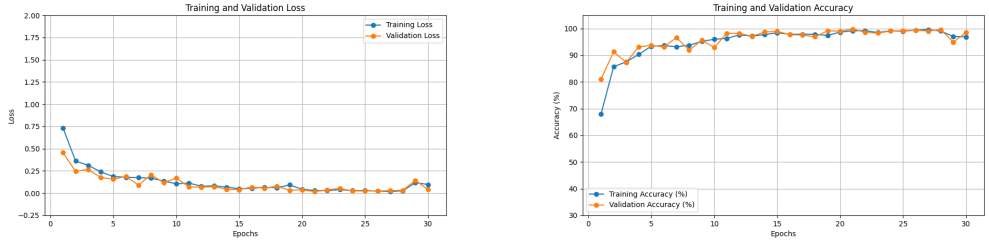
The figure 19 illustrates the structure of the backbone and head models, as well as the data flow between them under this strategy. The purpose of this strategy is to maximize the utilization of shared features across tasks while preserving the independent optimization direction of each task. We recorded the model's accuracy and loss variations for each task to analyze the collaborative effects among tasks and the learning efficiency of dynamic sampling.

| **Hyper-parameter** | **Value** |
|---|---|
| Number of Neurons in FC_T1 | 6 |
| Number of Neurons in FC_T2 | 2 |
| Number of Neurons in FC_T3 | 5 |
| Batch size of Task 1 | 23 |
| Batch size of Task 2 | 8 |
| Batch size of Task 3 | 28 |

Table 5: Hyper-parameter of model multitask

The hyper-parameter values used in the experiment are shown in the table 5.

### 4.4.1 Analysis of Model Training Methods




(a) Loss-MTL (b) Accuracy-MTL

Figure 20: Comparison of Training and Validation Metrics. Training and Validation Metrics for Multi-Task Learning (MTL). Subfigure (a) shows the training and validation loss trends, highlighting effective learning and convergence over 30 epochs. Subfigure (b) presents the training and validation accuracy trends, indicating the model's strong generalization and stability after the initial learning phase.

**Accuracy and Loss Rate Analysis**

From the two graphs 20, which describe the evolution of **accuracy** and **loss** during training and validation, the following conclusions can be drawn :

*Training and Validation Loss (Loss-MTL [20a])*

- Training loss decreases rapidly during the initial epochs and approaches 0 after epoch 10, indicating effective learning from the training data.
- The validation loss follows a similar decreasing trend initially but stabilizes after Epoch 10, with occasional fluctuations. Slight increases are observed around epochs 15 and 20, but these do not significantly affect the overall trend.

*Training and Validation Accuracy (Accuracy-MTL [20b])*

- Training accuracy increases rapidly in the first few epochs and eventually reaches 100%, demonstrating the strong ability of the model to fit the training data.
- The validation accuracy improves steadily during the first 10 epochs, reaching approximately 95%, and then stabilizes. Minor fluctuations are observed around epochs 15 and 20, but the overall trend remains stable.

**Phenomenon Analysis**

*Model Performance*

- The model performs exceptionally well on the training set (*training accuracy close to 100%, training loss close to 0*), showing excellent learning capabilities for the training data.
- Validation set performance is satisfactory, with validation accuracy stabilizing around 95% and validation loss remaining at a low level, indicating good generalization ability.

**Confusion Matrix Analysis**

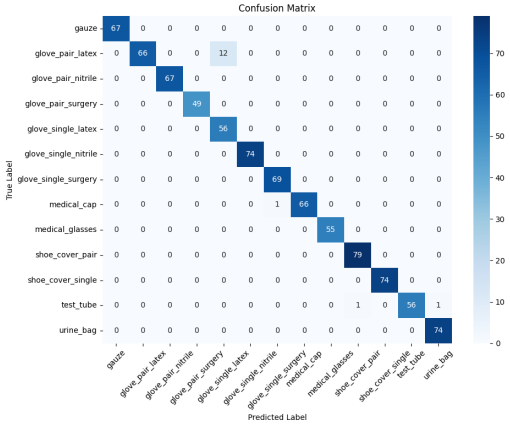



Figure 21: Confusion Matrix for Multi-Task Learning (MTL). The majority of predictions align with the diagonal, indicating high classification accuracy, while minor confusions occur in visually similar or imbalanced categories.

In the Multi-Task Learning (MTL) experiment, the model achieved the highest classification accuracy among all experiments, with the majority of predictions concentrated along the diagonal of the confusion matrix, demonstrating exceptional classification performance. Notably, for categories such as gauze, glove_pair_nitrile, and shoe_cover_pair, the model achieved nearly 100% accuracy, highlighting the advantages of the MTL approach in feature extraction and sharing. However, some confusion was observed for similar categories, such as glove_pair_latex and glove_pair_surgery, where 12 glove_pair_latex samples were misclassified as glove_pair_surgery, likely due to their highly similar appearance or features. Additionally, misclassification in a small number of cases, such as between test_tube and urine_bag, could be attributed to imbalanced data distribution or overlapping visual features. Overall, the results of the MTL experiment validate the effectiveness of the multi-task learning framework, demonstrating its ability to significantly enhance classification performance in dynamic environments.

**Advantages**

- In this approach, the dataset is split into multiple tasks, and during each training epoch, a batch of data is dynamically sampled from each task. Compared to the first method, where data from a single class is fed into the model in a concentrated manner, this dynamic sampling strategy demonstrates significant advantages in terms of training efficiency and generalization.

- Furthermore, in the head layer, task-specific outputs are separated by tasks, which further improves performance by leveraging task-specific distinctions.

### 4.5 Experiment 4 : Continual Learning

Sequential continual training introduces data gradually in the order of tasks. The model is first trained on Task 1 until the desired accuracy is achieved, after which data from Task 2 is added for further training, followed by the addition of Task 3 data. At each stage, we recorded the accuracy and loss values for both the training and test sets, observing whether the performance of previous tasks was maintained or affected.

This method simulates real-world scenarios where data arrives continually, focusing on the model's adaptability in continual learning and the interactions between tasks. The model diagram illustrates the process of gradually introducing continual data and its dynamic impact on the model's training structure.

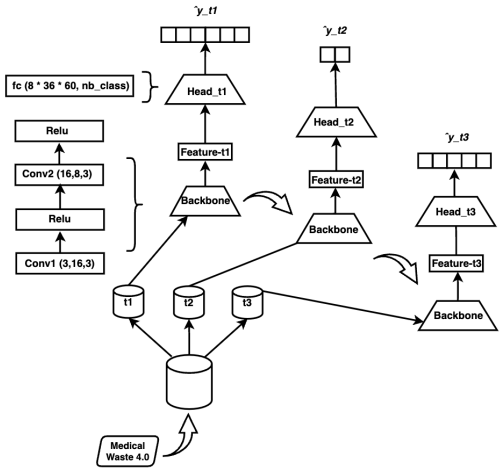



Figure 22: Continual Learning Model Architecture

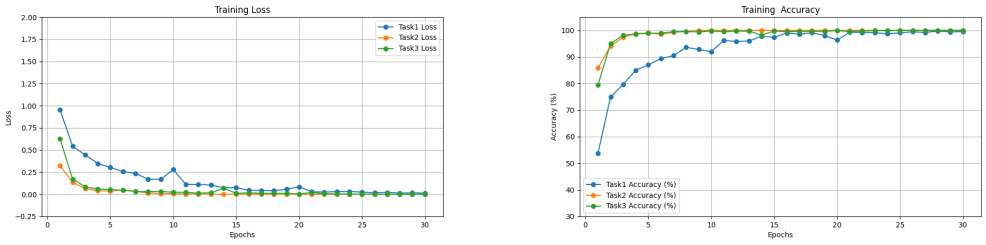
The figure 22 illustrates a continual learning model architecture, where the input dataset (Medical Waste 4.0) is divided into multiple tasks ($t_1, t_2, t_3$) to simulate incremental data scenarios. Each task's data passes through an independent backbone network to extract task-specific features ($Feature_{t1}, Feature_{t2}, Feature_{t3}$), which are then fed into corresponding task heads (Head) for classification or prediction. The backbone network consists of convolutional layers (Conv1 and Conv2), activation functions (ReLU), and fully connected layers (fc), processing input features step by step.

| **Hyper-parameter** | **Value** |
|---|---|
| Number of Neurons in FC_T1 | 6 |
| Number of Neurons in FC_T2 | 2 |
| Number of Neurons in FC_T3 | 5 |

Table 6: Hyper-parameter of model continual

The hyper-parameter values used in the experiment are shown in the table 6.

### 4.5.1 Analysis of Model Training Methods




(a) Loss-continual (b) Accuracy-continual

Figure 23: Loss and accuracy of Training

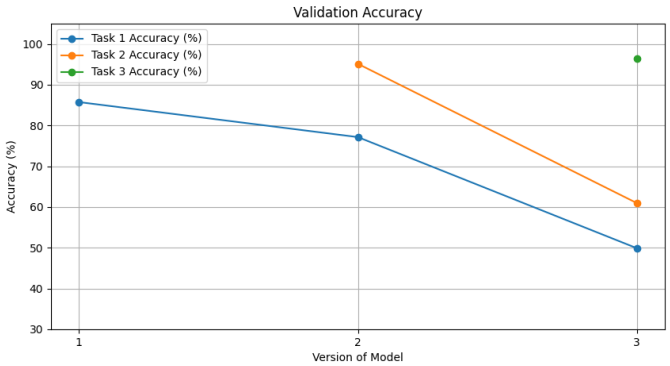



Figure 24: Accuracy of validation

**Accuracy and Loss Rate Analysis**

*Training Loss and Accuracy (Loss-Continual & Accuracy-Continual [23])*

- In the loss rate graph on the left, the loss values for all tasks decrease gradually over 30 epochs, indicating that the model is learning and converging.
- The initial loss for Task 1 is the highest, as it contains more data categories with higher complexity. However, by the end of training, the loss for all three tasks approaches zero.
- The loss for Task 2 and Task 3 decreases more rapidly, suggesting that these tasks are relatively simpler, making it easier for the model to capture their features.
- In the accuracy graph on the right, the training accuracy improves progressively, corresponding to the decrease in loss. The final training accuracy indicates that the model fits the training data well. However, this does not fully guarantee generalization performance on the validation set.

*Validation Accuracy (Accuracy of validation [24])*

- Task 1 : The initial model (Version 1) achieved a validation accuracy of approximately 85% on the Task 1 test set. However, after training on Task 2 and Task 3, the validation accuracy on the Task 1 test set gradually decreased. By Version 3, the validation accuracy on Task 1 dropped to approximately 50%, showing a significant reduction and demonstrating a clear case of catastrophic forgetting.
- Task 2 : In Version 2, the validation accuracy on the Task 2 test set was approximately 95%. However, in Version 3, the validation accuracy on the Task 2 test set dropped to approximately 60%. After training on Task 3, the validation accuracy for Task 2 also decreased significantly, indicating the presence of catastrophic forgetting.
- Task 3 : After completing training on Task 3 in Version 3, the validation accuracy on the Task 3 test set was approximately 96%. Task 3, being the most recently trained task, performed the best, showing that the model has strong adaptability to the most recent task in the continual learning process.

**Phenomenon Analysis**

Catastrophic Forgetting

- Definition: Catastrophic forgetting refers to the phenomenon in continual learning where the model, while learning new tasks, overwrites or loses the features and knowledge from previous tasks.
- Manifestation: The validation accuracy on the Task 1 and Task 2 test sets significantly decreased as the model trained on newer tasks.

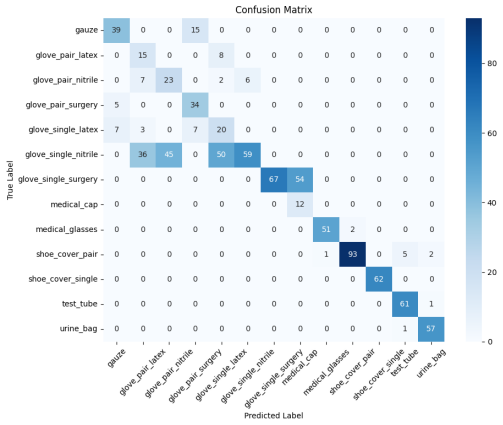



Figure 25: Confusion Matrix for Continual Learning

**Confusion Matrix Analysis**

From the confusion matrix, it can be observed that the model demonstrates overall excellent classification performance. Most of the classification results are concentrated along the diagonal, indicating that the model can accurately identify these categories. The classification results for shoe_cover_pair and urine_bag are particularly outstanding, with almost no apparent misclassifications. This highlights the distinctiveness of these categories' features and the model's strong ability to recognize them.

However, there is still some degree of confusion between certain categories, especially among glove-related categories. For instance, glove_single_latex and glove_single_nitrile are frequently misclassified as each other, possibly due to the high visual similarity in color and material between single gloves, making them difficult to distinguish. Additionally, some instances of glove_pair_surgery are misclassified as glove_pair_nitrile, which may reflect overlapping features between paired gloves during feature extraction.

In contrast, other categories show more stable classification performance, such as medical_glasses and test_tube, with minimal classification errors. This suggests that these categories have more distinct features, enabling the model to effectively capture their key characteristics.

Overall, in model Version 3, the validation performance on the test set for Task 1, which is related to glove categories, was the worst, followed by Task 2, while Task 3 showed the best performance. This aligns with the characteristics of Catastrophic Forgetting (Li et al. (2024)), where the model forgets the features and knowledge of previously trained tasks as it learns new categories.

### 4.6 Experiment 5 : Continual Learning through Clustering-Based Task Classification

This experiment utilizes the sequence of learning tasks obtained in Section 3.4.2 i.e., using hierarchical clustering to generate four tasks and determines the task transfer order for continual learning using the LEEP score. The model gradually introduces data according to the task sequence, starting with training on Task $t_1$. Once the desired performance is achieved, data from Tasks $t_2$, $t_3$, and $t_4$ are sequentially introduced. At each stage, the model's accuracy and loss values on the training and test sets are recorded to evaluate whether the model retains knowledge from previous tasks or experiences catastrophic forgetting.

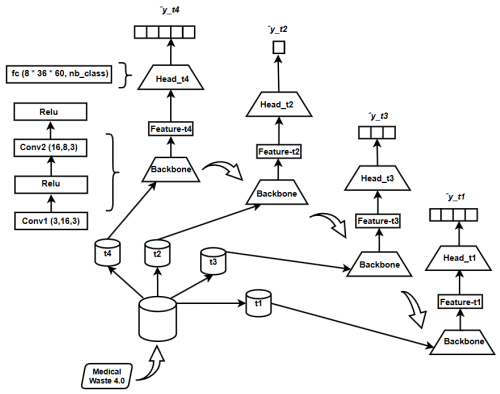



Figure 26: Continual Learning Model Architecture-V2

Figure 26 illustrates the architecture of the continual learning model. The input data (Medical Waste 4.0) is divided into multiple tasks ($t\_1, t\_2, t\_3, t\_4$), each of which passes through a shared backbone network to extract task-specific features ($Feature\_t1, Feature\_t2, Feature\_t3, Feature\_t4$). These features are then processed by corresponding task-specific heads (Head) for classification or prediction. The backbone network comprises convolutional layers (Conv1, Conv2), activation functions (ReLU), and fully connected layers (fc) to progressively process the input features. Through this structure, the model shares feature extraction capabilities across tasks while maintaining the independence of task heads, effectively enabling continual learning and reducing the risk of catastrophic forgetting.

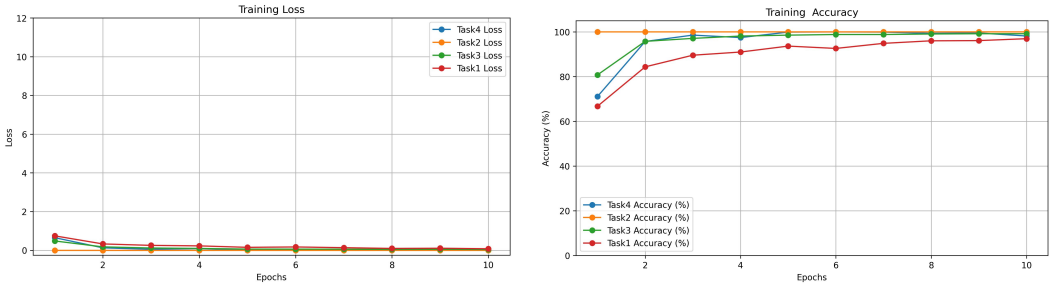
### 4.6.1 Experiment with Transform Resolution

| **Hyper-parameter** | **Value** |
|---|---|
| Number of Neurons in Conv1 | $16 \times 38 \times 62 = 37{,}696$ |
| Number of Neurons in Conv2 | $32 \times 36 \times 60 = 69{,}120$ |
| Number of Neurons in FC_T4 | 5 |
| Number of Neurons in FC_T2 | 1 |
| Number of Neurons in FC_T3 | 3 |
| Number of Neurons in FC_T1 | 4 |

Table 7: Hyper-parameter of Continual Learning Model-V2 transform

The hyper-parameter values used in the experiment are shown in the table 7.

### 4.6.2 Analysis of Model Training Methods under Transform Resolution




(a) Loss-Continual V2

(b) Accuracy-Continual V2

Figure 27: Training Loss and Accuracy on Transform Resolution

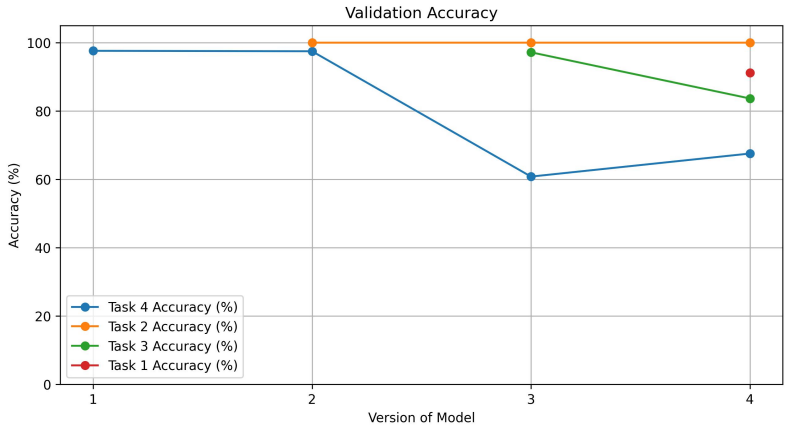



Figure 28: Validation Accuracy on Transform Resolution

**Accuracy and Loss Rate Analysis**

*Training Loss and Accuracy (Loss and Accuracy of Training under Transform Resolution [27])*

Figure 27 presents the training loss (a) and training accuracy (b) across 10 epochs for four tasks (Task 1 to Task 4) under the Transform Resolution setting. Subfigure (a) shows the gradual decrease in training loss for all tasks, with convergence near zero after the initial epochs. Subfigure (b) highlights the steady increase in training accuracy, with all tasks achieving near-perfect accuracy by the end of training.

The results demonstrate that the model effectively learns all tasks under the Transform Resolution setting. The rapid convergence of training loss in subfigure (a) suggests that the tasks are relatively easy for the model to optimize, likely due to the preprocessing and resolution transformation applied. Subfigure (b) shows that Task 2 reaches near 100% accuracy almost immediately, reflecting its simplicity as a single-class task. Tasks 1, 3, and 4 also achieve high accuracy after a few epochs, indicating the model's capability to generalize across different tasks. The minimal variance in loss and accuracy across tasks suggests consistent model performance, though the simplicity of some tasks (e.g., Task 2) could contribute to these results.

*Validation Accuracy (Validation Accuracy on Transform Resolution [28])*

Figure 28 presents the validation accuracy for four tasks (Task 1 to Task 4) across different versions of the model under the Transform Resolution setting. The results show variations in accuracy among tasks, with Task 2 maintaining a consistent 100% accuracy across all model versions. In contrast, the other tasks exhibit fluctuations in accuracy, particularly Task 4, which experiences a notable drop in version 3 before partially recovering in version 4.

- Task 4 : The accuracy of Task 4 drops significantly from version 2 to version 3, which aligns with the catastrophic forgetting effect often observed in continual learning, where performance on earlier tasks deteriorates as new tasks are introduced. However, Task 4's accuracy improves in version 4, likely due to the shared feature representations learned during the training of Task 1 and Task 3, which contributed to the recovery of Task 4's performance.

- Task 2 : Task 2 maintains a consistent 100% validation accuracy across all model versions. This is because Task 2 consists of only a single class, making it inherently robust to forgetting since the model can always predict correctly regardless of further training.

- Task 3 : Task 3's accuracy peaks in version 3 and slightly declines in version 4. This decline reflects the typical forgetting behavior in continual learning, where earlier tasks lose priority as the model focuses on new tasks.

- Task 1 : Task 1 shows a slight decrease in accuracy as the model progresses to later versions, consistent with the order of task training. This decline indicates that earlier tasks are gradually deprioritized as new tasks are added to the model.

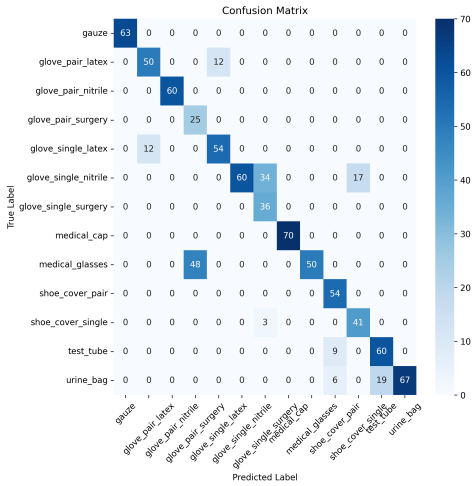



Figure 29: Confusion Matrix for Continual Learning-based Hierarchical Clustering under Transform Resolution

*Confusion Matrix (Confusion Matrix for Continual Learning-based Hierarchical Clustering under Transform Resolution [29])*

- The confusion matrix shows a total of 160 incorrect predictions and 690 correct predictions, with an overall sample size of 850. The model achieved an overall accuracy of 81.17%, indicating good classification performance.

- However, the errors are mainly concentrated in cases such as glove_single_nitrile being misclassified as glove_single_surgery, medical_glasses being misclassified as glove_pair_surgery, and urine_bag being misclassified as test_tube.

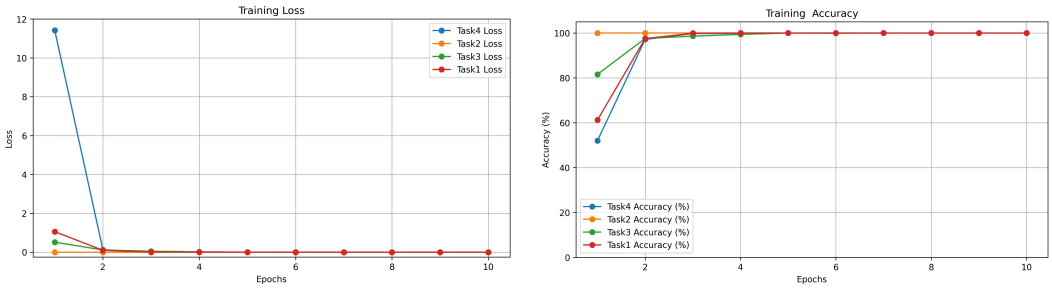
### 4.6.3 Experiment with Full Resolution

| **Hyper-parameter** | **Value** |
|---|---|
| Number of Neurons in Conv1 | $16 \times 398 \times 638 = 4062784$ |
| Number of Neurons in Conv2 | $8 \times 396 \times 636 = 2014848$ |
| Number of Neurons in FC_T4 | 5 |
| Number of Neurons in FC_T2 | 1 |
| Number of Neurons in FC_T3 | 3 |
| Number of Neurons in FC_T1 | 4 |

Table 8: Hyper-parameter of model continual-V2 full

The hyper-parameter values used in the experiment are shown in the table 8.

#### 4.6.4 ANALYSIS OF MODEL TRAINING METHODS UNDER FULL RESOLUTION




(a) Loss-Continual V2 (b) Accuracy-Continual V2

Figure 30: Training Loss and Accuracy on Full Resolution

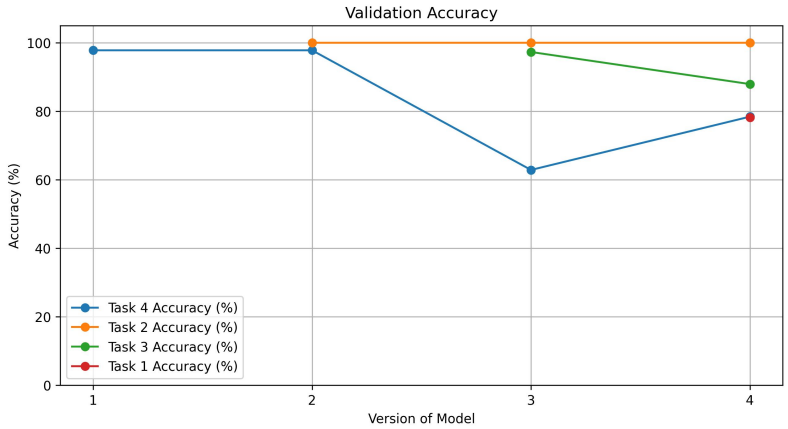



Figure 31: Validation Accuracy on Full Resolution

**Accuracy and Loss Analysis**

*Training Loss and Accuracy (Training Loss and Accuracy on Full Resolution [30])*

The left graph shows the change in loss values for the four tasks during training over 10 epochs. The model achieves rapid convergence within the first 3 epochs, with the training loss dropping close to 0 and accuracy nearing 100%, demonstrating the model's high learning efficiency.

- Task 4 has an initial loss value significantly higher than the other tasks (close to 12), likely due to its more complex data distribution or a larger number of categories. However, through training, the model is able to quickly converge on the features of $t\_4$.
- Tasks (Task1, Task2, Task3) have lower initial loss values and faster convergence rates, indicating that these tasks are relatively simpler and easier for the model to learn.

The right graph shows the change in training accuracy for each task over the epochs. Corresponding to the loss curves, the accuracy of all tasks increases rapidly within the first 5 epochs and approaches 100% by the end of training.

- The training accuracies of different tasks converge to similar levels, indicating that the model adapts well to the incremental addition of tasks during the continual learning process and maintains a high learning capability.
- However, although the training accuracy approaches 100%, the performance on training data alone is insufficient to fully evaluate the model's generalization capabilities. It is

necessary to analyze results on the validation or test set to determine whether catastrophic forgetting has occurred.

*Validation Accuracy (Validation Accuracy on Full Resolution [31])*

- The accuracy decreased from 98%, indicating that the model's performance on Task 4 deteriorated as new tasks were trained. This is because the model partially lost its memory of previous tasks while learning new ones. In Version 2, the accuracy showed only a slight decrease because Task 2 contains only a single class, causing minimal interference with the model's memory of Task 4. However, in Version 3, the accuracy dropped significantly, suggesting that the training of Task 3 caused greater interference with the model's memory of Task 4. In Version 4, the accuracy increased, indicating that the classes in Task 1 may share similar features with those in Task 4.
- Task 2 : The accuracy remained constant at 100%. This is because the data distribution for Task 2 is very simple, with only a single class. The predictions always match the labels, resulting in consistently perfect accuracy.
- Task 3 : The accuracy was maintained at approximately 96%, showing overall strong performance. Even after training on a new task (Task 1), the accuracy only showed a slight decrease and stabilized at around 90%. This indicates that the model retained its knowledge of Task 3 relatively well during short-term continual learning.
- Task 1 : After training on Task 1, the accuracy on its test set stabilized at approximately 80%.

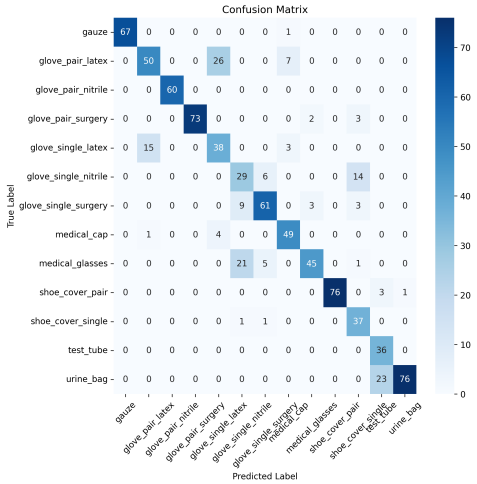



Figure 32: Confusion Matrix for Continual Learning-based Hierarchical Clustering under Full Resolution

*Confusion Matrix (Confusion Matrix for Continual Learning-based Hierarchical Clustering under Full Resolution [32])*

- The confusion matrix shows a total of 153 incorrect predictions and 697 correct predictions, with an overall sample size of 850. The model achieved an overall accuracy of 81.88%, indicating good classification performance.
- However, the errors are mainly concentrated in cases such as glove_pair_latex being misclassified as glove_pair_surgery, medical_glasses being misclassified as glove_single_latex, and urine_bag being misclassified as test_tube. These errors are related to interference from previously trained tasks, which likely reduced the model's ability to distinguish between certain categories.

#### 4.6.5 Analysis of Validation Performance Across Different Random Seeds

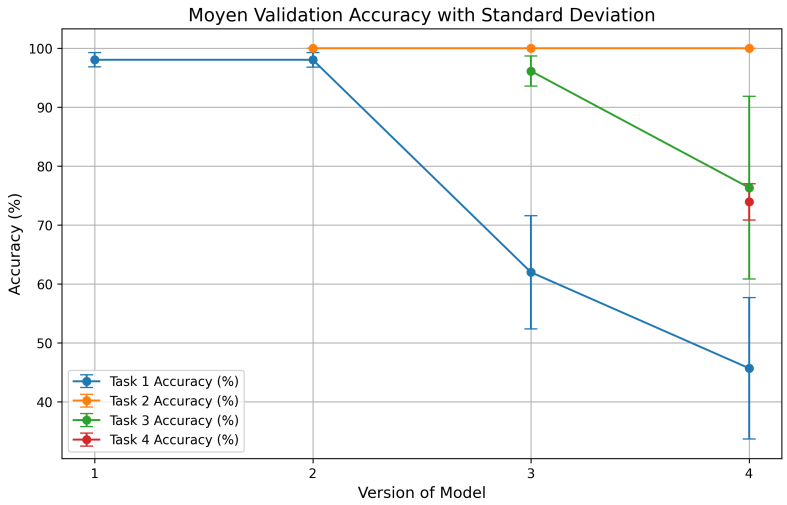



Figure 33: Average Training Accuracy on Full Resolution

To analyze the model's performance stability and consistency under different random initialization conditions, we conducted experiments using five different random seeds (seed0–seed4). For each random seed, we recorded the validation accuracy for all tasks and calculated the average values across the five runs. The results are summarized in the figures above.

The figure 33 presents the average validation accuracy and its standard deviation for each task across different model versions. It can be observed that the accuracy of Task 1 and Task 4 drops significantly in model version 3, particularly for Task 4, where the accuracy plummets from nearly 80% in model version 2 to below 50% in version 3. This trend highlights the poor performance stability of model version 3 on these tasks, with Task 4 being especially problematic.

In contrast, the validation accuracy of Task 2 remains consistently at 100% across all model versions. This phenomenon reflects the "pseudo-stability" of Task 2, as it contains only a single class, requiring no actual learning to achieve high accuracy. Meanwhile, Task 3 shows slight fluctuations in accuracy but generally maintains a high performance level.

These findings further underscore the potential performance variability in multi-task learning systems when faced with differences in task complexity and class distribution. Moving forward, we aim to optimize the model design to enhance overall performance stability and consistency, ensuring better adaptability to the unique requirements of each task.

## 5 Discussion

We used the **Flatten** method to train and test all training data at once. From the confusion matrix 13, it can be observed that the model performed well in classification, with no significant errors, achieving an overall accuracy of approximately **90%**. Based on this, we introduced the **Independent Task** method, which reduces the number of categories in the classifier layer to further optimize the model's performance. The confusion matrix 18 shows that the overall accuracy improved to **93%**. Subsequently, we used the **MTL** method to enhance the Independent Task approach. The model utilized a shared backbone and sequentially introduced a batch of task data during each iteration of training, significantly improving the accuracy to **97%** 21.

To simulate the training and testing performance of the model on new task data, we conducted experiments using the **Continual Learning** method 25. By measuring the semantic distances between categories, we grouped 13 categories into 3 tasks. The confusion matrix revealed that during continual learning, the model experienced forgetting of previously trained tasks and interference between tasks, resulting in a decline in overall accuracy to **69.7%**.

In **Experiment 5**, based on this, we further investigated more reasonable classification methods, the impact of training task order on performance, and the effect of dataset resolution on model performance. Using two distance metrics, Euclidean and task2vec, we calculated the distances between categories and performed hierarchical clustering to determine the categories included in each task. By comparing leep scores, we identified the optimal training sequence.

we tested datasets with different resolutions. For the **Transform Resolution** dataset, the confusion matrix 29 showed an accuracy of **81.17%**, while for the **Full Resolution** dataset, the confusion matrix 32 showed an accuracy of **81.88%**. However, when calculating the average accuracy with fixed random seeds (0 to 4), the model's overall performance was found to be suboptimal. This could be attributed to an increase in the number of neurons per layer in the model, which exacerbated the forgetting of previously trained tasks. Future research will focus on this issue to explore more effective optimization methods.

## 6 CONCLUSION

This study conducted five progressive experiments: *Full-Batch Training*, *Independent Task Training*, *Multi-Task Learning*, *Continual Learning*, and *Continual Learning through Clustering-Based Task Classification*, to validate the feasibility of a **Never-Ending Learning Model** for medical waste classification. The experimental results demonstrated that the continual learning approach, by leveraging clustering-based task classification and determining optimal task training sequences, significantly enhanced the model's performance in dynamic environments and ensured stable classification performance. These findings provide strong support for the realization of the Never-Ending Learning Model. Furthermore, the experiments confirmed that the integration of image analysis techniques and continual learning methods not only enables dynamic adaptation to new categories but also optimizes classification efficiency, addressing the complex and evolving demands of medical waste management.

The results obtained at this stage indicate that the **Never-Ending Learning Model** is feasible in the context of medical waste classification. This provides a solid foundation for future research to further optimize the model and expand its application scenarios while offering reliable technical support for dynamic adaptation and long-term management in complex medical waste classification settings.

## 7 FUTURE WORK

By implementing multiple training sessions with result averaging, reevaluating dataset task separation, optimizing training parameters, expanding the dataset scope and mitigating catastrophic forgetting, the stability, generalization, and adaptability of the continual learning model can be significantly enhanced. These improvements offer effective solutions to address bias issues and task adaptability in continual learning.

**Multiple Training and Result Averaging**

Due to the potential bias in prediction results on the same test set after training the model on the same training set, multiple rounds of training can be performed to improve the stability of the results. After each training session, the prediction results are recorded, and the average of the results is taken as the final evaluation metric for model performance. Additionally, the variance of the results can be calculated to quantify the bias range of predictions, providing a more accurate assessment of model performance. This approach effectively reduces fluctuations caused by the randomness of single training sessions.

**Optimizing Model Training Parameters**

During the model training process, optimization tools (such as grid search, random search, or Bayesian optimization) can be used to systematically optimize hyperparameters and training parameters (e.g., nb epoch) to enhance training efficiency and validation performance. Furthermore, different learning rates can be set for each task to adjust the training pace according to task complexity. For instance, smaller learning rates can be used for complex tasks to stabilize gradient updates, while larger learning rates can accelerate convergence for simpler tasks. Such optimization

allows the model to learn task-specific features more efficiently, thereby further improving overall performance.

**Expanding Dataset Scope**

To enhance the inclusivity of the continual learning model, new datasets can be introduced for training. These new datasets can help the model learn more diverse features and improve its adaptability in multi-task and multi-category environments. By training on richer data, the model can better handle more complex scenarios and simultaneously validate its generalization ability in diverse task environments. This expansion not only contributes to improving model performance but also provides stronger foundational support for continual learning.

**Mitigating Catastrophic Forgetting**

To mitigate catastrophic forgetting and improve model testing performance, methods such as Elastic Weight Consolidation (EWC) can be employed. EWC identifies and protects important parameters of previous tasks using the Fisher Information Matrix, restricting their excessive updates during the training of new tasks. This approach effectively preserves the features of previous tasks, reduces forgetting, balances new and old tasks, and enhances the model's overall performance and validation results.

## REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 6430–6439, 2019.

A. Bruno, C. Caudai, G. R. Leone, M. Martinelli, D. Moroni, and F. Crotti. Medical waste sorting: a computer vision approach for assisted primary sorting, 2023a. URL https://arxiv.org/abs/2303.04720.

Antonio Bruno, Massimo Martinelli, and Davide Moroni. Medical-waste-4.0-dataset: v0.1, February 2023b. URL https://doi.org/10.5281/zenodo.7643417.

Arslan Chaudhry, Albert Gordo, Puneet K. Dokania, Philip H. S. Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. CoRR, abs/2002.08165, 2020. URL https://arxiv.org/abs/2002.08165.

Yayong Li, Peyman Moghadam, Can Peng, Nan Ye, and Piotr Koniusz. Inductive graph few-shot class incremental learning, 2024. URL https://arxiv.org/abs/2411.06634.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In International Conference on Machine Learning, pp. 7294–7305. PMLR, 2020.

World Health Organization. Healthcare waste. WHO News Report, 2024. URL https://www.who.int/news-room/fact-sheets/detail/health-care-waste.

Haiying Zhou, Xiangyu Yu, Ahmad Alhaskawi, Yanzhao Dong, Zewei Wang, Qianjun Jin, Xianliang Hu, Zongyu Liu, Vishnu Goutham Kota, Mohamed Abdulla, Sohaib Ezzi, Binjie Qi, Juan Li, Bixian Wang, Jianyong Fang, and Hui lu. A deep learning approach for medical waste classification. Scientific Reports, 12, 02 2022. doi: 10.1038/s41598-022-06146-2.

# A APPENDIX

## A.1 DATASET VISUALIZATION

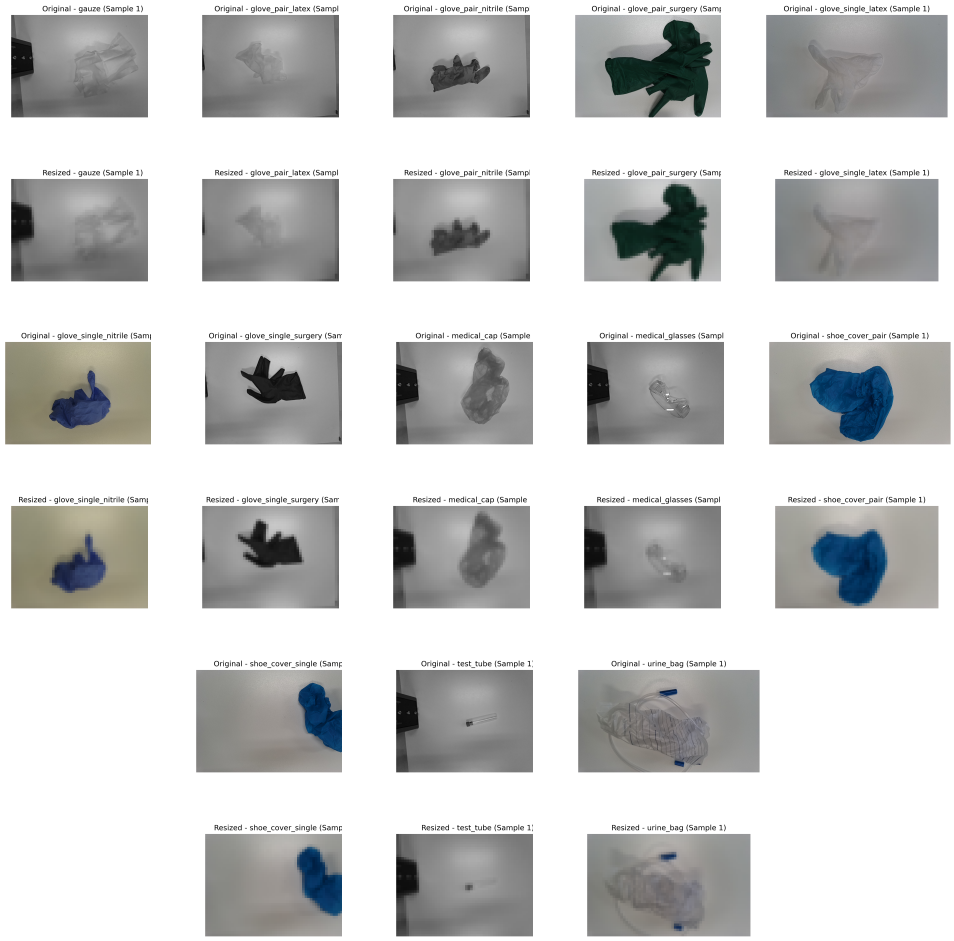



Figure 34: Original and Resize Dataset sample

The Resize operation adjusts an image to a specified target height and width, here we adjust images to (40, 64), by processing the original pixel matrix. For each target pixel, its corresponding position in the original image is calculated based on the scaling ratio. The pixel value is then determined using bilinear interpolation, a method that considers the four nearest pixels in the original image. This interpolation calculates the target pixel's value as a weighted average of these four neighbors, with weights based on their distances to the target position. This process effectively produces a resized image matrix with the desired dimensions, preserving the overall structure and quality of the original image while ensuring smooth transitions between pixels.

By selecting the first image from each category and comparing its original version with the one processed by the Resize operation, we can visually observe the overall reduction in pixel count and the change in dimensions. To facilitate comparison, the resized images are displayed at the same scale as the original images, allowing for a clearer examination of the details in both. The comparison reveals that the resized images exhibit a certain degree of blurriness due to the reduction in pixel count, resulting in a loss of details and slightly lower clarity compared to the original images. This blurriness is an unavoidable consequence of downscaling but, within the target dimensions suitable for the task, key features are typically preserved.

The input image is converted from a PIL image or NumPy array format to a PyTorch Tensor format, with pixel values normalized from the range 0-255 to 0-1. Additionally, the image data is standardized to have a mean of 0.5 and a standard deviation of 0.5, enhancing the stability and performance of model training.