

Towards a Never-ending Learning Model of Recyclable Medical Waste

Supervisé par Monsieur Massinissa Hamidi

FENG Jiaqi

SHAO Qichen

Plan

01 Problématique

**02 Setting de Continual
Learning**

03 Génération des Tâches

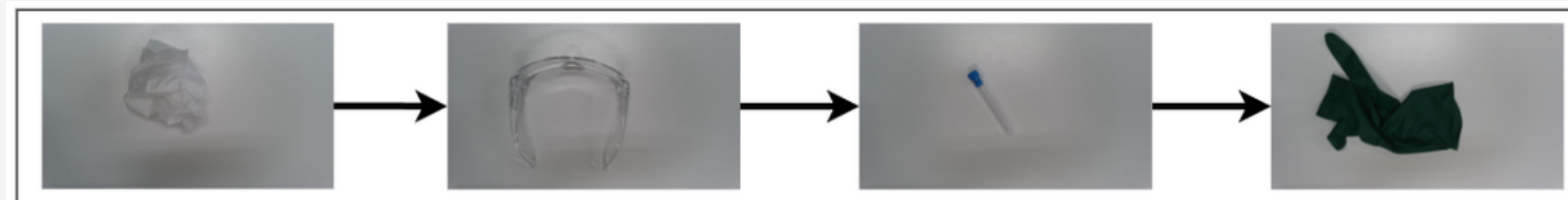
04 Expériences

05 Discussion

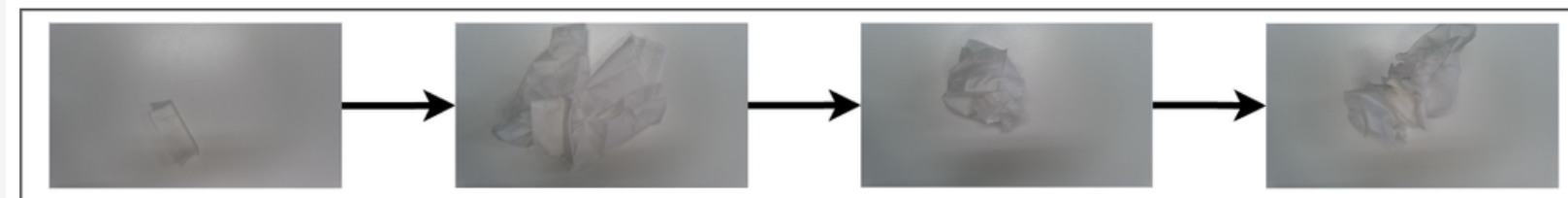
06 Travaux Futurs

Problématique

- La gestion des déchets médicaux constitue un défi mondial majeur pour le secteur de la santé. Selon les statistiques de l'WHO en 2024, environ 15 % des déchets médicaux sont infectieux, toxiques ou radioactifs, ce qui complique considérablement leur gestion, en particulier lors de la phase de classification.
- Une mauvaise gestion de ces déchets dangereux représente non seulement une menace directe pour la santé humaine, mais également pour l'environnement, causant des dommages écologiques graves et aggravant la propagation des maladies..
- Nouveaux types de déchets, changements d'éclairage, changements de résolution de caméra ...



Scenario 1 : Category variation



Scenario 2 : Form variation



Setting de Continual Learning

Definition

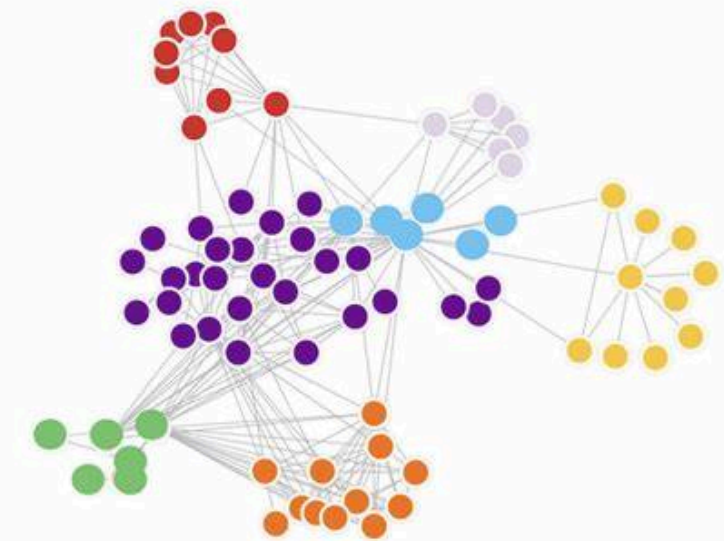
Un modèle d'apprentissage continu est capable d'accumuler des connaissances de manière continue dans un environnement dynamique et multitâche, tout en s'adaptant à de nouvelles tâches et en évitant l'oubli catastrophique.

Accumulation et transfert des connaissances

Un modèle d'apprentissage continu est capable d'accumuler des connaissances de manière continue dans un environnement dynamique et multitâche, tout en s'adaptant à de nouvelles tâches et en évitant l'oubli catastrophique.

Adaptabilité

Il s'adapte à de nouveaux environnements ou à des changements dans la distribution des données, en soutenant un apprentissage dynamique sans avoir besoin de réentraîner complètement le système.



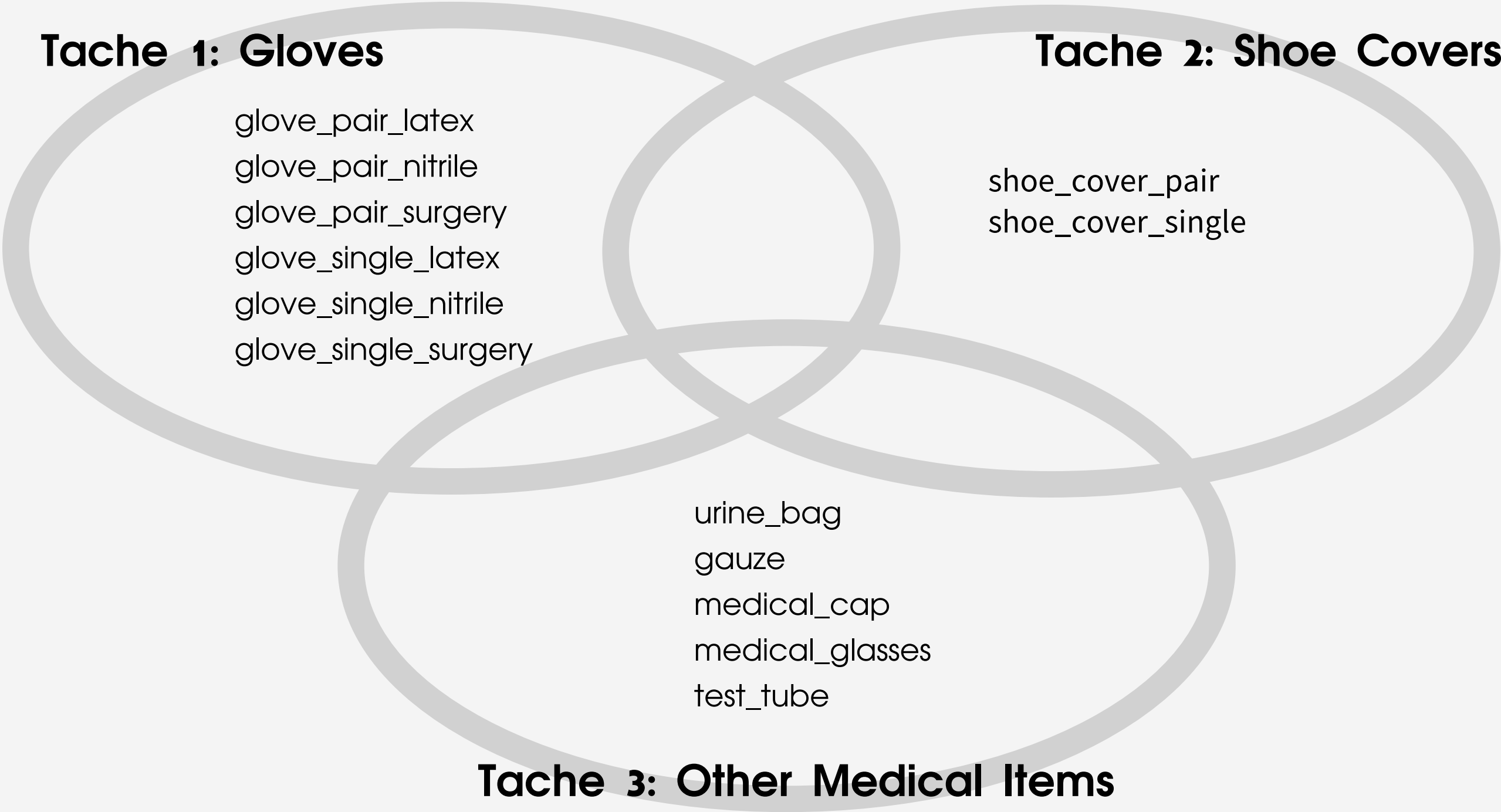
Classes -> Tache

Génération des Tâches

01 Génération basée sur Sementique

02 Génération basé sur Clustering Hiérarchique

Génération basée sur Sementique



Génération basée sur Clustering Hiérarchique

01

Distance Euclidean

Nous calculons la distance euclidienne entre les points centraux de chaque classe.

02

Distance Task2Vec

Nous représentons les classes sous forme de vecteurs d'embedding dans un espace de caractéristiques, puis nous calculons la distance cosinus entre ces vecteurs.

03

Hiérarchique Clustering

En utilisant les deux types de distances inter-classes obtenues précédemment, nous regroupons récursivement les classes pour finalement former des groupes de tâches.

04

Leep Scores

Nous évaluons la capacité de transfert du modèle pré-entraîné vers une tâche cible en fonction de la distribution des prédictions et des étiquettes de la tâche cible, afin de déterminer un meilleur ordre de transfert entre les tâches.

Dataset

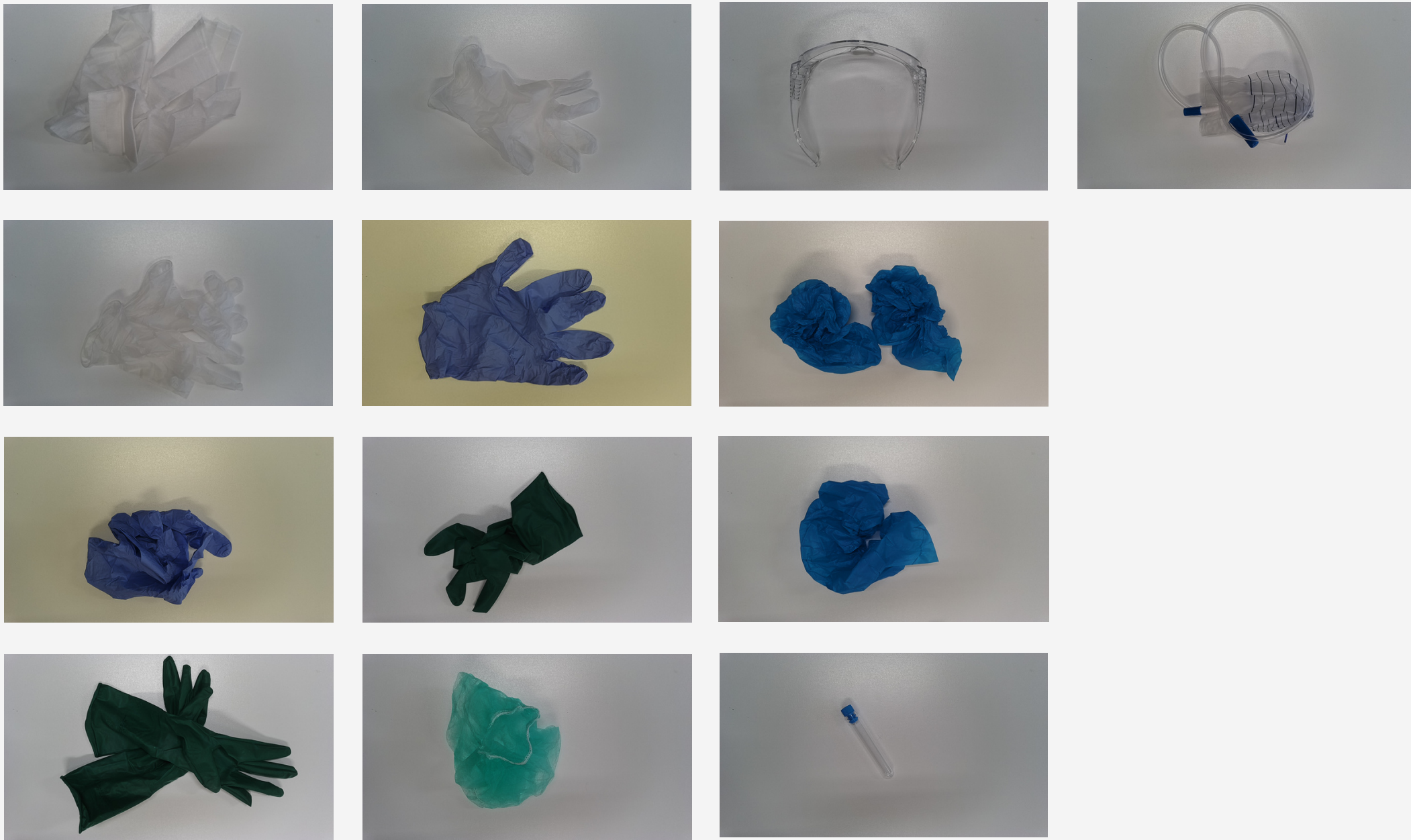
Medical-Waste-4.0-Dataset

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

Distribution du dataset

80% pour l'entraînement

20% pour la validation



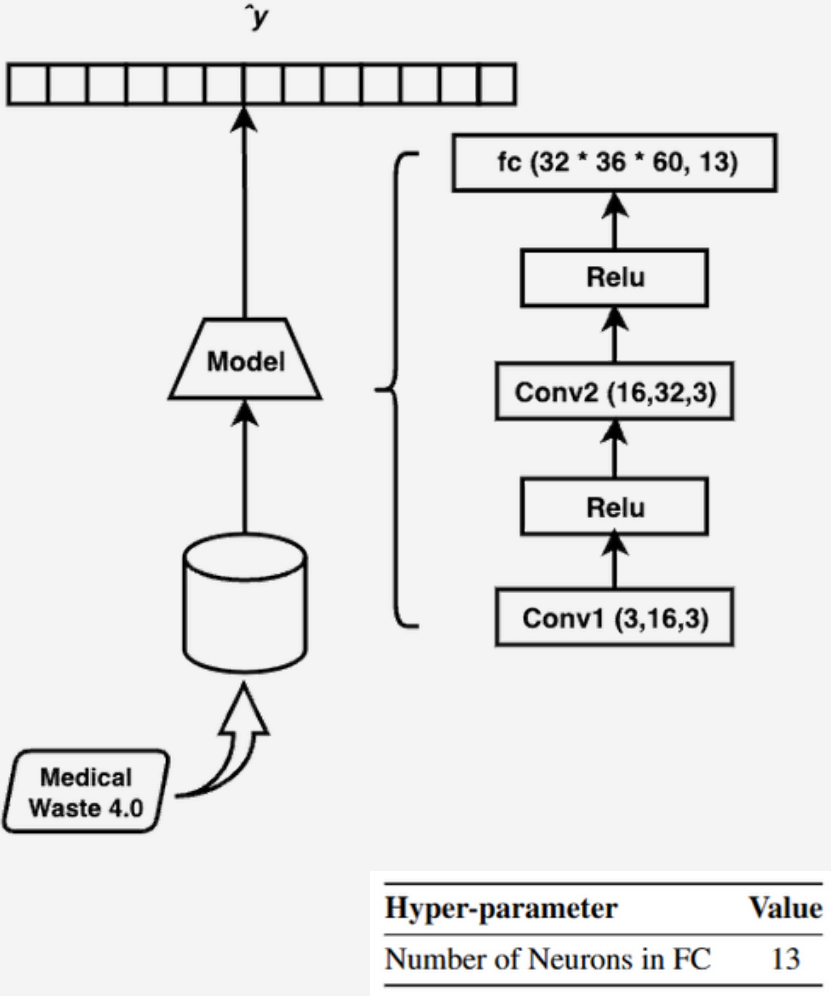
Architecture et Hyperparamètre

Baseline

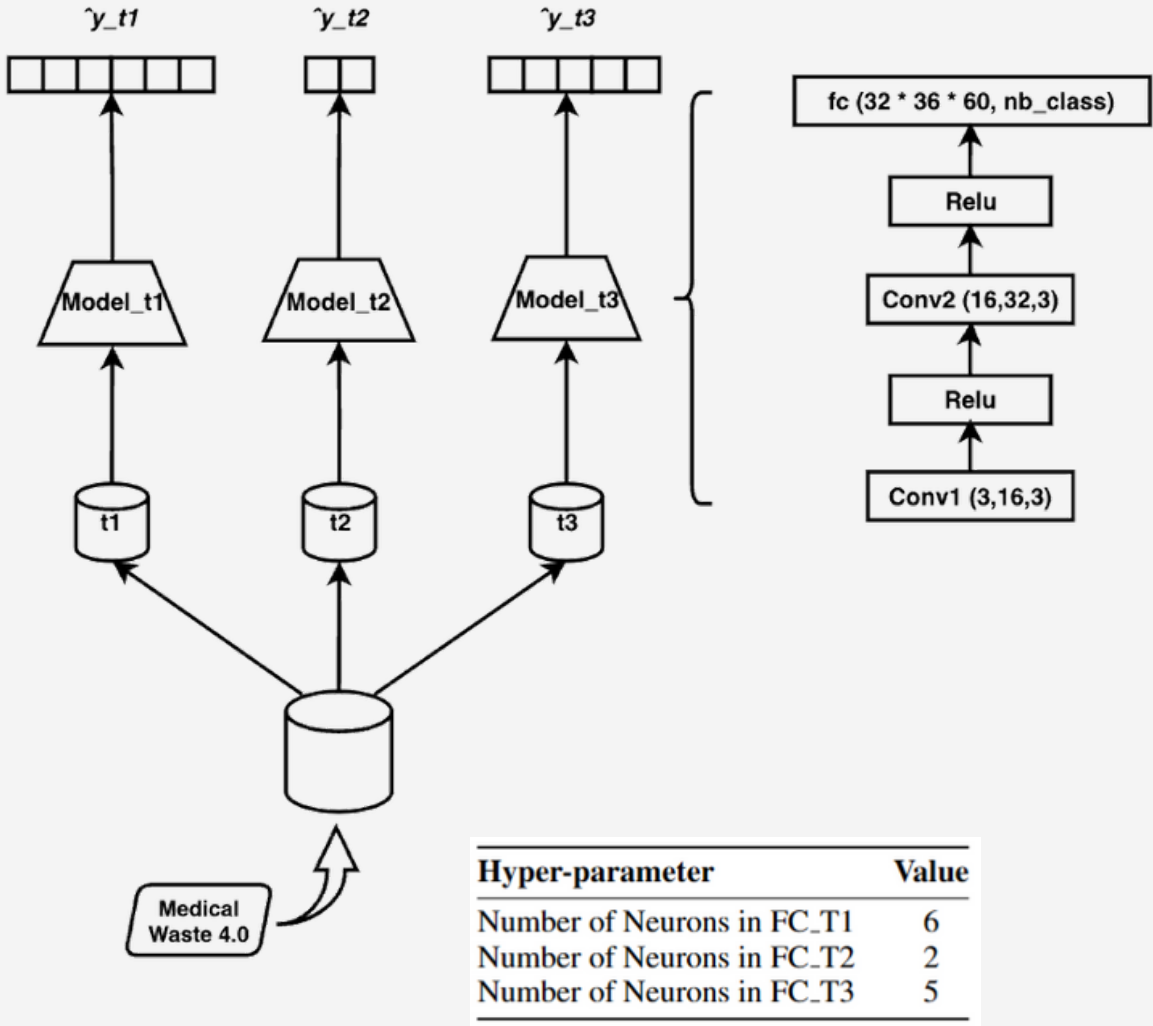
Hyper-parameter	Value
Validation split	0.2
Number of layers	3
Number of Neurons in Conv1	$16 \times 38 \times 62 = 37,696$
Number of Neurons in Conv2	$32 \times 36 \times 60 = 69,120$
Activation function	ReLU
Dropout	Not used
Learning rate	0.001
Optimizer	Adam
Batch size	16

Hyper-paramètre en commun

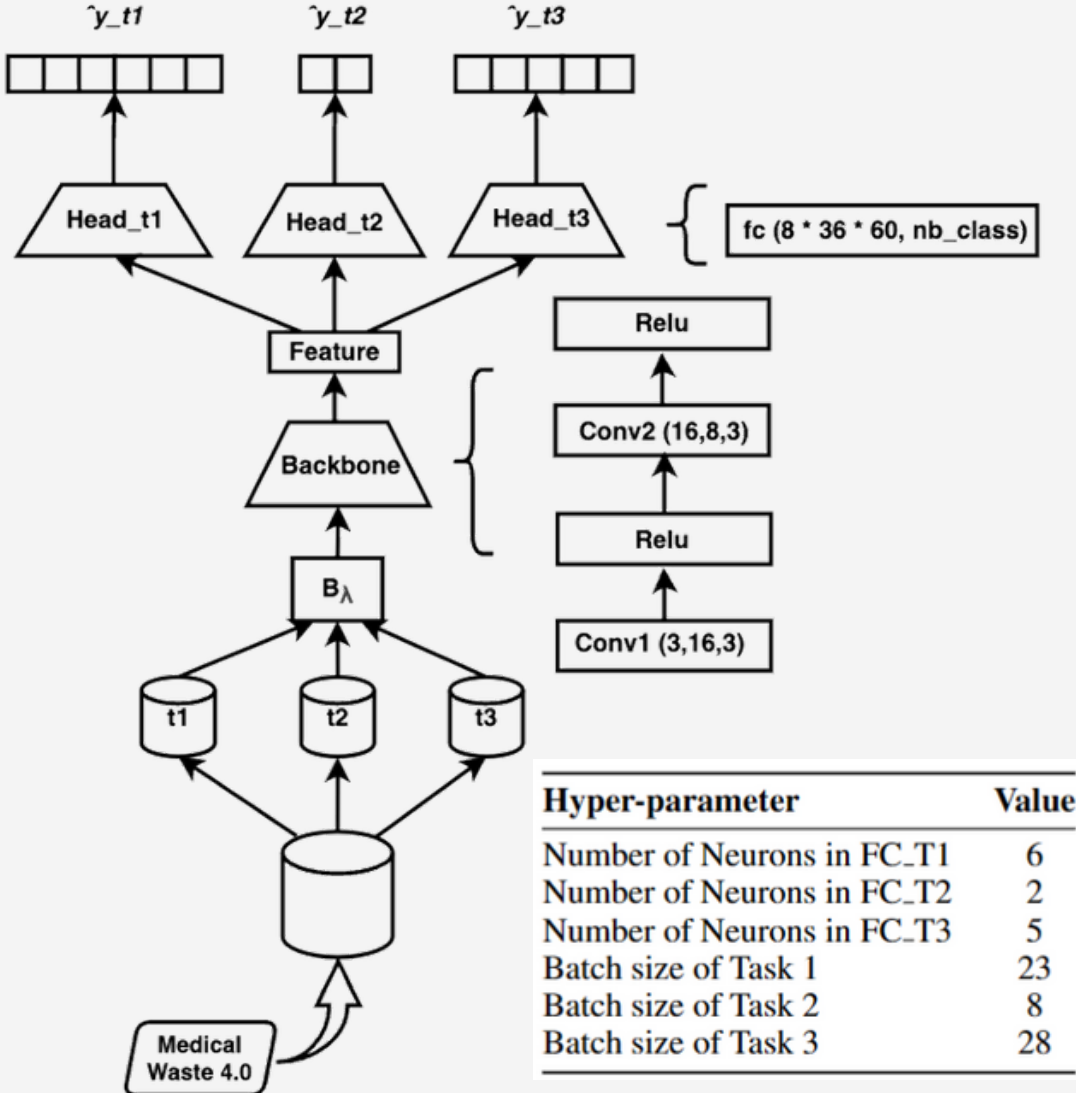
Modèle flatten



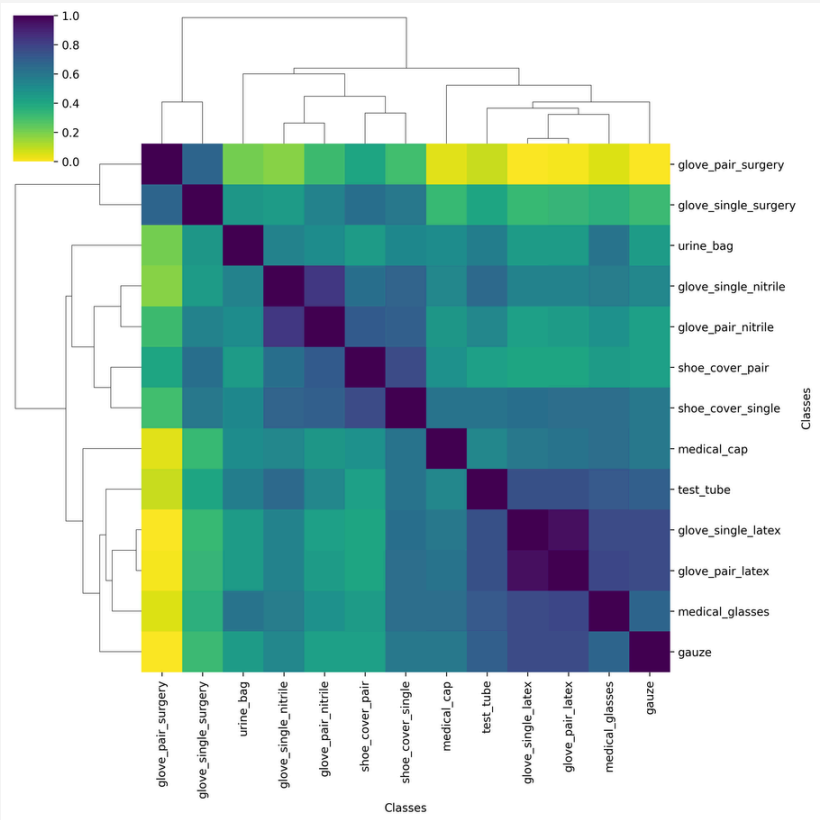
Modèle spécifique



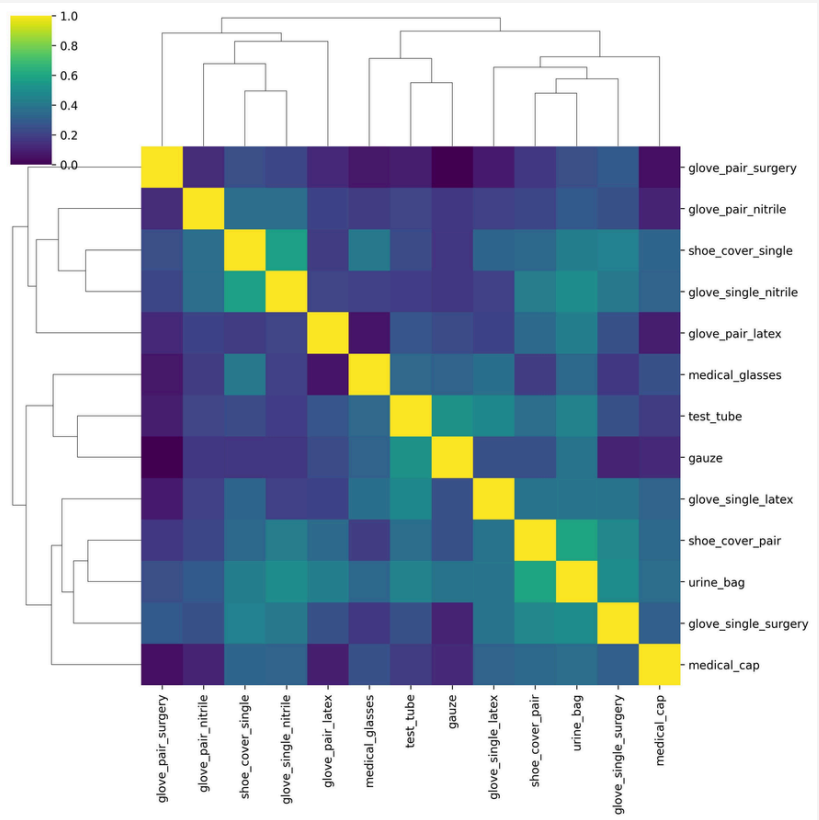
Modèle multi-task-learning



Génération de Tâches et de l'ordre



Matrice de Similarité (Euclidean)



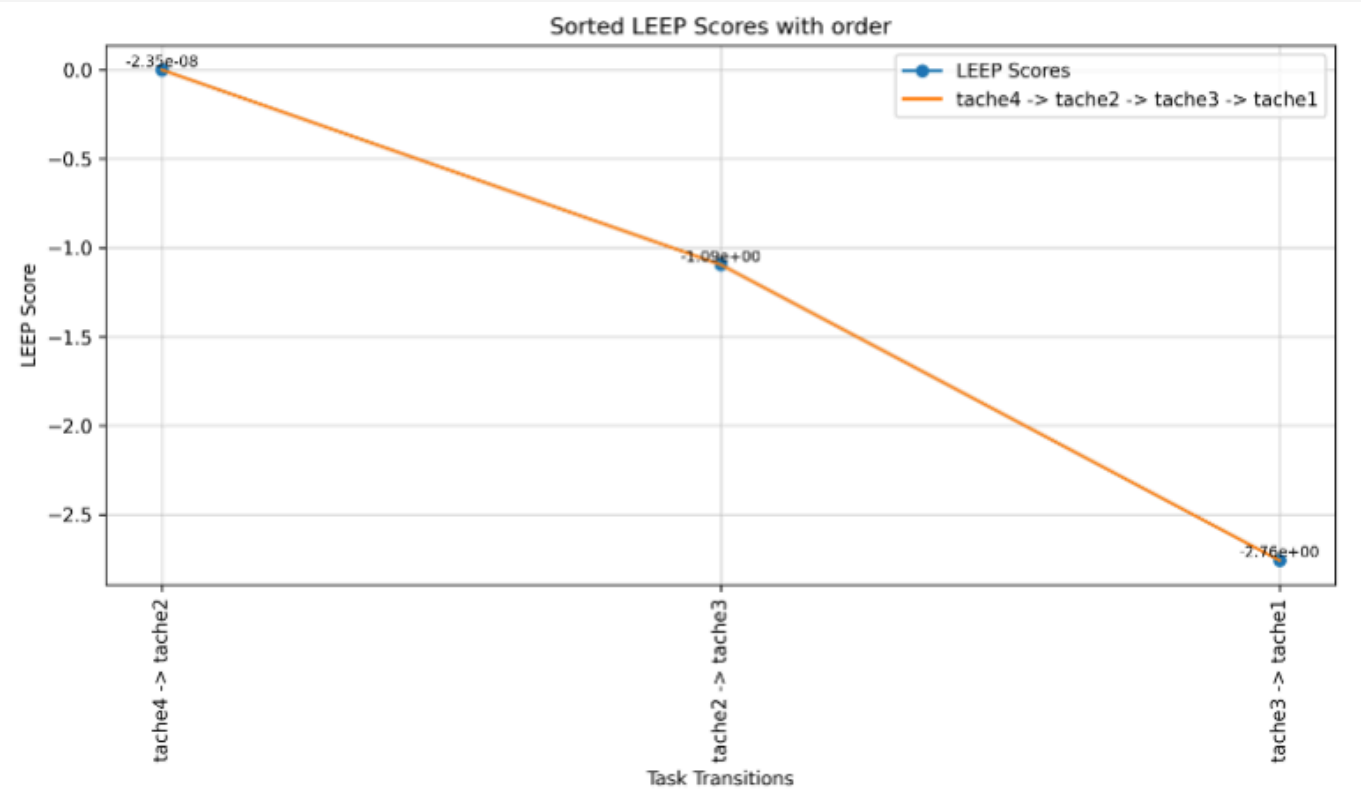
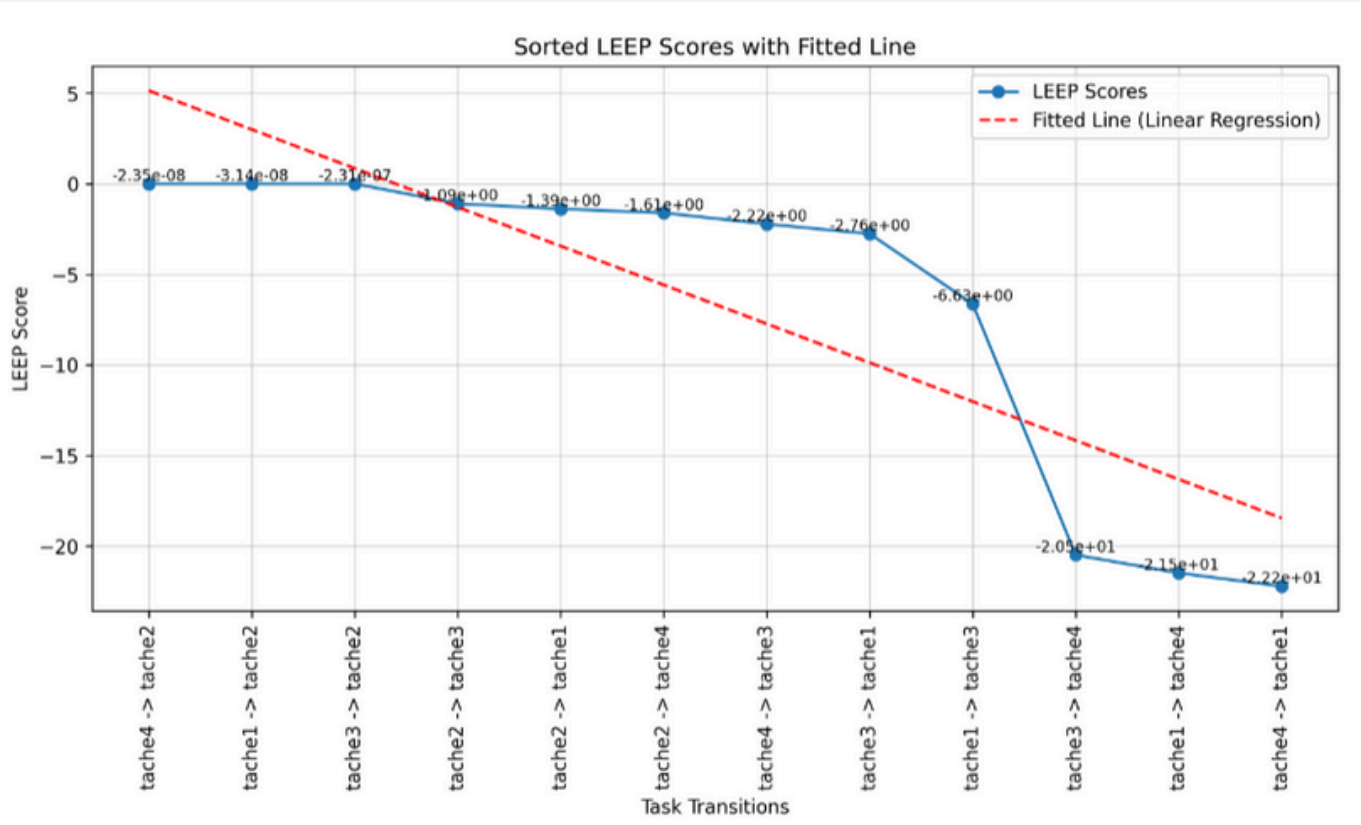
Matrice de Similarité (Task2Vec)

Tâche 4
glove_single_latex
glove_single_surgery
shoe_cover_pair
urine_bag
medical_cap

Tâche 2
glove_pair_surgery

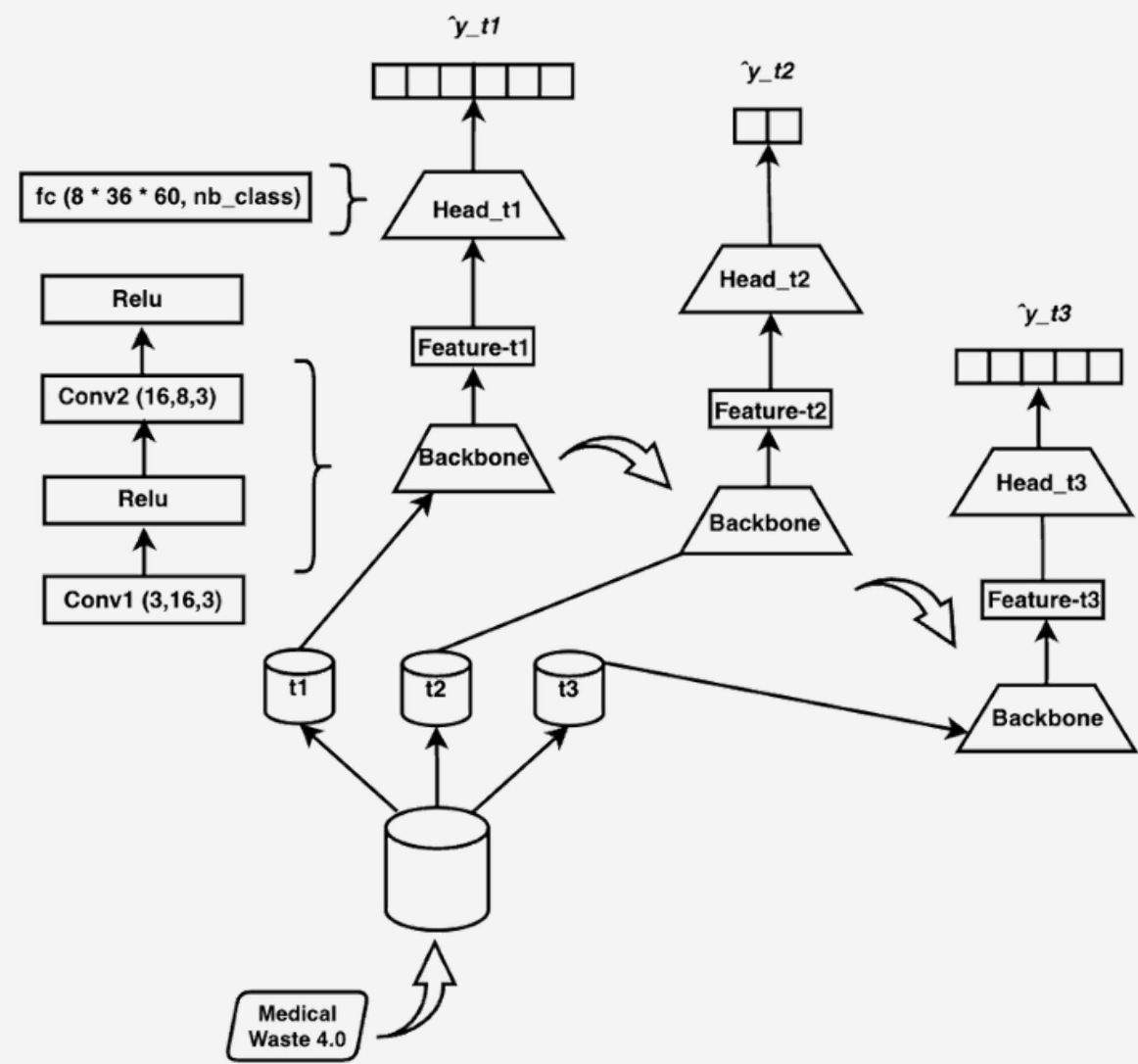
Tâche 3
gauze
medical_glasses
test_tube

Tâche 1
glove_pair_latex
glove_pair_nitrile
glove_single_nitrile
shoe_cover_single



Architecture et Hyperparamètre

Continual Learning



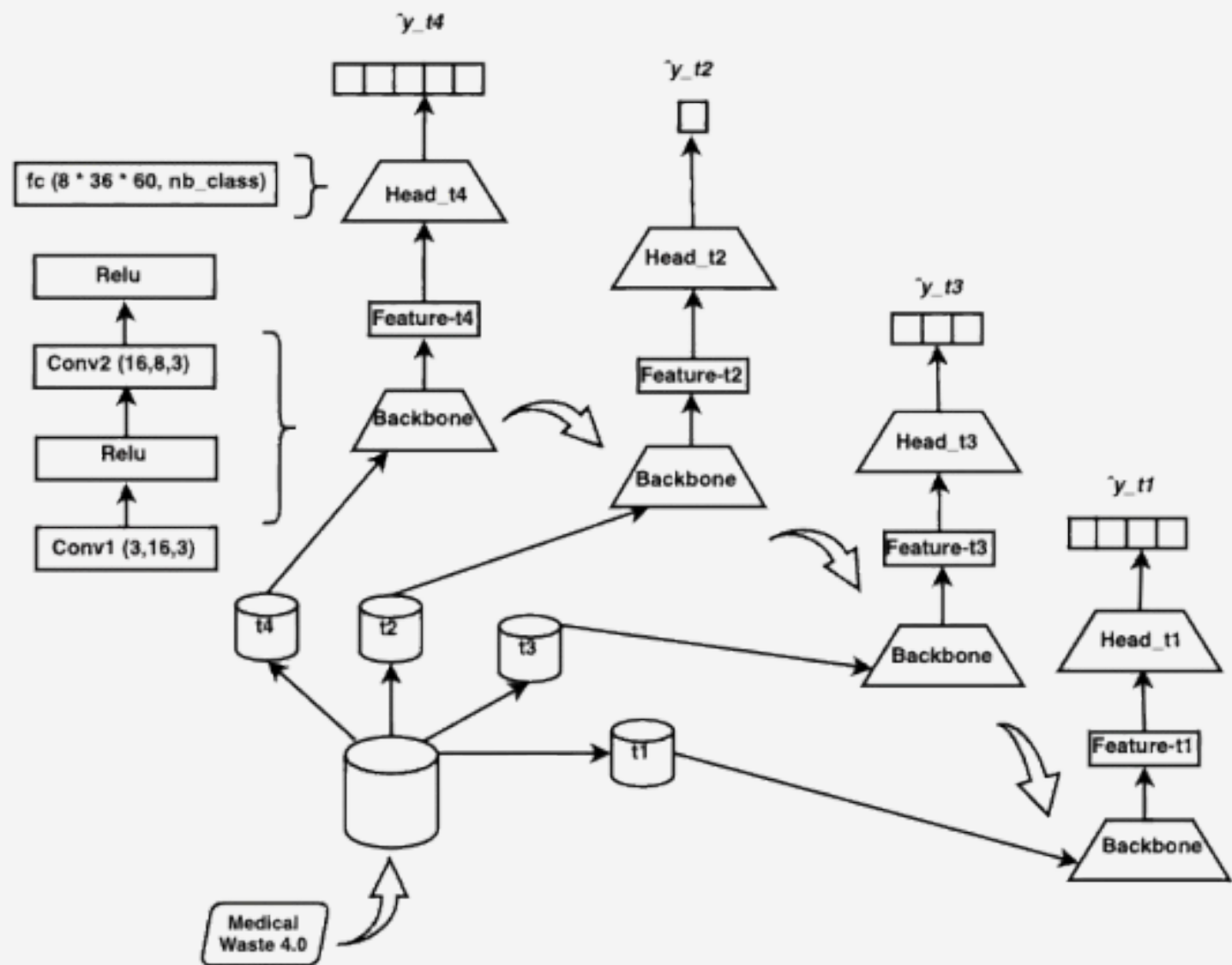
Continual Learning basé sur la Génération des tâches via Sémantique

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

Hyper-paramètre
en Sémantique

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

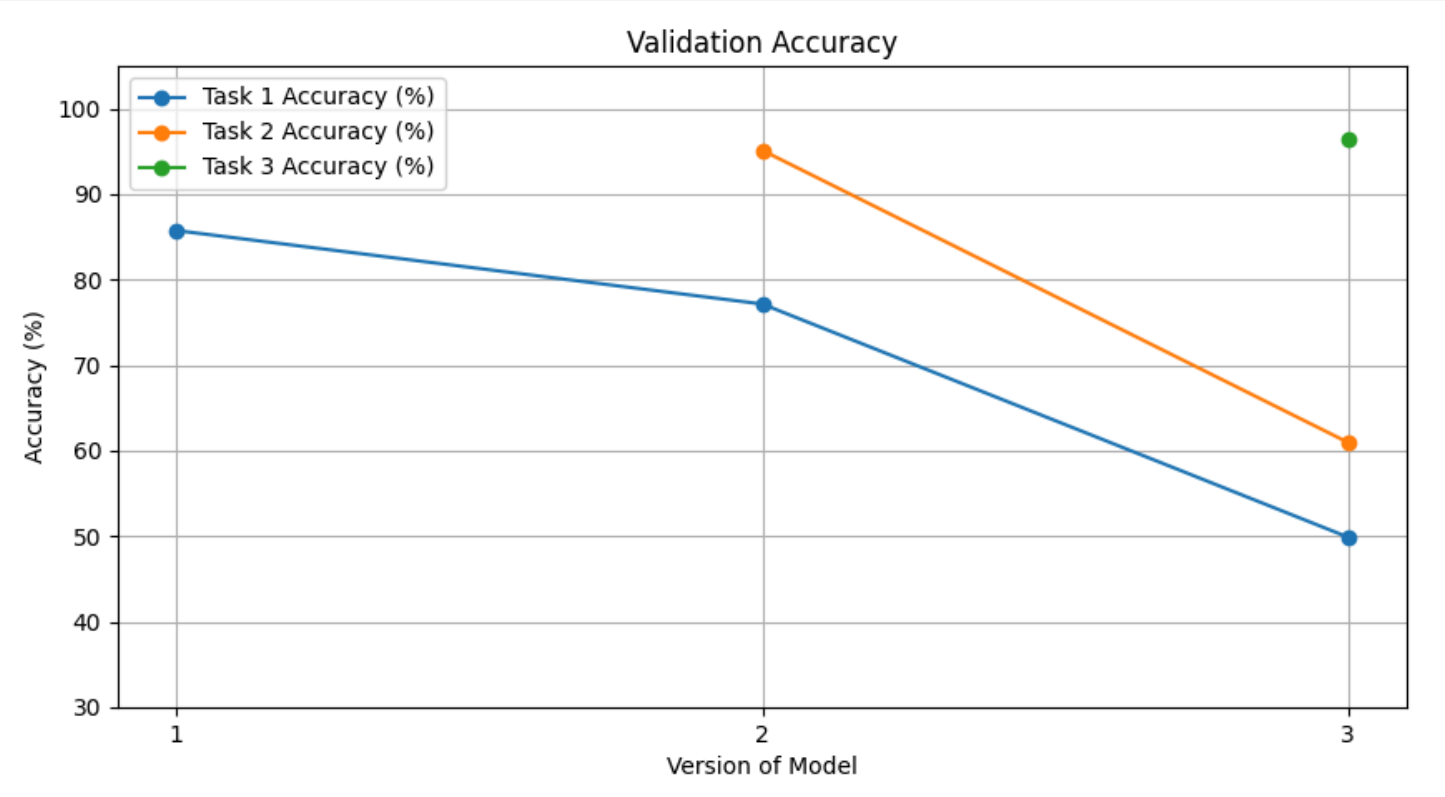
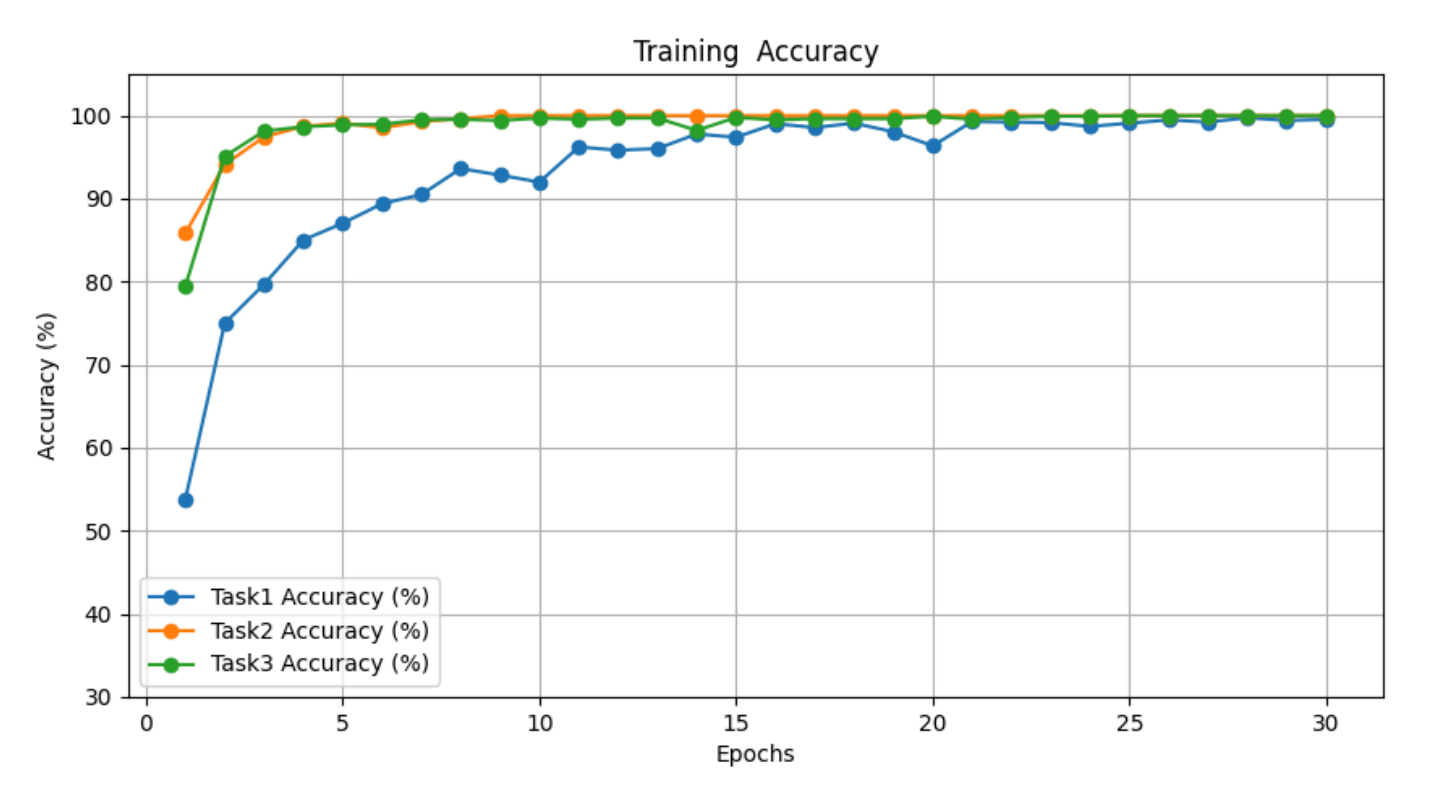
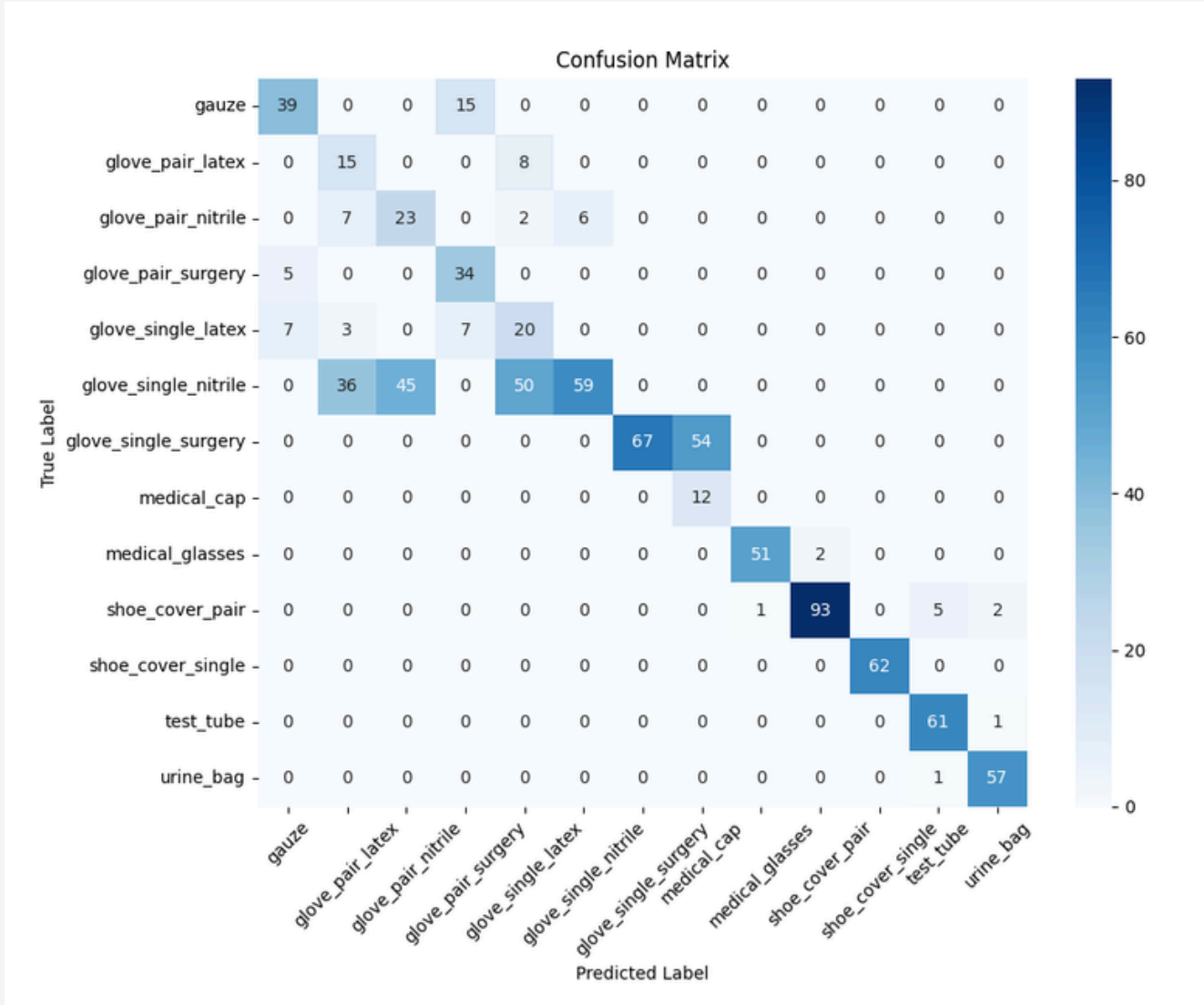
Hyper-paramètre en
Clustering Hiérarchique



Continual Learning basé sur la Génération des tâches via Clustering Hiérarchique

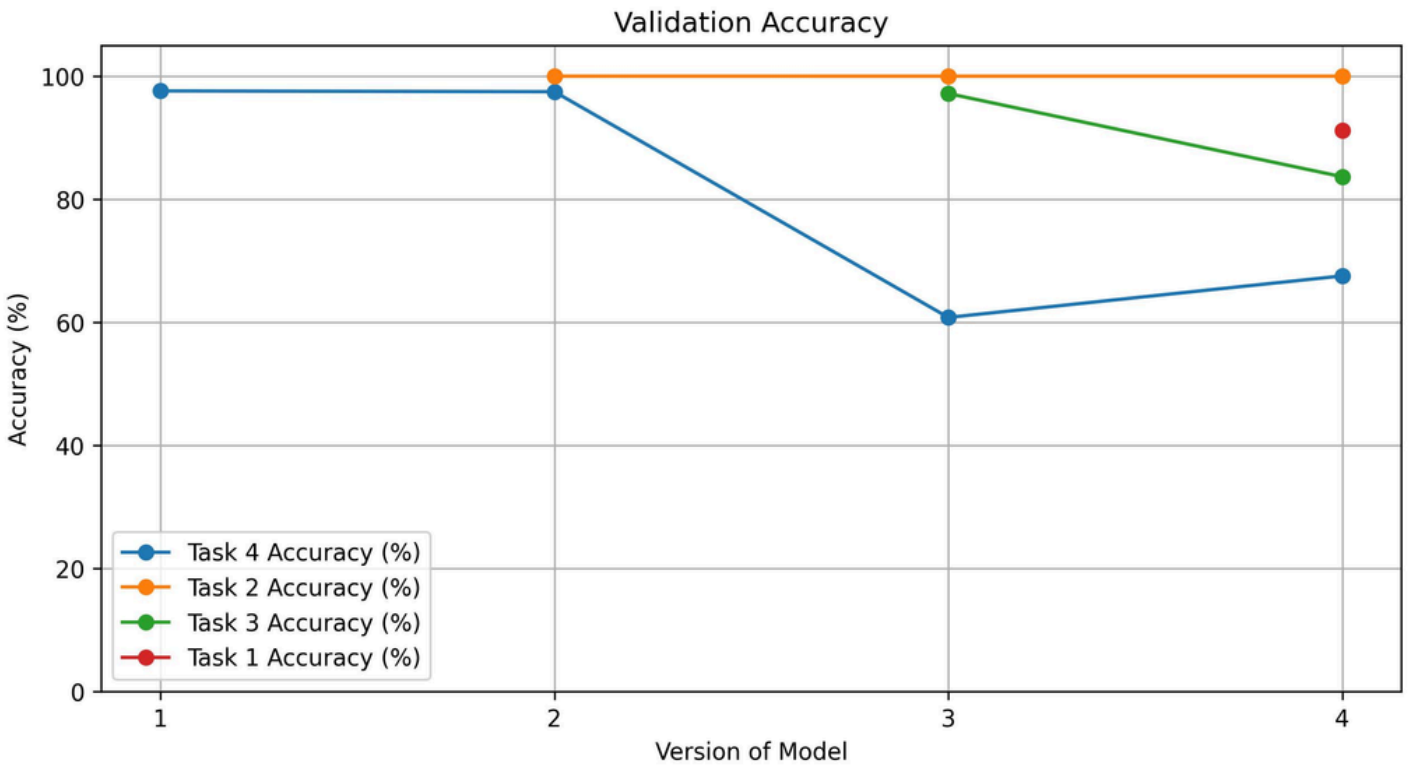
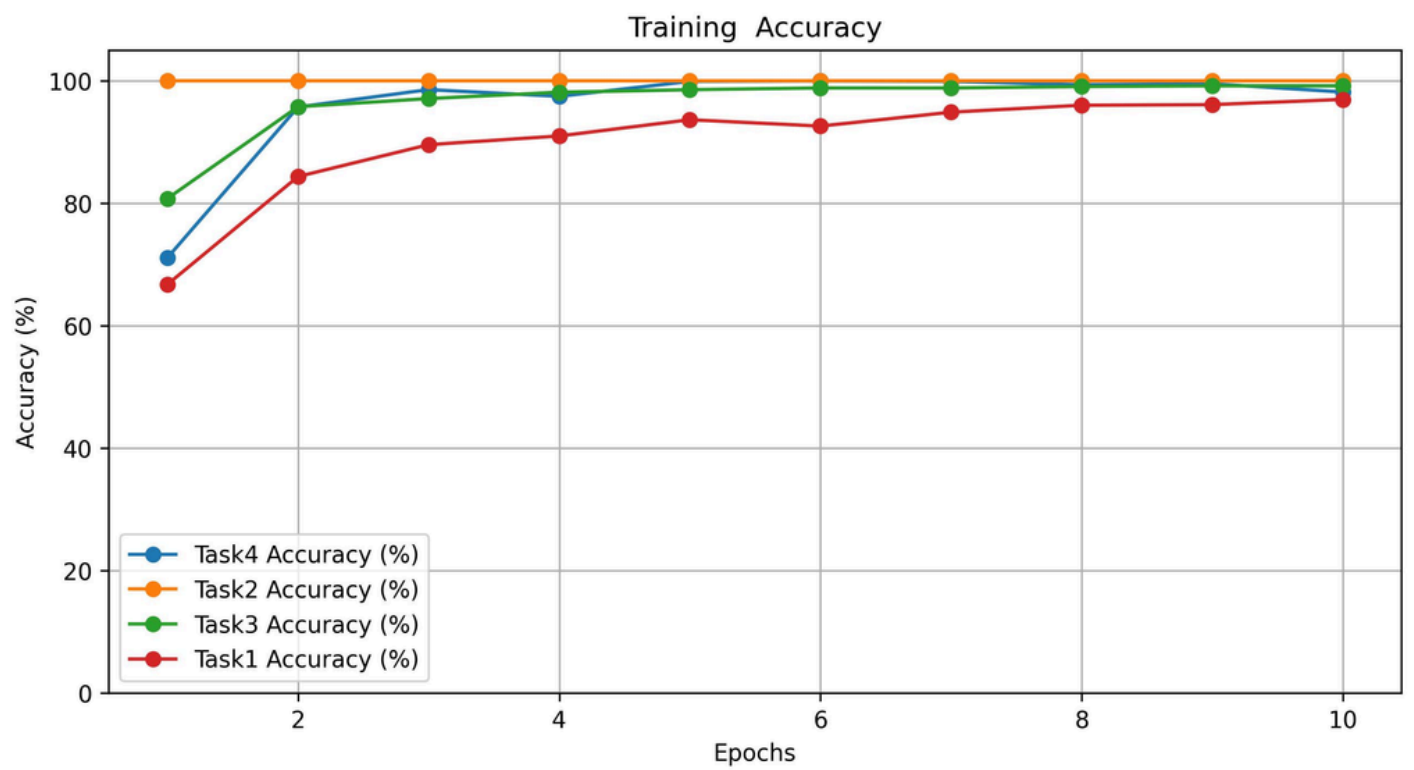
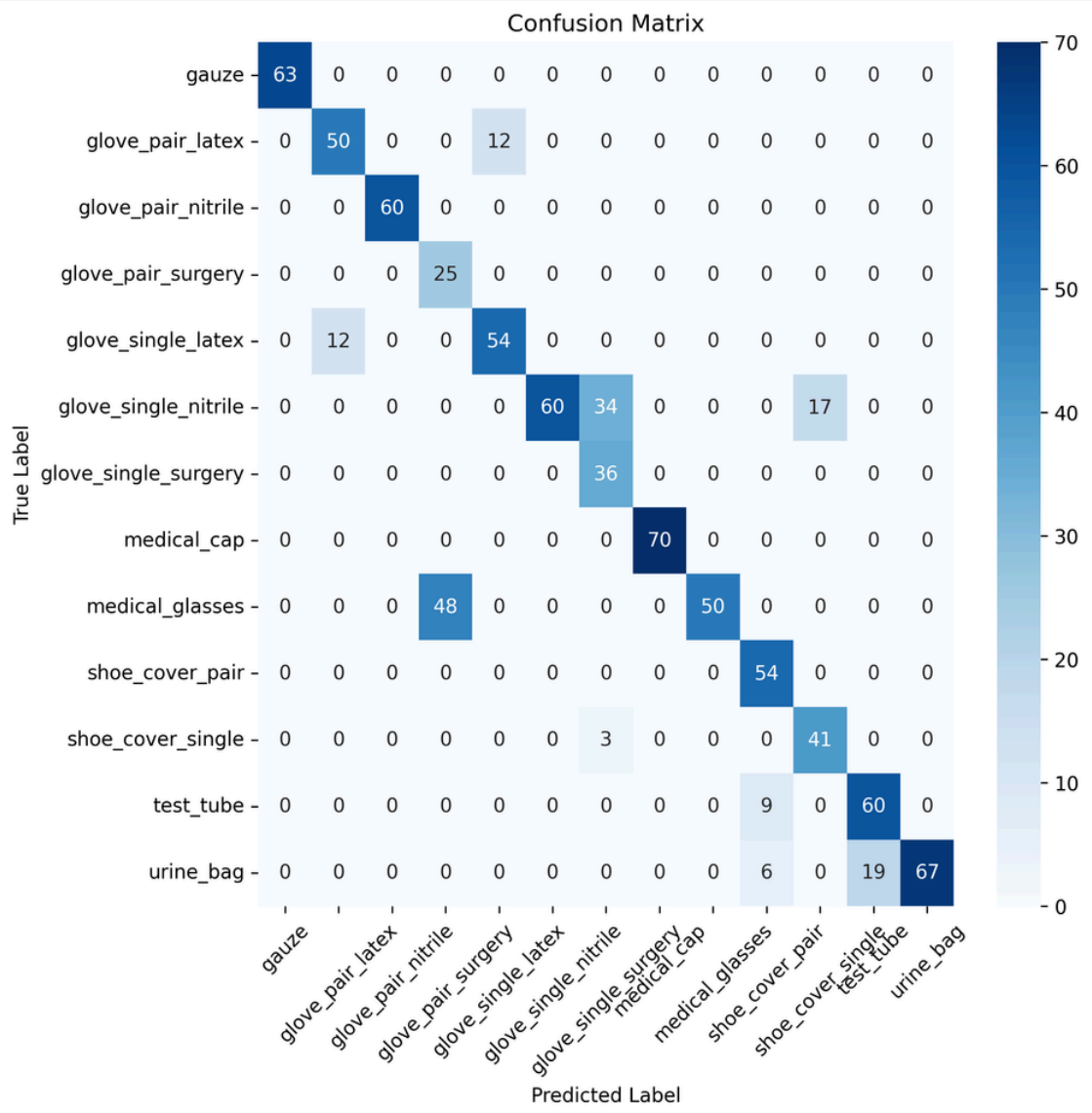
Résultat de Continual Learning

Continual Learning basé sur la Génération des tâches via Sémantique



Résultat de Continual Learning

Continual Learning basé sur la Génération des tâches via Clustering Hiérarchique



Discussion

La méthode *Flatten*, en traitant toutes les données d'entraînement en une seule fois, et une précision globale de 90 %.

La méthode *MTL* améliore davantage la méthode *Independent Task*. En partageant le réseau principal du modèle et en introduisant un lot de données de tâche à chaque itération d'entraînement, augmentant la précision globale à 97 %.

La méthode *Independent Task* rend le modèle plus concentré sur des tâches spécifiques, ce qui améliore significativement la précision de classification, atteignant 93 %.

Lors de la simulation avec de nouvelles données de tâches, la méthode *Continual Learning* a révélé des problèmes liés à l'oubli des tâches précédemment apprises et aux interférences entre les tâches, entraînant une diminution de la précision globale à 69,7 %.

Approfondi l'étude de méthodes plus raisonnables pour classer les catégories, de l'impact de l'ordre des tâches pendant l'entraînement, et de l'effet de la résolution des jeux de données sur les performances du modèle.

Les résultats montrent que, que ce soit avec le dataset *Transform Resolution* (précision de 81,17 %).

Le dataset *Full Resolution* (précision de 81,88 %).

Travaux Futurs

01

Extension de la portée du DataSet

En s'entraînant sur des données plus riches, le modèle peut mieux gérer des scénarios plus complexes et valider simultanément sa capacité de généralisation dans divers environnements de tâches.

02

Optimisation des paramètres du modèle

Des outils d'optimisation (tels que la recherche de grille, la recherche aléatoire ou l'optimisation bayésienne) peuvent être utilisés pour optimiser systématiquement les hyperparamètres et les paramètres d'entraînement (par exemple, nb epoch) afin d'améliorer les performances de validation.

03

Atténuer l'oubli catastrophique

Préserve les caractéristiques des tâches précédentes, réduit les oublis, équilibre les tâches nouvelles et OL, et stabilise et améliore les performances globales des modèles et les résultats de validation.



Merci Beaucoup

31/01/2025