Jaheer Alee GOULAM Option SIA

Abdelhamied OMAR YOUSIF

Rapport de projet 3A

Generative Multi-modal Models for Class-Incremental Learning

Encadrant :

VU son 16/01/2025

Abstract

L’apprentissage incrémental de classes (CIL) est un défi majeur en intelligence artificielle : les modèles doivent apprendre de nouvelles classes au fil du temps tout en conservant ce qu’ils ont déjà appris. Les modèles classiques, dits discriminatifs, ont tendance à "oublier" les anciennes classes lorsqu’ils en apprennent de nouvelles, un problème connu sous le nom d’oubli catastrophique. Pour résoudre ce problème, ce projet propose une approche innovante basée sur les modèles génératifs multi-modaux (GMM).

Contrairement aux modèles discriminatifs, qui doivent sans cesse adapter leur architecture pour intégrer de nouvelles classes, les GMM fonctionnent différemment : ils génèrent des descriptions textuelles détaillées à partir des images, puis comparent ces descriptions aux noms des classes déjà connues. Cette méthode évite d’avoir à modifier la structure du modèle à chaque nouvelle classe, réduisant ainsi les risques d’oubli et les biais en faveur des tâches récentes.

Le modèle GMM s’appuie sur des composants pré-entraînés, comme un encodeur d’images et un décodeur de texte (inspiré de MiniGPT-4), pour transformer les caractéristiques visuelles en descriptions textuelles. Une couche de projection linéaire permet d’aligner les représentations visuelles avec l’espace textuel, ce qui facilite l’intégration de nouvelles classes sans perturber les connaissances existantes. Cette approche est non seulement efficace, mais aussi économique en termes de calcul, car elle ne nécessite pas de réentraîner l’ensemble du modèle.

Les tests réalisés sur le dataset ImageNet-R montrent que les GMM surpassent les méthodes traditionnelles et même les modèles discriminatifs pré-entraînés les plus récents. Par exemple, le GMM atteint une précision moyenne de 83,95 % sur 20 tâches avec Tiny-ImageNet et 89,41 % sur ImageNet-R, démontrant ainsi sa capacité à apprendre de nouvelles classes tout en conservant une performance élevée sur les anciennes.

En résumé, les GMM offrent une solution robuste et évolutive pour l’apprentissage incrémental, combinant flexibilité, efficacité et performance. Cette approche ouvre des perspectives prometteuses pour des applications où l’adaptation continue et la préservation des connaissances sont essentielles.

Remerciement :

Nous souhaitons exprimer notre gratitude envers notre professeur, M. Son VU, pour son accompagnement et ses précieux conseils tout au long de la réalisation de ce projet.

Nos remerciements s'adressent également à l'ensemble de l'équipe pédagogique de l'ENSEA pour les connaissances approfondies qu'elle nous a transmises dans divers domaines, notamment les mathématiques et l'intelligence artificielle, qui ont grandement facilité notre compréhension et notre progression.

Nous remercions tout particulièrement M. Laurent PROTOIS et M. Nicolas Larue pour la création des comptes Serveur ETIS, nous permettant ainsi d'accéder aux ressources informatiques du laboratoire.

Enfin, nous tenons à remercier chaleureusement toutes les personnes qui ont contribué, de près ou de loin, au développement de notre projet de troisième année, intitulé "Generative Multi-modal Models.

Table des matières

I. Introduction

I.1 Contexte et motivation

L'apprentissage incrémental (CIL) est un domaine crucial de l'intelligence artificielle, car il reflète la capacité des systèmes à s'adapter à des environnements en constante évolution. Contrairement aux modèles traditionnels, qui nécessitent un accès simultané à l'ensemble des données pour s'entraîner, les systèmes d'apprentissage incrémental doivent intégrer de nouvelles classes ou tâches au fil du temps tout en conservant leurs connaissances précédentes. Ce paradigme imite l'apprentissage humain, où les connaissances accumulées sont enrichies en continu sans pour autant oublier les acquis passés.

Cependant, les modèles classiques, souvent discriminatifs, rencontrent un obstacle majeur : **l'oubli catastrophique**. Ce phénomène se manifeste par une perte progressive des connaissances précédentes lorsque le modèle apprend de nouvelles classes. Dans des applications telles que la reconnaissance faciale ou les systèmes embarqués, où la mémoire et les capacités de calcul sont limitées, ce problème limite considérablement les performances des modèles.

I.2 Problématique

L'apprentissage incrémental de classes (CIL) est une approche de l'intelligence artificielle dans laquelle les modèles doivent apprendre de nouvelles classes ou tâches de manière séquentielle, tout en conservant les connaissances acquises sur les classes déjà apprises. Contrairement aux méthodes classiques, où toutes les données sont disponibles simultanément pour entraîner le modèle, le CIL suppose que les données arrivent progressivement, reflétant des scénarios du monde réel, tels que l'intégration continue de nouveaux concepts ou catégories dans des systèmes existants.

L'intérêt de l'apprentissage incrémental réside dans sa capacité à répondre à des besoins critiques dans divers domaines :

- **Adaptation dynamique** : Il permet aux systèmes d'intelligence artificielle de s'adapter en continu à des environnements changeants, par exemple dans des systèmes de reconnaissance faciale où de nouveaux utilisateurs sont ajoutés au fil du temps.
- **Réduction des coûts de stockage et de calcul** : Plutôt que de conserver toutes les données précédentes pour un nouvel entraînement, le CIL vise à minimiser la mémoire utilisée tout en évitant la nécessité de recalculer les paramètres à chaque ajout.

- **Imitation du processus humain** : Inspiré de la façon dont les humains apprennent de manière cumulative, le CIL cherche à doter les modèles d'une capacité similaire, leur permettant de combiner de nouvelles connaissances tout en préservant les acquis.

Dans les scénarios d'apprentissage incrémental de classes (CIL), les modèles discriminants classiques sont confrontés à un problème majeur : l'oubli catastrophique. Ce phénomène survient lorsque ces modèles, après entraînement sur de nouvelles classes ou tâches, perdent tout ou partie des connaissances acquises sur les anciennes classes. En d'autres termes, le modèle devient "amnésique" et ne parvient plus à reconnaître ou classer correctement les données des tâches précédentes.

Cette limitation découle directement du fonctionnement des modèles discriminants, qui reposent sur une tête de classification. Cette couche spécifique associe les représentations des données (comme les caractéristiques d'une image) à des classes prédéfinies. Lorsque de nouvelles classes doivent être ajoutées, il est souvent nécessaire d'étendre cette tête de classification, ce qui entraîne une modification de l'architecture du réseau.Par ailleurs, cette extension nécessite une réévaluation complète des paramètres du modèle, ce qui augmente le risque de biais structurels, où le modèle devient plus performant sur les nouvelles classes au détriment des anciennes, favorisant ainsi les classes récentes et renforçant l'oubli des connaissances précédemment acquises.

Ces contraintes posent un défi particulièrement aigu dans des contextes où les données sont disponibles de manière séquentielle ou lorsque la mémoire pour stocker les données anciennes est limitée. Les approches traditionnelles peinent alors à concilier l'intégration efficace de nouvelles connaissances avec la préservation des savoirs précédents.

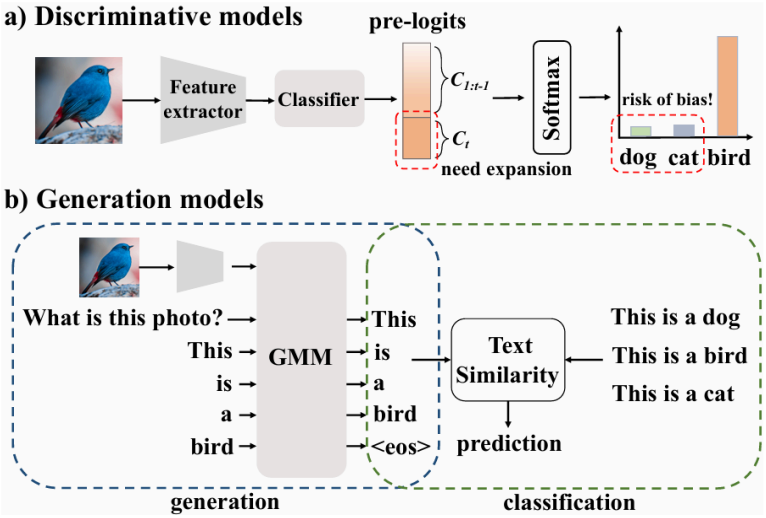

Figure 1. Illustration de modèles discriminants conventionnels et de modèles génératifs (GMM) pour CIL.

Les modèles génératifs multi-modaux (GMM) : Une approche innovante pour l'apprentissage incrémental

Contrairement aux modèles discriminants traditionnels, les modèles génératifs multimodaux (GMM) offrent une approche novatrice et efficace pour surmonter les défis de l'apprentissage incrémental. Alors que les modèles discriminants nécessitent une extension constante de leur tête de classification pour intégrer de nouvelles classes, les GMM adoptent une méthodologie fondamentalement différente, basée sur la génération et l'analyse textuelle.

Une nouvelle vision pour l'apprentissage incrémental

Les modèles génératifs, au lieu d'associer directement des caractéristiques visuelles à des classes spécifiques, se concentrent sur la création de descriptions textuelles détaillées à partir des images. Ces descriptions, riches en informations, permettent de capturer les caractéristiques visuelles et contextuelles des images d'une manière plus exhaustive. Par exemple, une image d'un oiseau peut être décrite par le modèle comme : *« Ceci est une photo d'un oiseau avec des plumes colorées et un long bec. »*

Ces textes générés sont ensuite comparés aux noms des classes préexistantes en utilisant un encodeur textuel. L'encodeur évalue la similarité entre le texte produit par le modèle et les descriptions des catégories déjà apprises. Cette comparaison repose sur la similarité cosinus, permettant ainsi de déterminer la classe la plus proche. Ce processus élimine la nécessité d'une tête de classification élargie et réduit le risque de biais en faveur des classes les plus récentes, un problème récurrent dans les modèles discriminants.

Avantages clés des GMM dans le CIL

L'utilisation des GMM pour l'apprentissage incrémental présente plusieurs avantages significatifs :

Élimination de l'extension de la tête de classification
Contrairement aux modèles discriminants, les GMM ne nécessitent pas d'agrandir constamment leur tête de classification à chaque nouvelle tâche ou classe. Cette caractéristique simplifie non seulement l'architecture du réseau, mais réduit également les risques d'introduire des biais structurels dans le modèle.

Réduction du biais vers les tâches récentes
L'un des problèmes majeurs des modèles discriminants est leur tendance à privilégier les tâches les plus récentes, entraînant ainsi l'oubli des anciennes classes. Avec les GMM, cette dépendance est supprimée, car le processus de classification repose sur une comparaison textuelle plutôt que sur une structure hiérarchique étendue. Cela permet de maintenir une performance constante sur toutes les classes apprises.

Diminution de l'oubli catastrophique
En évitant les modifications fréquentes de l'architecture du réseau, les GMM réduisent considérablement les risques d'oubli catastrophique. Le modèle conserve ses connaissances sur les classes précédentes tout en intégrant efficacement de nouvelles classes.

I.3 Objectifs du projet

Face à ces défis, ce projet vise à concevoir une solution innovante pour surmonter l'oubli catastrophique. Les objectifs principaux sont :

1. Analyser les causes fondamentales de l'oubli catastrophique dans les architectures discriminantes et proposer une alternative basée sur des modèles génératifs multi-modaux (GMM). Cette méthode est proposée par le papier scientifique sur lequel est basé notre projet.

2. Exploiter les GMM pour générer des descriptions textuelles riches et contextuelles à partir des caractéristiques visuelles, permettant une classification sans modification de l'architecture.

3. Tester et valider cette approche sur des jeux de données standard comme ImageNet-r.

II. État de l'art

II.1 Apprentissage incrémental par classe (CIL)

L'apprentissage incrémental par classe (CIL) permet à un modèle d'apprendre de nouvelles classes sans oublier celles précédemment acquises. Ce paradigme est toutefois confronté au problème de l'oubli catastrophique, causé par un biais vers les données des nouvelles tâches.
Méthodes classiques :

1. **Réentraînement** (Rehearsal-based) :
Conservation d'exemples des classes passées (originaux, générés ou sous forme de caractéristiques) pour guider l'apprentissage. Exemple : iCaRL.
2. **Adaptation architecturale** (Architecture-based) :
Expansion dynamique du réseau ou ajout de modules pour intégrer de nouvelles classes. Exemple : Dynamic Expandable Representation (DER).
3. **Régularisation** (Regularization-based) :

Contrôle des mises à jour des poids critiques pour les anciennes tâches, comme dans Elastic Weight Consolidation (EWC).

Limites :

- Dépendance à la mémoire pour stocker les anciennes données.
- Complexité accrue avec les approches architecturales.
- Biais résiduel vers les nouvelles tâches, même avec régularisation.

II.2 Modèles génératifs multimodaux existants

Les modèles génératifs multi-modaux (**GMM**) exploitent à la fois des données visuelles et textuelles pour produire des représentations riches et interconnectées. Ces modèles génèrent des descriptions détaillées des images, permettant une classification qui ne dépend pas de la structure classique des modèles discriminatifs. Ils constituent une avancée majeure pour des tâches comme l'apprentissage incrémental par classe.
Exemples de GMM existants :

1. **CLIP** (Contrastive Language-Image Pretraining) :

CLIP aligne les représentations d'images et de textes dans un espace latent partagé à l'aide d'une perte contrastive. Il est particulièrement performant pour des tâches de classification sans étiquettes spécifiques grâce à sa capacité d'appariement texte-image.

2. **MiniGPT-4** :

Ce modèle combine un encodeur visuel (ViT) et un décodeur textuel basé sur un grand modèle de langage (LLM). Une couche de projection aligne les caractéristiques visuelles sur l'espace textuel, facilitant la génération de descriptions précises et exploitables pour la classification.

3. **BLIP-2 et Flamingo** :

Ces modèles exploitent des mécanismes avancés comme la cross-attention pour fusionner les informations visuelles et textuelles, améliorant les performances dans diverses tâches multi-modales.

III. Présentation globale du projet

III.1 Aperçu de l'architecture

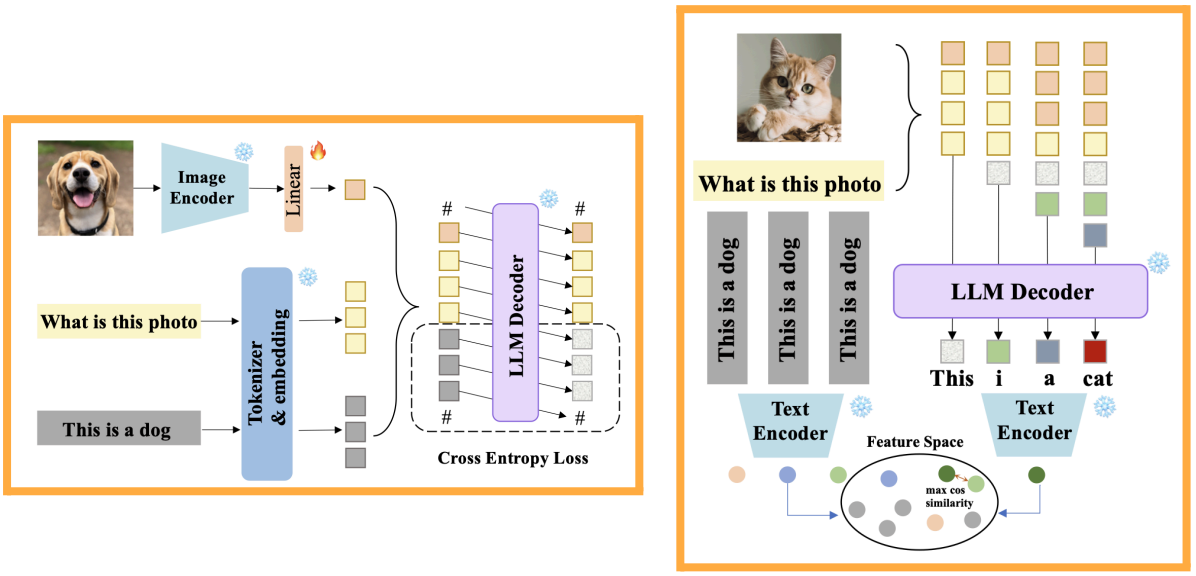

Figure 1. Présentation de l'architecture. A gauche, la partie génération avec Minigpt-4. A droite la partie classification avec CLIP

L'architecture proposée repose sur un modèle génératif multi-modal qui combine des images et du texte pour effectuer une classification incrémentale, tout en minimisant l'oubli catastrophique. Voici les étapes clés de son fonctionnement :

1. **Entrée de l'image avec une question :**

Une image est fournie comme entrée, accompagnée d'une question générique : « Décris cette image ». Cette approche guide le modèle pour produire une description textuelle centrée sur le contenu principal de l'image.

2. **Extraction des caractéristiques visuelles :**

L'image est traitée par un encodeur d'images (par exemple, Vision Transformer - ViT) qui extrait des caractéristiques visuelles riches et compactes. Ces caractéristiques sont transmises à un décodeur textuel.

3. **Génération d'une description textuelle :**

Le décodeur de texte, basé sur un LLM (Large Language Model), génère une phrase descriptive sous le format :
« Ceci est un [classe] ».
Par exemple : « Ceci est un chien », « Ceci est un chat », etc.

4. **Ajout progressif des classes dans CLIP :**

Dans le cadre de l'apprentissage incrémental par classe (CIL), à chaque nouvelle tâche, les classes nouvellement apprises sont ajoutées à l'ensemble des classes cibles existantes. Chaque classe est représentée par une phrase standardisée, telle que :

- « Ceci est un chien »,
- « Ceci est un chat »,
- « Ceci est un oiseau »,
- « Ceci est un classe N » (pour une nouvelle classe ajoutée).

Ainsi, le modèle peut effectuer une classification sur l'ensemble des classes vues jusqu'à présent.

5. **Comparaison dans l'espace latent :**

Les descriptions générées par le LLM et les noms des classes cibles sont comparés dans l'espace latent partagé de CLIP, qui aligne les représentations texte-image. La similarité cosinus est utilisée pour déterminer la classe ayant la plus grande correspondance avec la description générée.

6. **Optimisation par apprentissage incrémental :**

Seule une **couche linéaire** reliant les caractéristiques visuelles à l'espace textuel est entraînée pour chaque nouvelle tâche. Cela permet au modèle d'apprendre les nouvelles classes tout en préservant les représentations des anciennes, sans nécessiter de réentraînement complet.

Avantages de l'architecture :

- **Évolution naturelle des classes** : Les nouvelles classes sont simplement ajoutées aux représentations textuelles sans perturber les anciennes.
- **Réduction de l'oubli catastrophique** : Le gel des poids des encodeurs d'images et de textes limite les perturbations liées aux tâches précédentes.
- **Flexibilité et économie computationnelle** : En évitant l'expansion de la tête de classification, l'architecture reste légère et performante.

III.2 Scénarios d'application et défis

Les modèles génératifs multi-modaux pour l'apprentissage incrémental trouvent des applications dans divers domaines où l'adaptation continue et la préservation des connaissances sont cruciales. Voici quelques scénarios clés et défis associés :

Scénarios d'application :

1. **Systèmes embarqués :**

   - Reconnaissance faciale ou d'objets sur des appareils mobiles ou des robots, où les nouvelles classes doivent être intégrées sans compromettre les performances passées.
   - Exemple : Ajouter une nouvelle catégorie d'objet à une caméra intelligente sans réentraîner entièrement le modèle.

2. **Surveillance et sécurité :**
   - Surveillance vidéo avec détection de nouvelles menaces ou activités anormales. Les systèmes doivent apprendre en temps réel tout en maintenant les connaissances des scénarios précédents.

3. **Applications médicales :**
   - Classification d'images médicales où de nouvelles pathologies ou modalités d'imagerie doivent être intégrées progressivement. Cela réduit les besoins en stockage des anciennes données sensibles.

IV. Phase de Training

IV.1 Processus général de training

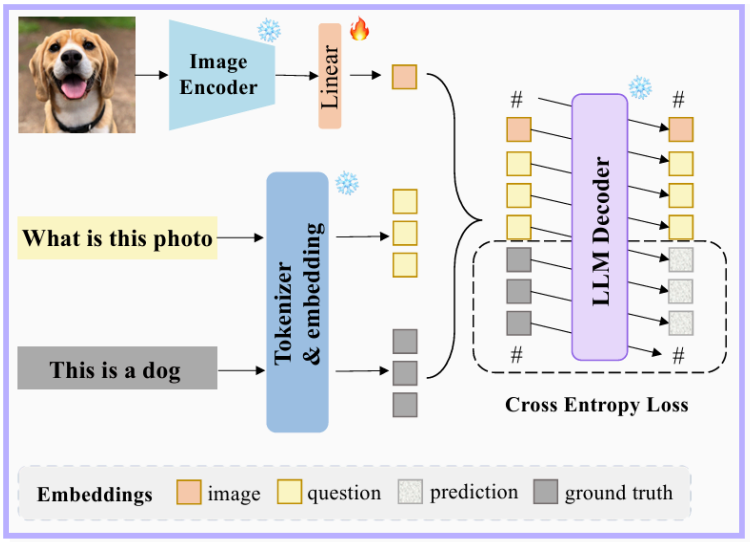

Figure 2. Training process

Datasets utilisé

Pour évaluer l'efficacité du modèle proposé, le dataset ImgR-alin est utilisé. Ce dataset est conçu pour tester la robustesse des modèles face à des styles visuels variés et des représentations inhabituelles des classes.

Caractéristiques d'ImgR-alin :

- **Classes et diversité :**
  • Comprend 200 classes, avec des styles visuels variés, comme des croquis, des peintures et des photos réelles.
  • Les images ne suivent pas toujours des conventions visuelles classiques, ce qui pose un défi pour les modèles traditionnels.

- **Taille et distribution :**
  • Le dataset contient des images avec une distribution équilibrée dans 10 tasks différents. Chaque task contient 20 classes et 100 par classe..

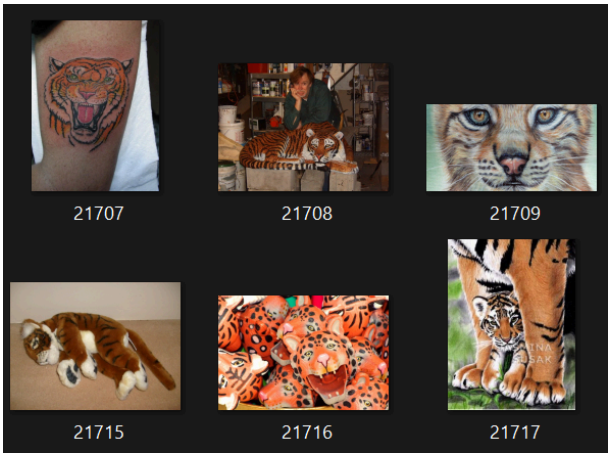

Figure 3. Exemple d'images dans le dataset ImgR-alin

Pour chaque image, une phrase descriptive est générée automatiquement. Par exemple, une image d'un chien est associée à la phrase : *"This is a photo of a dog."* Ces descriptions textuelles sont utilisées comme intermédiaires pour relier les représentations visuelles aux catégories cibles.

> **L'encodeur d'images** : extrait des caractéristiques visuelles riches de chaque image.
>
> **Le décodeur de texte** : lui, utilise ces caractéristiques pour produire des descriptions textuelles.

Ces deux composants pré-entraînés ne sont pas modifiés afin de conserver leurs connaissances initiales et de limiter les ajustements nécessaires. Cela permet également de réduire les coûts computationnels et d'accélérer l'entraînement.

IV.2 Projection linéaire (celui qu'on cherche à perfectionner) :

Dans le cadre de l'utilisation des modèles génératifs multi-modaux pour l'apprentissage incrémental par classe (CIL), l'élément central de l'entraînement réside dans l'optimisation exclusive de la couche de projection linéaire. Cette approche permet une adaptation efficace tout en minimisant le risque d'oubli catastrophique.

Rôle de la couche de projection linéaire (figure 2)

L'encodeur d'images produit des embeddings dans un espace visuel riche, représentant les caractéristiques des images de manière quantitative. Cependant, cet espace visuel est intrinsèquement différent de l'espace textuel utilisé par le décodeur, qui s'appuie sur des représentations linguistiques pour générer des descriptions en langage naturel.

La couche de projection linéaire agit comme un **transformateur clé**, remplissant plusieurs fonctions essentielles :

> **Alignement des espaces** : Elle ajuste les embeddings visuels pour qu'ils soient compatibles avec l'espace textuel du décodeur, facilitant ainsi le passage des caractéristiques visuelles aux descriptions textuelles précises.
>
> **Simplification des interactions** : En réduisant la complexité des interactions entre les composants visuels et textuels, cette couche permet au modèle de mieux exploiter les informations des images tout en s'appuyant sur les capacités linguistiques du décodeur.
>
> **Flexibilité pour les nouvelles classes** : Grâce à sa capacité à transformer dynamiquement les embeddings visuels, la couche linéaire s'adapte facilement à de nouvelles catégories ou tâches sans nécessiter de modifications structurelles majeures dans le reste du modèle.

Optimisation ciblée

L'entraînement du modèle se concentre exclusivement sur l'ajustement de la couche de projection linéaire, laissant les **autres composants** (encodeur et décodeur) **inchangés**. Cette approche offre plusieurs avantages stratégiques :

> **Adaptation rapide aux nouvelles classes** : La simplicité de cette couche permet une optimisation rapide, même lorsqu'elle est confrontée à de nouvelles données ou à des classes inconnues. Le modèle peut ainsi apprendre de manière efficace sans exiger de longues périodes d'entraînement.
>
> **Réduction de l'oubli catastrophique** : Dans les scénarios CIL, le principal défi réside dans la capacité à conserver les connaissances des tâches précédentes tout en apprenant de nouvelles informations. En limitant l'entraînement à la couche linéaire, les paramètres des composants principaux – encodeur et décodeur – restent stables, minimisant ainsi la perte de performance sur les classes apprises auparavant.

Minimisation de la perte d'entropie croisée

L'objectif d'entraînement est de minimiser une **perte d'entropie croisée**, qui évalue la divergence entre les descriptions générées par le modèle et les labels corrects associés aux images :

> Pour une image de chien, le modèle génère la phrase : *"This is a photo of a dog"*. La perte d'entropie croisée s'assure que cette sortie correspond parfaitement à la classe cible *dog*, en attribuant un score élevé aux mots pertinents tout en pénalisant les erreurs ou incohérences.

- **<u>Encodeur d'images et décodeur linguistique</u>** :

L'un des éléments essentiels du modèle génératif multi-modal est l'interaction fluide et efficace entre l'encodeur d'images et le décodeur linguistique. Ce dernier, basé sur un large langage model (LLM) inspiré de MiniGPT-4, s'appuie sur une combinaison de composants avancés, notamment ViT, Q-Former et Vicuna. Cette architecture garantit une transformation robuste des informations visuelles en descriptions textuelles claires et précises.

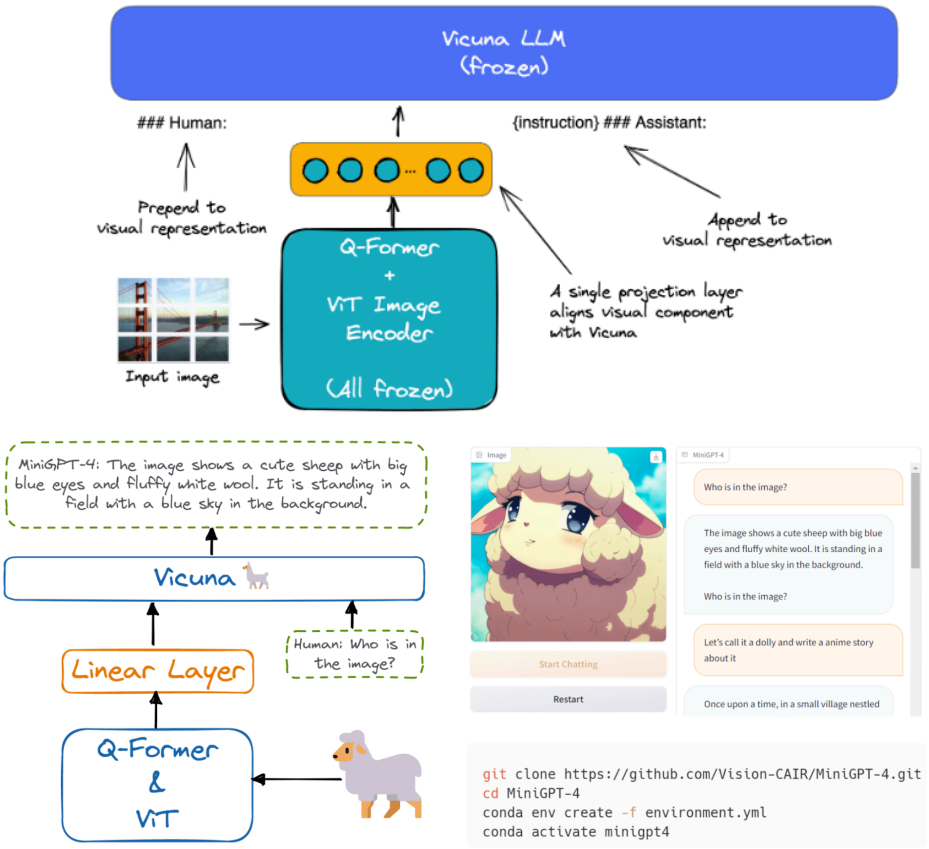

Figure 4. Architecture de mini-GPT4

IV.3 ViT (Vision Transformer)

Le rôle du ViT : Transformer les images en embeddings visuels riches

Transformation des pixels en "mots visuels" :

Le processus commence par une étape importante : la segmentation de l'image en *patches*, c'est-à-dire en petites portions de taille fixe qui représentent des parties spécifiques de l'image. Ces *patches* sont ensuite transformés en vecteurs numériques, ce qui permet de les traiter plus facilement par le modèle. De plus, un *embedding positionnel* est ajouté pour

donner un sens à la position de chaque patch dans l'image, afin de conserver l'ordre et la structure originale. Une fois transformés, ces *patches* sont traités comme des "mots visuels", de la même manière qu'un modèle de langage traite des mots dans une phrase.

Mécanismes d'attention :

Le ViT utilise des **mécanismes d'attention** pour analyser non seulement chaque patch individuellement, mais aussi les relations entre tous les patches en même temps. Ce processus permet au modèle de comprendre non seulement les détails locaux, comme les textures et les couleurs, mais aussi les relations globales entre les différentes parties de l'image. Globalement, le ViT se concentre sur les zones importantes de l'image et saisit les interactions complexes entre les éléments, ce qui est crucial pour bien comprendre le contenu visuel.

Représentation dans un espace latent :

Le résultat est un ensemble d'**embeddings visuels**, des vecteurs dans un espace latent qui encapsulent les informations essentielles de l'image, prêtes à être exploitées par les étapes suivantes.

Le rôle du Q-Former :

Le Q-Former intervient comme un composant intermédiaire, assurant une transition harmonieuse entre les embeddings visuels produits par ViT et le décodeur linguistique Vicuna.

Affinage des embeddings : À l'aide de mécanismes d'attention, le Q-Former affine les embeddings visuels en filtrant et en sélectionnant uniquement les informations pertinentes pour la tâche de classification ou de description.

Alignement intermodal : Il agit comme une passerelle entre les espaces visuel et textuel, traduisant les informations visuelles complexes en un format exploitable par le modèle linguistique. Cette étape est cruciale pour éviter toute perte d'information lors de la transition entre ces deux modalités.

Le rôle de Vicuna : Générer des descriptions textuelles explicites

Vicuna, en tant que décodeur linguistique, est responsable de la génération des descriptions textuelles finales.

Exploitation des embeddings transformés : En utilisant les embeddings affinés par le Q-Former, Vicuna génère des phrases textuelles claires, cohérentes et alignées sur les catégories visuelles.

Format des sorties : Le décodeur suit un format standardisé, produisant des descriptions telles que *"This is a photo of [CLS]"*, où [CLS] correspond à la classe de l'image (par exemple, *"dog"* ou *"cat"*).

Préentraînement sur de vastes ensembles : Grâce à un préentraînement sur de vastes ensembles de données textuelles, Vicuna excelle dans la génération de phrases linguistiquement correctes et riches en contexte.

Après l'entraînement, le modèle génératif multi-modal (GMM) est soumis à une phase de testing afin d'évaluer sa capacité à classifier des images de manière progressive, tout en conservant les connaissances acquises lors des tâches précédentes. Cette étape repose sur un alignement précis entre la vision et le langage, rendu possible grâce à l'utilisation de composants pré-entraînés comme CLIP. En exploitant cet alignement, le modèle transforme efficacement les descriptions textuelles qu'il génère en classifications précises, même dans des contextes d'apprentissage incrémental où de nouvelles classes s'ajoutent au fil du temps. Cette approche garantit robustesse et continuité dans les performances du modèle.

V. Phase de Testing

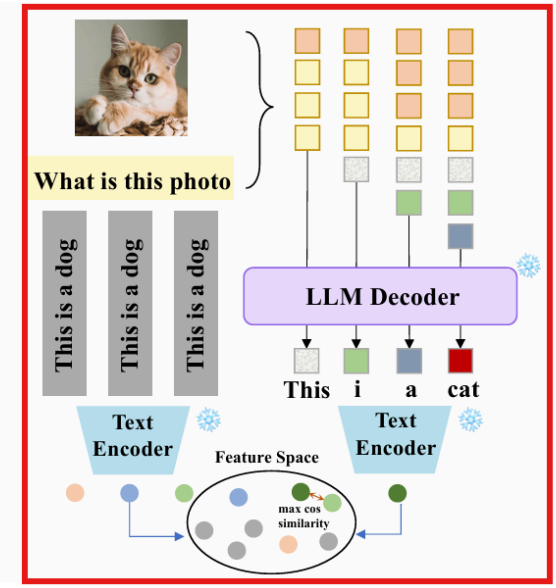

Figure 5. Testing process

V.1 Rôle et avantages de CLIP

Le testing est une étape clé dans notre approche, où nous utilisons CLIP pour associer les descriptions textuelles générées à partir des images avec les noms de classes déjà apprises. Cette méthode permet d'améliorer la précision tout en gardant le système simple et efficace.

Concrètement, le testing se déroule en trois étapes :

1. **Génération des descriptions textuelles** : Pour chaque image, le modèle génère une courte description qui résume son contenu, comme "Ceci est une photo de voiture".
2. **Encodage et comparaison** : Ces descriptions et les images sont traduites en représentations numériques dans un espace partagé grâce à **CLIP**. Cela permet de les comparer directement.
3. **Identification de la classe** : La classe correspondant à la description la plus proche dans cet espace est choisie comme prédiction finale.

Ce processus exploite toute la puissance de CLIP pour garantir une correspondance précise et cohérente entre les images et les classes

Génération et traitement des descriptions pour la classification d'images

V.2 Génération des descriptions textuelles

Lorsqu'une image est testée, le système génère automatiquement une description courte et précise, comme par exemple : *"This is a photo of a car"*. Cette phrase standardisée simplifie l'analyse et assure une compatibilité directe avec les noms des classes apprises. Ce format concis permet de gérer facilement de nouvelles classes sans modifier la structure du modèle.

V.3 Encodage et comparaison

Après avoir généré une description, le modèle utilise CLIP pour encoder cette description de l'image et un autre texte qui ont le même format standardisé en représentations numériques (*embeddings*). Ces représentations sont comparées aux noms des classes déjà apprises. La comparaison s'appuie sur la **distance cosinus**, une méthode qui mesure la similitude entre deux vecteurs. Cela permet de déterminer à quel point le texte généré est proche des noms de classes existants.

V.4 Identification des classes

La prédiction finale est effectuée en sélectionnant la classe dont l'embedding est le plus similaire à celui de la description générée. Par exemple, si la phrase produite est : *"This is a photo of a dog"*, et que l'embedding du mot "dog" est le plus proche, alors la classe "dog" sera choisie. Cette méthode garantit que le modèle peut reconnaître des classes apprises lors de tâches précédentes, même si elles ne font pas partie de la tâche actuelle.

Pourquoi utiliser CLIP dans le Testing ?

CLIP joue un rôle central dans le testing en agissant comme un lien entre les descriptions générées par le système et les noms de classes, comme "This is a dog". Grâce à son espace latent multimodal, CLIP encode simultanément les descriptions textuelles et les noms de classes, permettant une comparaison directe et précise. Contrairement à une simple extraction

du mot-classe à partir de la sortie d'un modèle de langage (LLM), CLIP présente plusieurs avantages majeurs :

1. **Alignement précis de l'espace textuel et visuel** : CLIP encode les descriptions générées et les noms des classes dans un espace vectoriel partagé, ce qui permet de comparer facilement leurs représentations numériques.
2. **Robustesse et flexibilité** : Il s'adapte efficacement à des contextes variés, y compris dans des scénarios d'apprentissage incrémental où de nouvelles classes sont ajoutées progressivement.
3. **Réduction des erreurs linguistiques** : En comparant directement les embeddings (représentations numériques) plutôt que les textes bruts, CLIP évite les erreurs qui pourraient découler de descriptions imprécises ou ambiguës générées par le LLM.
4. **Capacité de généralisation** : CLIP peut identifier des classes apprises précédemment, même si elles ne font pas partie de la tâche en cours, ce qui est essentiel dans des environnements où les connaissances doivent être retenues sur le long terme.

VI. Résultats expérimentaux

VI.1 Résultats de notre models

```
95.65 mean:  95.65
96.87 95.83 mean:  96.32
95.35 95.46 96.43 mean:  95.8
94.36 95.22 95.15 95.41 mean:  95.05
95.88 94.84 94.56 92.77 96.08 mean:  94.85
94.55 94.67 94.19 91.06 91.46 96.35 mean:  93.68
91.91 93.44 93.99 92.24 93.67 93.18 97.49 mean:  93.85
93.06 93.48 94.56 91.38 91.93 92.16 95.46 94.69 mean:  93.38
```

Figure 6. Précision du modèle avec 8 tâches

Les résultats affichés représentent les performances du modèle GMM dans le cadre de l'apprentissage par Continual Learning (CIL). Chaque ligne correspond au testing du modèle en ajoutant un task de plus au fur et à mesure. Chaque colonne correspond à la valeur d'accuracy d'une tâche, la valeur mean étant l'accuracy pour toutes les tâches.

Globalement, le modèle montre de bonnes performances lorsqu'on lui ajoute les tâches progressivement, une par une. Cependant, on observe une **légère baisse** de **l'accuracy**, ce qui est attribué au phénomène d'**oubli catastrophique**, où le modèle a tendance à oublier des connaissances acquises précédemment lorsqu'il apprend de nouvelles tâches.

Ceci peut être dû à la **capacité au LLM à bien décrire l'image donnée** mais aussi à l'entraînement de la **couche linéaire** qui change de valeur de **poids** après chaque entraînement, mais comme ce dernier n'est qu'un "pont" son impact n'est pas important.

Malgré cette diminution, les performances restent excellentes par rapport à d'autres modèles, comme nous le verrons plus en détail par la suite.

VI.2 Comparaison avec les modèle actuels

Il s'agit d'une explication des résultats de l'expérimentation faite par le papier scientifique. En effet, par manque de temps, on n'a pas pu tester notre modèle avec d'autres paramètres, ni tester d'autres modèles de CIL pour les comparer entre eux. Le temps de calcul pour l'entraînement et le testing dure plus de 10 heures. De plus, on a eu à notre disposition le matériel adéquat que très tardivement.

Type	Method	Exemplar	Tiny-ImageNet						ImageNet-R
			5 tasks		10 tasks		20 tasks		10 tasks
			Avg	**Last**	**Avg**	**Last**	**Avg**	**Last**	Last
Conventional	EWC [26]	✗	19.01	6.00	15.82	3.79	12.35	4.73	35.00
	LwF [29]	✗	22.31	7.34	17.34	4.73	12.48	4.26	38.50
	iCaRL [48]	✓	45.95	34.60	43.22	33.22	37.85	27.54	-
	EEIL [7]	✓	47.17	35.12	45.03	34.64	40.41	29.72	-
	UCIR [24]	✓	50.30	39.42	48.58	37.29	42.84	30.85	-
	PASS [89]	✗	49.54	41.64	47.19	39.27	42.01	32.93	-
	DyTox [16]	✓	55.58	47.23	52.26	42.79	46.18	36.21	-
Discriminative PT models	Continual-CLIP[63]	✗	70.49	66.43	70.55	66.43	70.51	66.43	72.00
	L2P [72]	✗	83.53	78.32	76.37	65.78	68.04	52.40	72.92
	L2P [72]	✓	80.24	72.89	80.08	72.61	79.44	70.41	59.78
	DualPrompt [71]	✗	85.15	81.01	81.38	73.73	73.45	60.16	68.82
	DualPrompt [71]	✓	79.92	72.83	79.15	73.21	80.17	71.74	57.02
	CODA-Prompt [55]	✗	**85.91**	**81.36**	82.80	75.28	77.43	66.32	73.88
	Linear Probe	✗	74.38	65.40	69.73	58.31	60.14	49.72	45.17
	Linear Probe	✓	70.10	61.11	69.35	64.19	71.64	70.50	55.72
Generative PT models	Zero-shot	✗	58.16	53.72	58.10	53.72	58.13	53.72	67.38
	GMM (Ours)	✗	83.42	76.98	82.49	76.51	81.70	76.03	80.72
	GMM (Ours)	✓	84.16	78.46	**83.95**	**78.64**	**84.23**	**79.17**	**89.41**

Figure 7. Résultats de comparaison de la méthode GMM avec d'autres méthodes classiques et modèles pré-entraînés discriminatifs (PT) sur Tiny-ImageNet et ImageNet-R sous le cadre classique de CIL.

Ce tableau compare les performances de différentes méthodes d'apprentissage incrémental par classe (CIL) sur deux ensembles de données : Tiny-ImageNet et ImageNet-R. Les résultats sont regroupés en trois grandes catégories : les méthodes conventionnelles, les modèles discriminatifs pré-entraînés et les modèles génératifs pré-entraînés.

Les méthodes conventionnelles, comme EWC ou iCaRL ont des résultats moyens très faibles, avec seulement 12,35 % de précision moyenne pour 20 tâches sur Tiny-ImageNet.

Les modèles discriminatifs pré-entraînés, DualPrompt et CODA-Prompt, qui atteignent 80,17 % et 77,35% de précision moyenne pour 20 tâches sur Tiny-ImageNet. Cela montre leur capacité à bien intégrer de nouvelles classes tout en préservant les connaissances acquises, mais moins performants que le GMM.

Enfin, **les modèles génératifs pré-entraînés**, le modèle proposé dans ce document ("GMM (Ours)"), offrent les meilleurs résultats. Par exemple, GMM atteint84,34 % de précision moyenne pour 20 tâches surTiny-ImageNet et 89,41 % sur ImageNet-R. Ces performances supérieures démontrent l'efficacité des modèles génératifs pour apprendre sans oublier les classes précédentes.

VI.3 Performances sur chaque dataset

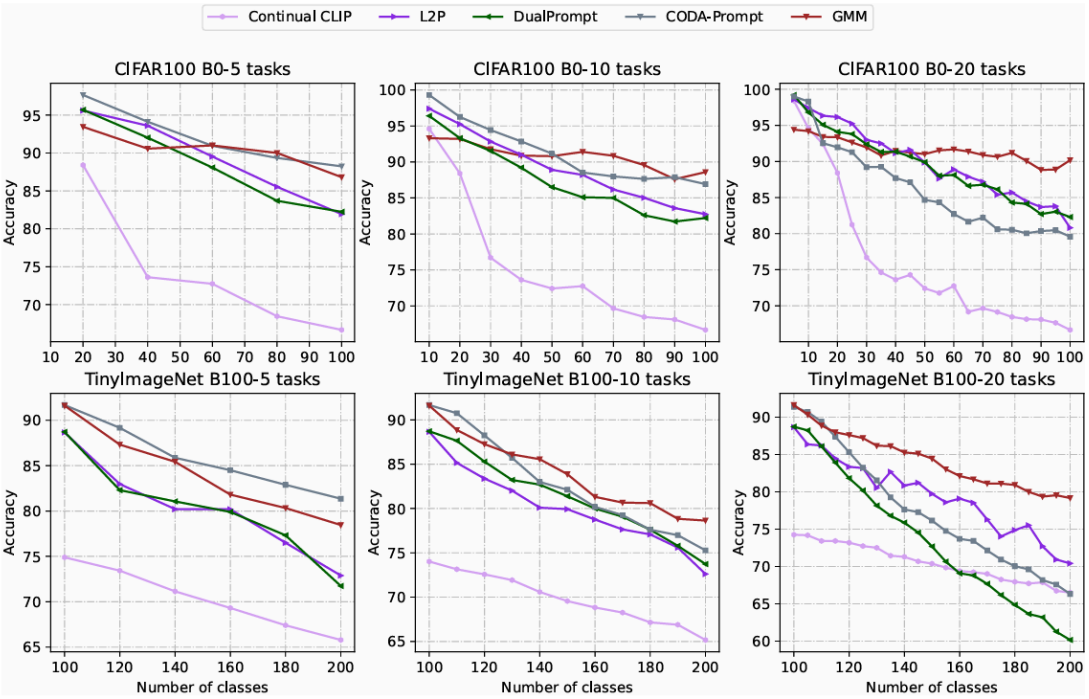

Figure 8. Comparaison de la méthode GMM avec d'autres baselines SOTA sur CIFAR100 et Tiny-ImageNet dans le cadre de l'approche CIL conventionnelle.

Cette figure illustre les performances de plusieurs méthodes d'apprentissage incrémental par classe (CIL) sur les ensembles CIFAR100 et TinyImageNet, avec des scénarios où le nombre de classes et de tâches augmente progressivement (5, 10 et 20 tâches). On observe que, de manière générale, la précision diminue au fur et à mesure que de nouvelles classes sont introduites, mais l'ampleur de cette baisse varie considérablement selon les approches.

Continual CLIP (courbe violette) se révèle peu performant, avec une chute rapide de la précision dès l'ajout de nouvelles classes, traduisant une faible capacité à retenir les connaissances acquises. L2P (courbe rose) fait un peu mieux, mais reste instable, surtout lorsque le nombre de classes devient élevé. Le GMM (courbe rouge), en revanche, surpasse largement toutes les autres méthodes. Il maintient des performances remarquablement stables, même avec un nombre important de classes et de tâches. Là où les autres approches voient leur précision fortement diminuer.

VII. Conclusion

En résumé, le modèle **GMM** (Generative Multi-modal Model) se distingue comme une solution particulièrement innovante pour l'apprentissage incrémental par classe (**CIL**) grâce à plusieurs aspects uniques qui le rendent supérieur aux approches existantes. Contrairement aux méthodes conventionnelles, limitées par leur **incapacité à conserver efficacement les connaissances acquises**, le GMM tire parti des capacités des modèles **génératifs** et **multimodaux**. Cela lui permet d'exceller dans la gestion des tâches successives sans subir l'oubli catastrophique.

L'une des principales forces du GMM réside dans sa capacité à modéliser et à **exploiter les relations complexes entre les données visuelles et textuelles**. En générant des descriptions textuelles précises pour chaque image, puis en utilisant des embeddings partagés pour effectuer la classification, il élimine la nécessité de conserver des exemples explicites des anciennes classes. Cette approche offre une flexibilité unique tout en maintenant une performance constante, comme en témoignent ses excellents résultats sur de grands ensembles de tâches, par exemple 83,95 % sur 20 tâches Tiny-ImageNet ou 89,41 % sur ImageNet-R.

De plus, le GMM surpasse même les modèles discriminatifs pré-entraînés, tels que CODA-Prompt ou DualPrompt, en combinant les avantages des modèles génératifs et des représentations pré-entraînées. Sa capacité à s'adapter aux nouvelles classes sans sacrifier les anciennes est une avancée significative pour le domaine du CIL. Cela en fait une solution non seulement innovante, mais également durable et performante dans des scénarios complexes où l'évolutivité et la précision sont cruciales.

VIII. Références

[1] Generative Multi-modal Models are Good Class-Incremental Learners : Xusheng Cao, Haori Lu, Linlan Huang, Xialei Liu, Ming-Ming Cheng, VCIP, CS, Nankai University NKIARI, Shenzhen Futian

https://openaccess.thecvf.com/content/CVPR2024/papers/Cao_Generative_Multi-modal_Models_are_Good_Class_Incremental_Learners_CVPR_2024_paper.pdf

https://github.com/DoubleClass/GMM

---

https://www.objective.inc/articles/minigpt-4-technical-deep-dive

---

GitHub - openai/CLIP: CLIP (Contrastive Language-Image Pretraining), Predict the most relevant text snippet given an image

---

https://arxiv.org/abs/2304.10592

https://www.researchgate.net/publication/370213476_MiniGPT-4_Enhancing_Vision-Language_Understanding_with_Advanced_Large_Language_Models ch

---

https://github.com/DoubleClass/GMM

---

https://huggingface.co/docs/transformers/main/model_doc/vit

---

https://huggingface.co/docs/transformers/main/model_doc/clip

---

https://huggingface.co/docs/transformers/main/tasks/prompting

---