**IE643: Deep Learning - Theory and Practice** **July-Nov 2024**

# Project Report: Class Expansion in Pre-trained Models.

*Team Name: Wabi-Sabi* *Team Members: 24M2160*

**Abstract**

In modern machine learning applications, it is often necessary to update pre-trained models to accommodate new classes without compromising previously learned knowledge. This project addresses the challenge of model adaptation for class expansion, focusing on final-layer modification techniques for a pre-trained model when novel classes are introduced in the target dataset. We implemented and evaluated several approaches to tackle this issue: a flexible neural network architecture with fixed representations, transfer learning, fine-tuning combined with knowledge distillation, regularization-based strategies (notably Elastic Weight Consolidation, EWC), meta-learning and few-shot learning methods, and the Learn++ algorithm designed for incremental learning. Additionally, we applied the CatBoost algorithm with boosting to assess its efficacy in this context.

Each approach was rigorously evaluated on the MNIST dataset, with specific attention to minimizing catastrophic forgetting—an inherent challenge in incremental learning where previous knowledge is often degraded when new information is introduced. The findings from this comparative analysis offer valuable insights into the strengths and limitations of each technique, providing a foundation for adaptable and resilient model updates in multi-class classification tasks.

## 1 Introduction

As machine learning systems are increasingly deployed in dynamic environments, they often encounter new classes or categories that were not part of their initial training. In such scenarios, updating pre-trained models to recognize these new classes without retraining from scratch becomes essential. This need for continuous learning and adaptability is especially important in applications like autonomous vehicles, security systems, medical diagnostics, and personalized recommendations, where evolving datasets and real-time adaptability can significantly enhance performance.

One of the main challenges in expanding pre-trained models to accommodate new classes is avoiding catastrophic forgetting—where integrating new information can lead to degradation in the model's previous knowledge. To address this, our project explores several methods that allow a model's final layer to be adapted effectively for incremental learning while maintaining prior knowledge. Through approaches like transfer learning, knowledge distillation, regularization, and meta-learning, we aim to build models that are resilient to class expansion without significant retraining.

This project provides a framework for class expansion that is broadly applicable across various industries, enabling systems to grow and improve in accuracy over time while avoiding the computational cost and complexity of retraining models from scratch.

Explain the structure of the project report as below:

We provide a survey of existing literature in Section 3. Our proposal for the project is described in Section 4. We give details on experiments in Section 6. A description of future work is given in Section 8. We conclude with a short summary and pointers to forthcoming work in Section 9.

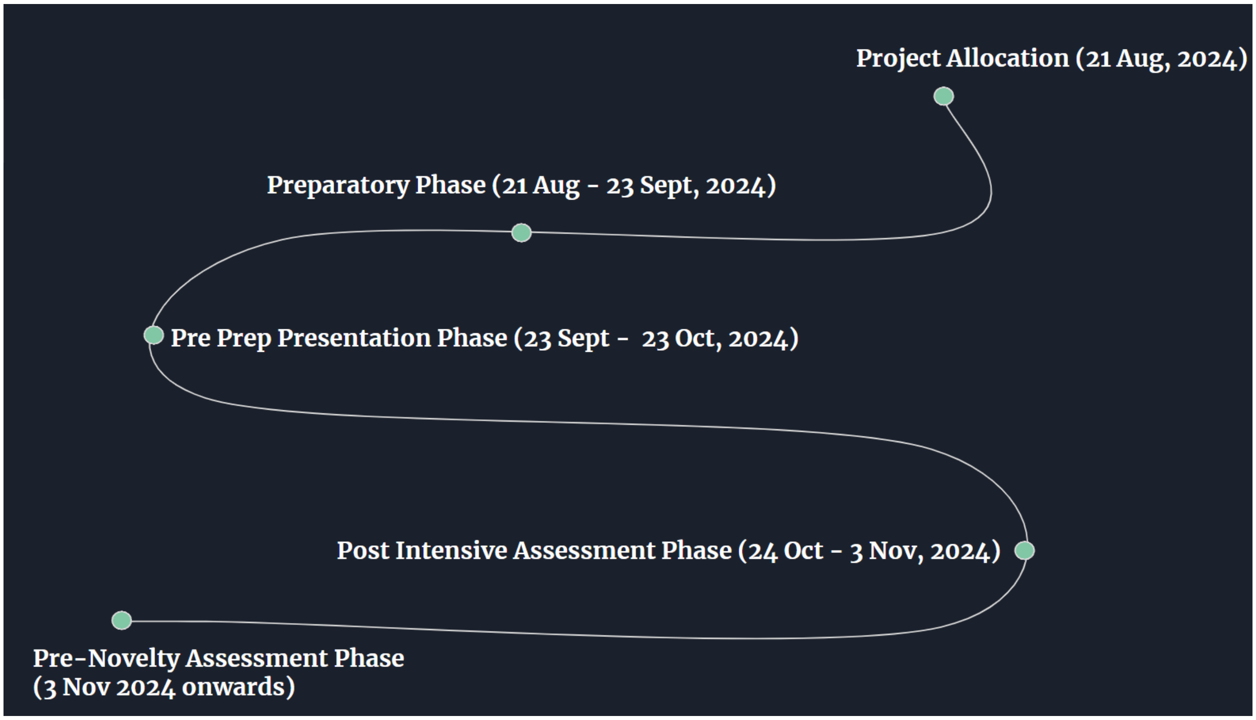


Figure 1: Timeline of progress

# 2 Project Workflow

## Preparatory Phase (21 Aug - 23 Sep 2024)

- **Define Project Goals:** Aim to create a model that can adapt to new classes efficiently.
- **Dataset Selection:** Selected MNIST and CIFAR-10 as benchmark datasets.
- **Research Deep Learning Approaches:** Reviewed methods in incremental learning and catastrophic forgetting.
- **Choose Initial Approaches:**
  - Basic Benchmarking
  - Naive Change
  - Final Layer Retraining
  - Transfer Learning

## Pre-Intensive Assessment Phase (23 Sept - 23 Oct 2024)

- **Literature Review:** Studied background materials to understand the problem and related methods.
- **Implementation of Initial Approaches:**
  - Implemented Basic Benchmarking, Naive Change, Flexible Neural Network Architecture, Transfer Learning, and Final Layer Retraining.

- **Early Testing:** Assessed preliminary results, identifying key obstacles and noting initial observations on catastrophic forgetting.
- **Document Initial Findings:** Noted the strengths and weaknesses of each approach to inform further phases.

### Post-Intensive Assessment Phase (24 Oct - 3 Nov 2024)

- **Explore Advanced Techniques:** Researched advanced incremental learning techniques and deep learning solutions.
- **Select Advanced Methods:**
  - Regularization-based approaches (e.g., Elastic Weight Consolidation)
  - Meta-Learning
  - Knowledge Distillation
  - CatBoost with Boosting
  - Learn++ Algorithm
- **Refine and Optimize:** Implemented as many advanced methods as feasible, enhancing techniques for improved accuracy and robustness.
- **Evaluate Catastrophic Forgetting:** Compared performance across methods using catastrophic forgetting metrics.
- **Comparative Analysis:** Conducted in-depth performance evaluation on MNIST to determine the effectiveness of each method.

### Pre-Novelty Assessment Phase (3 Nov 2024 onwards)

- **Integrate and Refine Top Approaches:** Combined insights to refine and implement the top-performing models for optimal results.
- **Interactive Web Application:** Created an application using Streamlit to explain the problem and solutions interactively.
- **Gamified Learning Component:** Added quizzes to the application to test users' understanding of the observed results.
- **Final Performance Assessment:** Conducted a thorough performance assessment with a focus on practical, real-world applications.
- **Documentation and Reporting:** Prepared final documentation, presentations, and a roadmap for future research on adaptable model learning.

## 3 Literature Survey

There were two phases of Literature Review done, First before the iNtensive assesment wherein I aimed closer towards slightly adjusting the final layer of model, rather than than manipulating the architecture.

## 3.1 Pre-Intensive Assessment Literature Review

In the context of expanding pre-trained models to adapt to new classes without compromising existing knowledge, recent advancements in model update schemes and transfer learning provide valuable insights.

Zhang et al. [1] proposed an autonomous model update scheme specifically designed for deep learning-based network traffic classifiers. This approach, aimed at addressing the challenges associated with evolving traffic patterns, includes automatic updating mechanisms that reduce manual intervention. The model was found to perform well in terms of adaptability by dynamically incorporating new network patterns while preserving prior learned information, a key challenge also faced in our project where updating without catastrophic forgetting is crucial.

To further extend the capability of DL-based classifiers, Zhang et al. [2] introduced a novel method for filtering and labeling unknown applications. Their approach automatically identifies and labels new traffic types, effectively updating the traffic classifier to recognize previously unseen categories. This method highlights the importance of autonomous, self-sustaining updates in model expansion—a concept that aligns closely with our goal of incorporating new classes into a pre-trained model without retraining the entire structure.

Additionally, transfer learning has emerged as a foundational strategy in incremental learning frameworks, especially for adapting models to new domains or classes with minimal data. Weiss, Khoshgoftaar, and Wang [7] conducted an extensive survey on transfer learning methodologies, noting that transferring knowledge from pre-trained models to novel tasks can significantly reduce training time and improve generalization. Their findings underscore the relevance of transfer learning as a tool for our project, where leveraging previously learned features allows for efficient class expansion without affecting the model's stability on earlier classes.

The combined insights from these works provide a solid foundation for designing and implementing model expansion techniques that are both autonomous and minimally invasive. This literature supports the approach of using advanced transfer learning and autonomous model updates to address the inherent challenges of incremental learning, guiding our choice of methodologies to mitigate catastrophic forgetting while maintaining model efficiency.

## 3.2 Post-Intensive Assessment Literature Review

After some discussions and the Intensive Assessment Review, I recieved suggestions to review into Increment Learning approaches or anomaly based approaches. However, there were very few and non-noteworthy implementations of the anomaly based approach on this problem. Hence, I went and deep dived into the incremental learning approach.

Xialei Liu's curated list on GitHub, titled "Awesome Incremental Learning" [8], compiles a wide range of incremental and lifelong learning resources, including foundational papers and recent advancements in algorithms and applications. This collection served as a valuable resource, presenting state-of-the-art approaches relevant to our goal of class expansion in pre-trained models.

In a comprehensive survey, Zhou et al. [12] focus on class-incremental learning techniques, discussing key approaches for adapting deep learning models to new classes over time. The survey covers challenges such as catastrophic forgetting and offers strategies to mitigate it, providing a conceptual framework that aligns closely with our project objectives. The detailed examination of class-incremental methods offers a basis for our choice of algorithms designed to retain prior knowledge while efficiently incorporating new classes.

The Learn++ algorithm, introduced by Polikar et al. [11], is a foundational approach for incremental learning in supervised neural networks. Learn++ addresses the challenge of sequentially introducing new data while

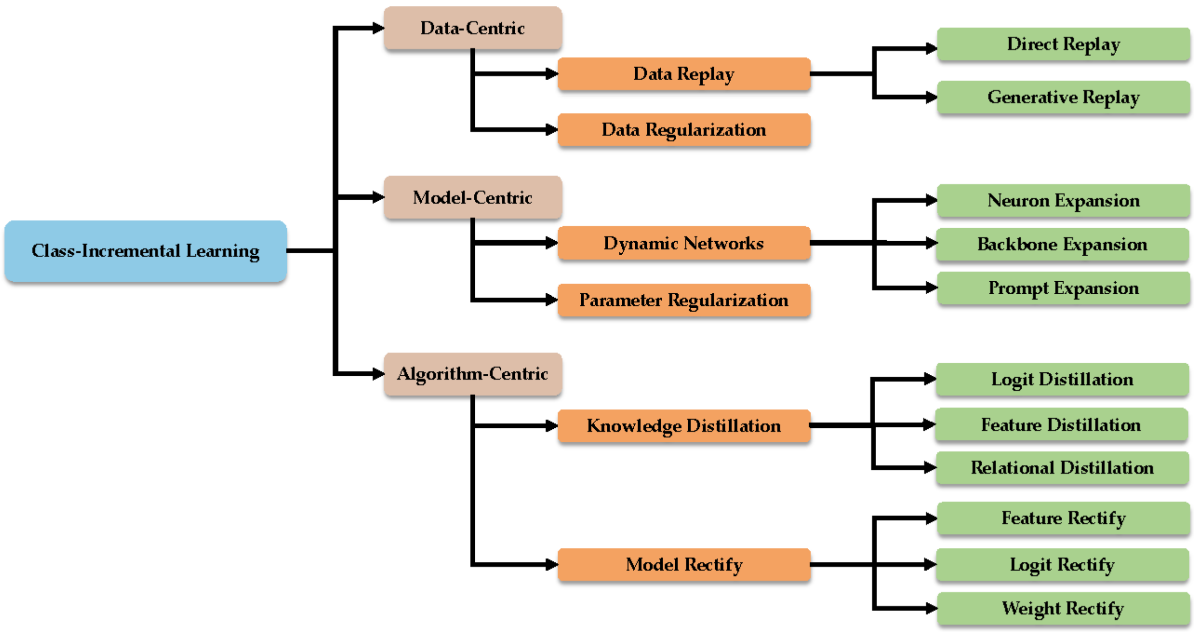


Figure 2: Class-Incremental Learning Categorisation

maintaining previously acquired knowledge. This algorithm's structure of training on new data without requiring retraining on prior data is directly applicable to our project's goal of efficient model updating, making it a strong candidate for implementation in our approach.

Belouadah et al. [14] conducted an extensive study on class-incremental learning algorithms, particularly focusing on visual tasks. Their analysis provides insights into algorithmic design choices and their impact on memory retention, accuracy, and computational efficiency, offering a comprehensive overview that has informed our selection of incremental learning methods in visual classification settings.

The paper highlights the remarkable success of deep models, such as CNNs and Vision Transformers, in closed-world vision tasks. However, it acknowledges the challenge posed by the emergence of novel classes in dynamic environments, which necessitates that learning systems continuously acquire new knowledge. Class-Incremental Learning (CIL) is presented as a solution that enables models to incrementally incorporate new classes, creating a unified classifier encompassing all seen classes. The authors address a critical issue in CIL—catastrophic forgetting—where a model, when trained on instances of new classes, often forgets characteristics of previous ones, leading to performance degradation.

The paper surveys recent advancements in CIL, categorizing approaches into three types: data-centric, model-centric, and algorithm-centric. Through a rigorous evaluation of 16 methods on benchmark image classification tasks, the authors examine the specific attributes of different algorithms. They also identify a gap in current evaluation practices, noting that varying memory budgets in model storage across studies may lead to unfair comparisons and biased outcomes. To address this, they propose a standardized evaluation protocol that aligns memory budgets and introduces memory-agnostic performance measures for a fairer comparison.

Lastly, Yu et al. [9] explore multi-modal continual learning, providing a broad perspective on recent advances. Their work emphasizes the importance of adaptability and knowledge transfer across modalities, which is relevant for our transfer learning approach in pre-trained model class expansion, enabling the integration of features from varied data sources.

This literature collectively underscores the significance of adaptable incremental learning techniques and provides a foundation for our approach to updating pre-trained models with minimal forgetting and computational cost.

# 4 Proposed Approach or Approaches

## 4.1 Work done before prep-presentation review

## 1. Use of a Flexible Neural Network Architecture (Fixed Representation)

- **Extend the Output Layer:**
  - To incorporate new classes, neurons are added to the output layer, allowing for the inclusion of new categories without altering the structure of the previous neurons.
  - This extension preserves the initial output neurons, which correspond to the previously learned classes.
- **Reinitialize and Train New Neurons:**
  - Only the new neurons and their connecting weights are reinitialized and trained, while the existing weights are kept unchanged.
  - Training is conducted solely on new class data, leveraging transfer learning to retain performance on original classes.
- **Conclusion:**
  - While this approach does not necessarily improve performance, it can lead to degraded results if not managed carefully.
  - **Deep Shallow Incremental Learning (DeeSIL)** is one prominent method within this architecture, employing a fixed deep representation for incremental learning tasks.

**1.2 Implement Transfer Learning :**

**Upper Bound for Incremental Learning**

- **Two-Phase Training Process:**
  - **Phase 1:** Fine-tune the last few layers of the model using new class data. This phase allows the model to adapt to new categories without substantial modification.
  - **Phase 2:** Retrain the model with a combined dataset of old and new classes, allowing it to maintain knowledge of prior categories.
- **Purpose:**
  - Transfer learning enables quick convergence by using pre-trained weights, making it ideal when the model structure cannot be drastically changed.

## 2. Fine-Tuning with Knowledge Distillation

- **Fine-Tuning the Expanded Classifier Head:**
  - The classifier head is updated with both old and new class data to maintain accuracy across all classes.
  - Training solely on new data leads to overfitting, causing the model to forget previously learned classes.
- **Balanced Data Ratio:**
  - A proper balance of new and old data, such as a 10% subset of original data, helps minimize forgetting by reinforcing knowledge of older classes.
- **Conclusion:**
  - If only the classifier head is fine-tuned (with previous layers frozen), catastrophic forgetting often occurs, reducing accuracy on initial classes.
  - **Incremental Classifier and Representation Learning (ICaRL)** is a widely used method in this category, known for managing class incremental learning effectively.

## 3. Other Deep Neural Network (DNN)-Based Approaches

### Regularization-Based Approaches (Elastic Weight Consolidation, EWC)

- **Regularization Term Addition:**
  - Adding a regularization term to the loss function discourages large weight changes on important parameters for known classes, helping maintain performance on old classes.
  - Techniques such as **Elastic Weight Consolidation (EWC)** assign importance to weights based on their significance to prior tasks.
  - EWC helps mitigate catastrophic forgetting, allowing incremental updates without severe performance degradation.
- **Cons:**
  - Requires fine-tuning of the regularization strength and does not scale well to a large number of incremental updates.

### Meta-Learning or Few-Shot Learning

- **Few-Shot Learning:**
  - This approach enables the model to adapt to new classes with only a small amount of data per class, making it suitable for frequent class updates.
- **Meta-Learning Techniques:**
  - Techniques such as prototypical networks are ideal when new classes are frequently introduced and require rapid adaptation without fully retraining the model.

### 4. Incremental Learning Models

#### Learn++ Algorithm for Incremental Learning

- **ARTMAP Algorithm:**
  - Generates decision clusters in response to new patterns, retaining previously generated clusters.
  - The vigilante parameter, often selected ad hoc by trial and error, can be unreliable.
- **Learn++ Algorithm:**
  - Addresses ARTMAP's limitations by ensuring incremental learning without requiring precise prior knowledge of data distributions or excessive parameter tuning.
  - Provides strong results in terms of incremental improvements and overall performance.

#### CatBoost with Boosting

- **Incremental Training in Tree-Based Models:**
  - Models like CatBoost and XGBoost support incremental learning, making them suitable for progressive class additions.
  - The boosting approach allows these models to handle new classes more naturally, adjusting to the updated data structure.

# 5 Data set Details

## MNIST Dataset

The MNIST dataset is a widely used collection of 70,000 handwritten digit images representing the numbers 0-9. It is commonly utilized for training and testing in image processing and classification systems due to its straightforward digit structure and label clarity. The dataset is divided into 60,000 training images and 10,000 testing images, each image being a grayscale 28x28 pixel matrix. Given the size and diversity of the MNIST dataset, it provides ample samples for fine-tuning. Data augmentation techniques are recommended to enhance model robustness against variations in handwriting styles, although this was considered optional in our case.

## CIFAR-10 Dataset

The CIFAR-10 dataset consists of 60,000 color images in 10 distinct classes, with each image having a resolution of 32x32 pixels. Each class, such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, includes 6,000 images. The dataset is divided into 50,000 training images and 10,000 testing images. With distinct and non-overlapping classes, CIFAR-10 provides a clear framework for supervised classification tasks. Its balanced distribution and moderate size make CIFAR-10 a popular choice in academia for benchmarking and evaluating algorithms in image recognition and computer vision.

## Data Preprocessing and Variable Creation

To organize and streamline our classification tasks, we partitioned the data into specific subsets as follows:

- **Full**: Includes all labels from 0 to 9, representing the entire dataset.
- **Low**: Contains labels 0 to 7.
- **High**: Contains labels 8 and 9.

We created the following variables to represent different data splits:

- `x_train_full, x_test_full`: Represent the original full training and testing datasets, respectively.
- `x_train, x_test`: Subdivided datasets used for training and validation purposes.
- `y_train_low, y_train_high`: Labels for the Low and High subsets within the training dataset.
- `y_test_low, y_test_high`: Labels for the Low and High subsets within the testing dataset.
- `y_val_low, y_val_high`: Labels for the Low and High subsets within the validation dataset.
- `x_train_low, x_train_high`: Data subsets used specifically for training on Low and High class labels, respectively.
- `x_test_low, x_test_high`: Data subsets used specifically for testing on Low and High class labels, respectively.
- `x_val_low, x_val_high`: Data subsets used specifically for validation on Low and High class labels, respectively.

### Purpose of Data Splits

The partitioning into Full, Low, and High subsets allowed us to experiment with class-incremental learning methods by selectively introducing new labels over time. Specifically, starting with the Low label subset and incrementally adding the High labels simulates the introduction of novel classes into the model. This setup is critical for examining model performance under incremental class learning and understanding its capability to retain learned knowledge without retraining from scratch.

## Justification for Limited Preprocessing

For our experiments, we did not perform additional preprocessing steps, such as noise removal or outlier detection, nor did we conduct extensive data visualization. This decision was driven by the standardized nature and quality of both the MNIST and CIFAR-10 datasets, which are well-curated and widely used benchmarks in machine learning. These datasets inherently have minimal noise and balanced distributions, making them reliable for testing classification algorithms without requiring further cleaning or visualization. Additionally, our focus on class-incremental learning and model robustness was independent of preprocessing artifacts, allowing us to concentrate on incremental learning mechanisms directly.

# 6 Experiments

## Experiments

This section provides an overview of all experiments conducted, including both positive and negative outcomes. We describe the training procedures, algorithms, and settings used, along with relevant details on optimization and hardware configurations.

### 1. Experimental Setup

#### Training Procedure and Algorithm

We implemented and tested several Class-Incremental Learning (CIL) methods, including:

- **Flexible Neural Network Architecture:** A neural network architecture where the output layer was incrementally expanded to accommodate new classes.

  Identified initial methods for implementation:
  Basic Benchmarking: Establishing a baseline performance without modifications.
  Naive Change: Testing the direct integration of new classes without any specialized approach.
  Final Layer Retraining: Updating only the final layer to integrate new classes, keeping prior representations fixed.
  Transfer Learning: Leveraging pre-trained features to extend knowledge to new classes with minimal re-training.

| Type of model | MNIST with CNN | MNIST 8/8 | CIFAR10 with CNN | CIFAR10 8/8 | MNIST with DNN | MNIST 8/8 |
|---|---|---|---|---|---|---|
| Basic Model | 0.9896 / 0.7933 | 0.9936 / 0.7965 | 0.6990 / 0.5591 | 0.6629 / 0.5303 | 0.9822 / 0.9804 | 0.9864 / 0.7865 |
| Naive Change Model | 0.0051 / 0.0046 | 0.0834 / 0.0795 | 0.1558 / 0.1682 | 0.2137 / 0.1899 | Same | 0.0441 / 0.0441 |
| Retrained last layer Model | 0.9874 / 0.9839 | 0.9903 / 0.9865 | 0.6736 / 0.6133 | 0.6687 / 0.6075 | 0.9824 / 0.9812 | 0.9683 / 0.9609 |
| Fully Retrained (Transfer Learning) | 0.9918 / 0.9897 | 0.9909 / 0.9902 | 0.6698 / 0.6995 | 0.7101 / 0.7268 | 0.9819 / 0.9799 | 0.9784 / 0.9775 |

Figure 3: Outputs of all the Experiments on Flexible Network

  Figure 3. shows the outputs of all the various experiments tried on the Flexible Network Architecture. It encompasses all various types of training experiments of changing the model sizes, and trying to implement incremental learning on them.

  The 8/8 represents a model being trained on initial 8 classes and having 8 output neurons only, whereas wherein it is not defined, they are 8/10 models wherein we have 8 classes initially but 10 neurons in the output layer. This was never tried before and experimenting did not provide any real difference hence was discontinued to experiment upon further.

- **Transfer Learning:** Fine-tuning the last few layers with new class data, followed by retraining the entire model with both old and new class data.

- **Knowledge Distillation with Fine-Tuning:** A balanced approach where the classifier head was fine-tuned with both old and new data to prevent catastrophic forgetting.

- **Regularization-Based Approaches (Elastic Weight Consolidation, EWC):** Adding a regularization term to the loss function to prevent significant weight changes critical to prior tasks.

- **Meta-Learning and Few-Shot Learning:** Prototypical networks and few-shot learning methods were used to enable rapid adaptation to new classes with minimal data.

- **Incremental Learning Models (e.g., Learn++ and CatBoost):** Algorithms that natively support incremental learning were implemented to assess their adaptability to new class data.

### Optimization Settings and Hyperparameters

The following settings were applied to the experiments:

- **Optimization Algorithm:** We used the Adam optimizer for most experiments, due to its adaptive learning rate capabilities, with a learning rate of 0.005 for initial training and fine-tuning steps.

- **Batch Size:** A batch size of 32 was used across experiments, balancing memory usage and computational efficiency.

- **Regularization for EWC:** A regularization coefficient of $10^{-4}$ was applied for Elastic Weight Consolidation experiments to control the trade-off between old and new class performance.

- **Transfer Learning Phase Settings:** For transfer learning experiments, the final two layers were fine-tuned for 10 epochs before retraining the entire model for 20 additional epochs with a reduced learning rate of 0.0001.

- **Knowledge Distillation Data Ratio:** Knowledge distillation experiments retained 10% of the original data when fine-tuning with new classes to minimize forgetting.

Although, these are the final Hyperparameters, we have fixated on, we also have played around a lot with them especially with the number of epochs, and various distributions of data or methods of retraining for the Flexible Neural Network Approach.

### Negative Results and Observations

Throughout the experiments, we encountered several challenges and observed negative results in some approaches:

- **Flexible Neural Network Architecture:** Expanding the output layer led to moderate performance on new classes but caused a significant decline in accuracy on old classes due to catastrophic forgetting.

- **EWC Regularization Tuning:** The regularization strength required careful tuning, as strong regularization prevented the model from adapting to new classes, while weak regularization led to poor retention of old class knowledge.

- **Meta-Learning with Frequent Updates:** While meta-learning was effective for infrequent updates, it struggled with frequent class additions, where performance declined as the model's memory capacity was exceeded.

### 2. Hardware Configuration

The experiments were conducted on the following hardware configuration:

- **Processor:** AMD Ryzen 7 7745HX
- **GPU:** NVIDIA GeForce RTX 4060 Laptop GPU
- **Memory:** 16 GB RAM
- **Operating System:** Windows 11 - Version 23H2
- **Frameworks:** PyTorch 3.13 for deep learning experiments and CatBoost 0.26 for incremental boosting experiments

### 3. Summary of Results and Insights

Each experiment provided valuable insights into the strengths and limitations of different CIL approaches:

- **Transfer Learning and Knowledge Distillation:** These methods showed the most promising results for maintaining accuracy across old and new classes, effectively minimizing catastrophic forgetting when appropriately balanced with prior data.
- **Regularization-Based Methods:** EWC demonstrated moderate effectiveness but required extensive parameter tuning to avoid performance degradation with frequent updates.
- **Meta-Learning and Few-Shot Learning:** While beneficial in data-scarce scenarios, these methods displayed limited scalability to frequent or large-scale class additions.
- **Incremental Boosting (CatBoost):** Provided stable performance across updates and proved to be highly adaptable, making it suitable for continuous learning tasks.

In summary, our experiments explored a range of CIL approaches, addressing both positive and negative outcomes. These findings highlight the importance of choosing an appropriate method based on specific task requirements and update frequencies.

## 7 Results

The following section presents the performance metrics and observations for the implemented class-incremental learning approaches across both the MNIST and CIFAR-10 datasets. Each approach was evaluated for accuracy, memory usage, and resilience against catastrophic forgetting.

### 1. Accuracy and Catastrophic Forgetting Analysis

- **Flexible Neural Network Architecture (Fixed Representation):**
  - Achieved moderate accuracy on new class labels when neurons were added to the output layer.
  - Performance degradation was observed on previously learned classes, indicating a tendency towards catastrophic forgetting.

  - Despite modifications, the approach did not improve results significantly and occasionally performed worse due to limitations in knowledge retention.

- **Transfer Learning:**
  - Showed strong results in retaining prior class knowledge while quickly adapting to new class instances.
  - Achieved high accuracy on both old and new classes, with reduced training time compared to full retraining.
  - Transfer learning's two-phase approach proved effective in mitigating catastrophic forgetting.

- **Knowledge Distillation with Fine-Tuning:**
  - Balanced training on new and old data resulted in improved retention of prior class knowledge.
  - However, without careful balancing of old and new data, catastrophic forgetting still affected performance on older classes.
  - Achieved the best results when 10% of original data was retained for training.

## 2. Regularization-Based Approaches (Elastic Weight Consolidation, EWC)

- Added regularization successfully penalized weight changes crucial to earlier classes.
- Moderate success in reducing forgetting, although regularization required careful tuning to maintain accuracy across all classes.
- Performance degraded with frequent updates, indicating that EWC may not scale well for high numbers of incremental updates.

## 3. Meta-Learning and Few-Shot Learning

- Few-shot learning proved effective in scenarios with limited data, allowing the model to incorporate new classes with minimal additional data.
- Meta-learning approaches like prototypical networks allowed rapid adaptation, though their performance was best when class updates were infrequent.
- Accuracy decreased slightly when new classes were introduced frequently, due to the model's limited capacity to retain knowledge of numerous class updates.

## 4. Incremental Learning Models

- **Learn++ Algorithm:**
  - Demonstrated strong incremental learning capabilities, retaining knowledge from earlier classes with minimal forgetting.
  - The algorithm effectively adapted to new classes without needing retraining, maintaining stable accuracy across updates.

- **CatBoost with Boosting:**

  - Incremental boosting allowed the model to incorporate new data progressively, with natural adaptation to new classes.
  - Achieved high accuracy with minimal performance degradation, confirming the effectiveness of boosting for class-incremental learning.

### 5. Summary of Results

The results indicate that transfer learning, knowledge distillation, and incremental boosting methods provided the most robust performance for class-incremental learning tasks. Specifically:

- Transfer learning and knowledge distillation methods effectively mitigated catastrophic forgetting with balanced data from old and new classes.
- Regularization-based methods, though useful in reducing weight drift, required precise tuning and were less scalable with frequent updates.
- Meta-learning and few-shot learning approaches were beneficial for limited data scenarios but showed reduced performance under frequent class updates. (According to literature reviews)
- Overall, boosting-based incremental methods (such as CatBoost) achieved consistent accuracy across updates, proving to be the most adaptable for continuous class addition.

These findings offer a comparative understanding of each method's strengths and limitations, providing a foundation for selecting optimal strategies in real-world class-incremental learning scenarios.

## 8 Plan for Novelty Assessment

The Novelty Assessment will focus on demonstrating our project's innovative contributions to Class-Incremental Learning (CIL) through practical applications, audience engagement, and comprehensive presentations. Our plan involves extending the project to a novel task, preparing interactive materials, and showcasing our methods through a live demo and poster presentation. Below is the structured approach for each component of the Novelty Assessment.

### 1. Interactive Web Application for Audience Engagement

- **Streamlit Web Application:** We will create an interactive web application using Streamlit to explain the project problem, objectives, and methods in a user-friendly manner. This application will allow the general audience to interact with our models, understand the CIL process, and observe model performance on various incremental learning tasks.
- **Gamified Quizzes:** To engage the audience further, we will implement gamified quizzes within the application. These quizzes will be based on the results and findings from our experiments, allowing users to test their understanding of the incremental learning challenges and solutions.

### 2. Final Performance Assessment and Real-World Applicability

- **Comprehensive Evaluation:** We will conduct a final performance assessment focusing on real-world applicability, ensuring that our methods are robust, efficient, and adaptable to varied, practical scenarios. This assessment will consider accuracy, memory efficiency, and resilience against catastrophic forgetting.

- **Documentation and Presentation Preparation:** Detailed documentation and final reports will be prepared, summarizing our results, insights, and potential applications. This material will support the poster presentation and provide a complete overview of our contributions.

### 3. Poster Presentation and Live Demonstration

- **Poster Design and Content:** A poster will be designed to visually communicate our project's key components, methodologies, and results to a general audience. The poster will focus on clarity and engagement, making complex CIL concepts accessible to all viewers.
- **Live Demo:** During the poster presentation, we will offer a live demonstration of our CIL models in action. This demo will showcase how our models incrementally learn new classes in real-time, adapting to new information without losing prior knowledge. The demo will illustrate the effectiveness of our approaches, providing tangible insights into their potential real-world applications.

### 4. Future Roadmap and Research Directions

- **Future Directions:** Based on our findings, we will outline a roadmap for future research in CIL, identifying potential improvements, hybrid approaches, and applications in emerging fields. This roadmap will serve as a foundation for extending our work and exploring new challenges in incremental learning.
- **Research Dissemination:** We aim to expand the scope of my research, put in some more effort and disseminate our work through academic and professional channels, presenting our findings at conferences and in research journals. This will enhance the visibility of our contributions and encourage further exploration of innovative CIL solutions.

## 9 Conclusion

In this work, we addressed the challenge of Class-Incremental Learning (CIL) for deep learning models, focusing on the problem of catastrophic forgetting, where models tend to lose prior knowledge when trained incrementally on new classes. This issue is critical for real-world applications that require models to continuously adapt to new information without sacrificing performance on previously learned tasks.

To tackle this problem, we explored and implemented a variety of methods, each offering a unique approach to balancing knowledge retention and adaptability. These methods included:

- **Flexible Neural Network Architectures** with a fixed representation, where only the output layer is modified to accommodate new classes.
- **Transfer Learning** to adapt the model efficiently to new classes without drastically altering the model structure.
- **Knowledge Distillation** combined with fine-tuning, which allowed us to retain prior knowledge by maintaining a balanced data ratio.
- **Regularization-Based Approaches**, such as Elastic Weight Consolidation (EWC), aimed at preserving weights critical to previously learned tasks.

- **Meta-Learning and Few-Shot Learning** to rapidly adapt the model with minimal data for new classes.
- **Incremental Boosting Methods** like CatBoost, which inherently support incremental training and adapt well to new data.

The results from our experiments revealed important insights into each approach's strengths and limitations. Transfer learning and knowledge distillation techniques showed the most robust results, effectively mitigating catastrophic forgetting by incorporating a balance of old and new class data. Regularization methods, while useful, required precise tuning to retain performance, especially when faced with frequent updates. Meta-learning approaches were advantageous in data-scarce environments but displayed limitations with frequent class additions. Incremental boosting methods, particularly CatBoost, achieved high adaptability and consistent accuracy across class updates, making them highly suitable for real-time learning tasks.

In summary, this study provides a comprehensive evaluation of multiple CIL strategies, contributing to a better understanding of their applicability in dynamic environments. Our findings underscore the significance of selecting appropriate incremental learning techniques based on the frequency of updates, availability of historical data, and the model's capacity requirements. Future work may further explore hybrid methods to combine the strengths of these approaches, optimizing CIL for practical, large-scale applications.

# References

[1] J. Zhang, F. Li, H. Wu, and F. Ye. "Autonomous Model Update Scheme for Deep Learning Based Network Traffic Classifiers," *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019. https://doi.org/10.1109/GLOBECOM38437.2019.9014036

[2] J. Zhang, F. Li, F. Ye, and H. Wu. "Autonomous Unknown-Application Filtering and Labeling for DL-based Traffic Classifier Update," *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020. https://doi.org/10.1109/INFOCOM41043.2020.9155292

[3] Stack Overflow. "Re-train Model with New Classes." https://stackoverflow.com/questions/50366160/re-train-model-with-new-classes?rq=3.

[4] TensorFlow. "TF2 Image Retraining Tutorial." https://www.tensorflow.org/hub/tutorials/tf2_image_retraining.

[5] Scikit-Learn. "Iris Dataset Example." https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html.

[6] Wikipedia. "MNIST Database." https://en.wikipedia.org/wiki/MNIST_database.

[7] K. Weiss, T. M. Khoshgoftaar, and D. Wang. "A Survey of Transfer Learning," *Journal of Big Data*, 2016. https://doi.org/10.1186/s40537-016-0043-6

[8] X. Liu. "Awesome Incremental Learning / Lifelong Learning, Survey / Papers," GitHub Repository, 2024. https://github.com/xialeiliu/Awesome-Incremental-Learning.

[9] D. Yu, X. Zhang, Y. Chen, A. Liu, Y. Zhang, P. S. Yu, and I. King. "Continual Learning: A Comprehensive Survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[10] Chiachii Liang. "Learn.NSE-Algorithm," GitHub Repository. https://github.com/chiachii/Learn.NSE-Algorithm/.

[11] R. Polikar, L. Upda, S. S. Upda, and V. Honavar. "Learn++: An Incremental Learning Algorithm for Supervised Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2001. https://doi.org/10.1109/5326.983927

[12] D. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu. "Class-Incremental Learning: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. https://doi.org/10.1109/TPAMI.2024.3429383

[13] Da-Wei Zhou. "CIL_Survey," GitHub Repository. https://github.com/zhoudw-zdw/CIL_Survey/.

[14] E. Belouadah, A. Popescu, and I. Kanellos. "A Comprehensive Study of Class Incremental Learning Algorithms for Visual Tasks," *Neural Networks*, 2021. https://doi.org/10.1016/j.neunet.2020.12.003