Continual Learning with Multimodal Concepts

Developing deep neural networks that can continuously learn new information without forgetting past knowledge, while also providing human-understandable explanations, is a critical challenge for real-world applications. This project aims to explore the intersection of continual learning and model interpretability, addressing the need for adaptable and transparent AI models in high-stakes domains like healthcare and autonomous systems.

Introduction / Background

AI models have become prominent in recent developments but are often regarded as black-box systems, making it challenging to understand their decision-making processes [1]. This has led to a growing emphasis on interpretability, especially in high-stakes fields such as healthcare. Recent advancements in explainable AI (XAI), including Concept Bottleneck Models (CBMs) [2, 3], have sought to create associations between human-understandable concepts and model outputs, enhancing interpretability. This project aims to develop interpretable AI models that provide insights into the underlying factors influencing disease detection using the HAM10000 dataset [5]. The latter is vital for dermatological research, showcasing deep learning's effectiveness in skin cancer classification and the need for interpretability in clinical settings.

Dataset Description

This project utilizes a medical imaging dataset: HAM10000. The HAM10000 dataset consists of 10,015 dermatoscopic images, each labeled with one of seven skin lesion conditions. These images will aid in developing an interpretable AI model that generates concepts based on visual features to explain skin condition predictions. We will enhance interpretability by generating concepts related to lung opacity using a language model like GPT.

Problem Definition

Deep neural networks face two critical challenges in real-world applications: the inability to learn continuously without forgetting past knowledge and the lack of interpretability in their decision-making processes. While there has been progress in continual learning and model interpretability,

the intersection of these areas remains underexplored. Our goal is to develop models that learn new information over time while retaining past knowledge and providing human-understandable, text-based explanations. This will lead to adaptable and transparent models, which are essential in healthcare and autonomous systems where trust and safety are paramount.

Motivation

The black-box nature of neural network models motivates this project. As AI systems in healthcare and autonomous fields become more prevalent, understanding these models is crucial. Trust in AI relies on clear explanations for decisions, especially in medicine, where professionals depend on AI for accurate diagnoses. Additionally, AI models must adapt over time to enhance reliability. Our ultimate goal is to develop AI systems that foster meaningful insights and collaboration between machines and humans in critical settings.

Methods

Data Preprocessing

The data preprocessing involved the following steps:

- **Data Collection:** The HAM-10000 dataset was downloaded and organized into seven separate folders, each representing a specific class based on the image-to-class mapping provided in the CSV file.
- **Initial Data Analysis:** An analysis of the dataset showed it was imbalanced, with the following label distribution:

Class Label	akiec	bcc	bkl	df	mel	nv	vasc
Initial Count	327	514	1099	115	1113	6705	142

- **Data Augmentation:** To address the class imbalance, augmentation techniques such as cropping, rotation, and vertical flipping were applied. This process ensured that each class contained 500 images, resulting in a total of 3,500 images. Due to compute limitations, the model was run on this set of 3,500 images.
- **Data Splitting:** The augmented dataset was divided into training, validation, and test sets using a 70/20/10 split and stored in .p files.
- **Concept Generation:** For concept generation, 25 images from each class were processed through the GPT-4.0 model, producing 50 distinct concepts per class. These concepts were stored in a JSON file and split into 35 for training and 15 for testing.

ML Algorithms/Models Implemented

Unsupervised Learning

K-Means Clustering

The K-Means algorithm is applied with multiple values of `n_clusters`, starting from 2, 4, 6, and 7. For each value of `K`, the algorithm attempts to classify the data into that number of clusters, which may correspond to different skin cancer types. After fitting the model for each `K`, cluster assignments for each image are stored. This process can be viewed as an attempt to simulate continual learning in clustering, where the model adapts and evolves by increasing the number of clusters progressively. The average precision scores for each `K` value are calculated to assess the purity of clusters at each step, providing insight into how the clustering evolves with each increase in `K`.

Supervised Learning

Concept Bottleneck Model (CBM)

The Concept Bottleneck Model (CBM) introduces interpretability into image classification by utilizing human-specified concepts as intermediaries in the prediction process. The model follows a structured pipeline:

1. **Feature Extraction:** CBM employs traditional feature encoders such as resnets as the backbone to extract meaningful visual features from images. These features are aligned to an intermediate output which represent human-interpretable concepts.
2. **Concept Prediction:** A linear layer is used to transform the extracted features into probabilities for each concept, enabling the model to reason through these interpretable concepts.
3. **Classification:** The final class prediction is made using the probabilities of the predicted concepts. This two-step process ensures that the model's decisions are explicitly tied to the intermediate concepts.

CBM’s training involves two stages:

1. The first stage focuses on aligning the image features with human-defined concepts using a loss function designed to maximize concept accuracy.
2. The second stage optimizes class prediction accuracy based on these intermediate concepts, ensuring robust performance.

During training, we do a joint training of the feature extractors and the concept layer. This approach ensures the model learns a good mapping of features and concepts while learning to map concepts effectively to the target classes.

Concept-Guided Visual Classifier (CGVC)

The Concept-Guided Visual Classifier (CGVC) model employs large language models to autonomously generate interpretable concepts for image classification, eliminating the need for manual annotations. The pipeline of CGVC comprises several crucial components:

1. **Concept Generation:** As part of the data preprocessing, we utilize ChatGPT-4.0 to automatically generate a pool of candidate concepts for each image class.
2. **Concept Selection:** From the generated candidate pool, we select a subset of concepts for each class using a submodular optimization approach. The objective function balances two essential criteria:
    - **Discriminability:** We compute a discriminability score for each concept based on its alignment with the images of different classes. Concepts that exhibit strong alignment with a specific class, while differing from others, are prioritized.
    - **Coverage:** We apply a facility location function to ensure that the selected concepts cover a broad range of unique aspects within each class.
3. **Concept Bottleneck Construction:** We employ CLIP (Contrastive Language-Image Pre-training) to embed both the selected textual concepts and the images into a shared feature space. The concept bottleneck layer computes similarity scores between each image and the corresponding selected concepts.
4. **Classification:** A linear layer is used to map the concept similarity scores to final class predictions. The weights for this layer are initialized based on the concepts selected for each class, providing a strong prior to the classification process.
5. **Training:** The model is trained using 3500 images (500 per class) with cross-entropy loss, while the CLIP encoders remain frozen during the training process. The linear classification layer is fine-tuned to optimize the model's accuracy.

Results and Discussion

Quantitative Metrics

- **Final Average Accuracy (FAA):** Measures overall classification accuracy after all tasks.
- **Average Forgetting (AF):** Indicates how much the model forgets earlier tasks when learning new ones.

K-Means Clustering Results with Forgetting

K (Number of Clusters)	Final Average Accuracy	Average Forgetting
2	0.6691	-
4	0.6450	0.0241
6	0.6448	0.0002
7	0.6283	0.0165

CBM

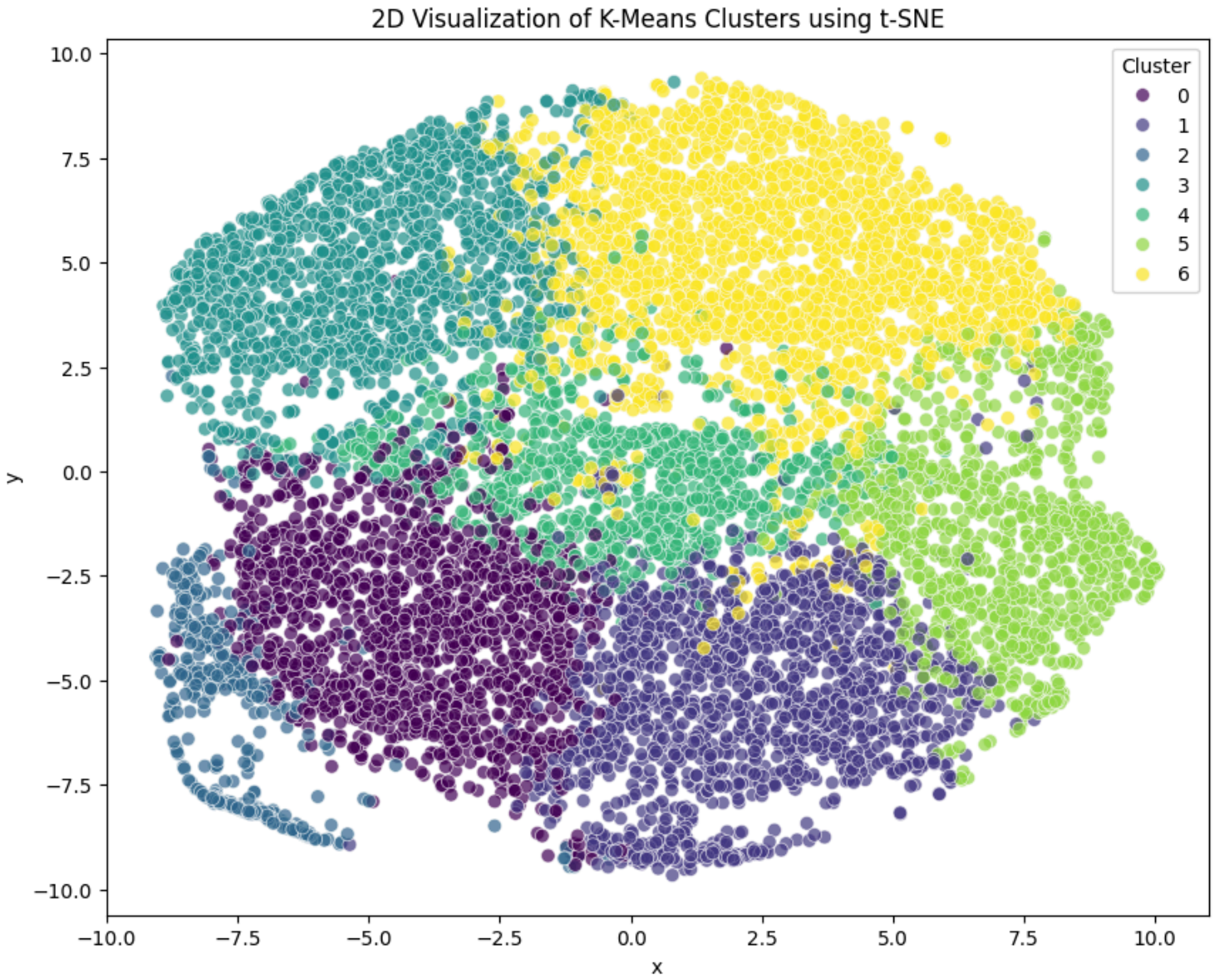
Experiment	Final Average Accuracy (FAA)	Average Forgetting
Exp 1	0.50230102380	-
Exp 2	0.41340288900	0.75411123850
Exp 3	0.36200123345	0.82013572833
Exp 4	0.30766666667	0.81890458075

CGVC

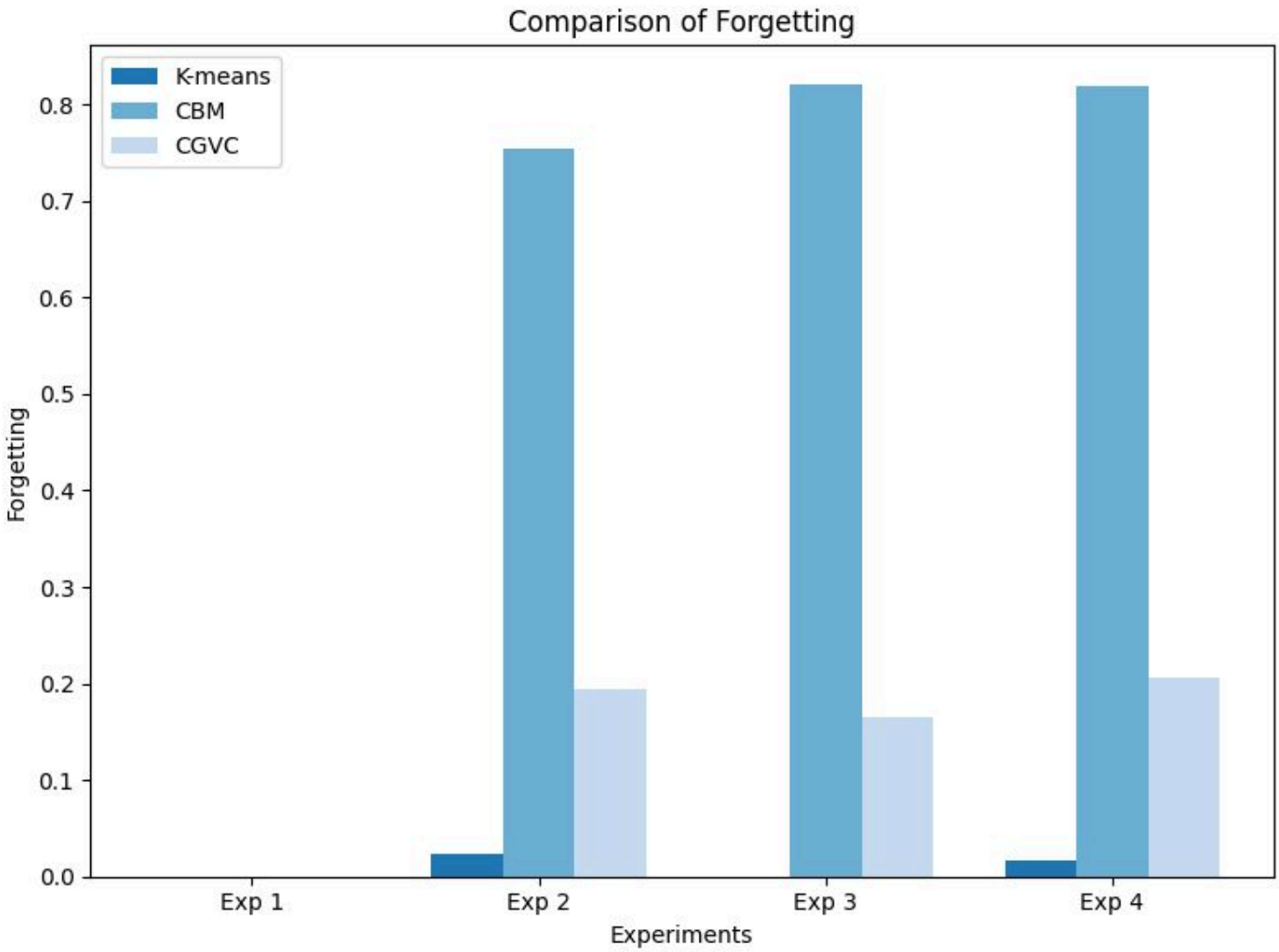
Experiment	Final Average Accuracy (FAA)	Average Forgetting
Exp 1	0.74000000954	0
Exp 2	0.54500001669	0.19499999285
Exp 3	0.43999999762	0.16500001907
Exp 4	0.42381848818	0.20618150944

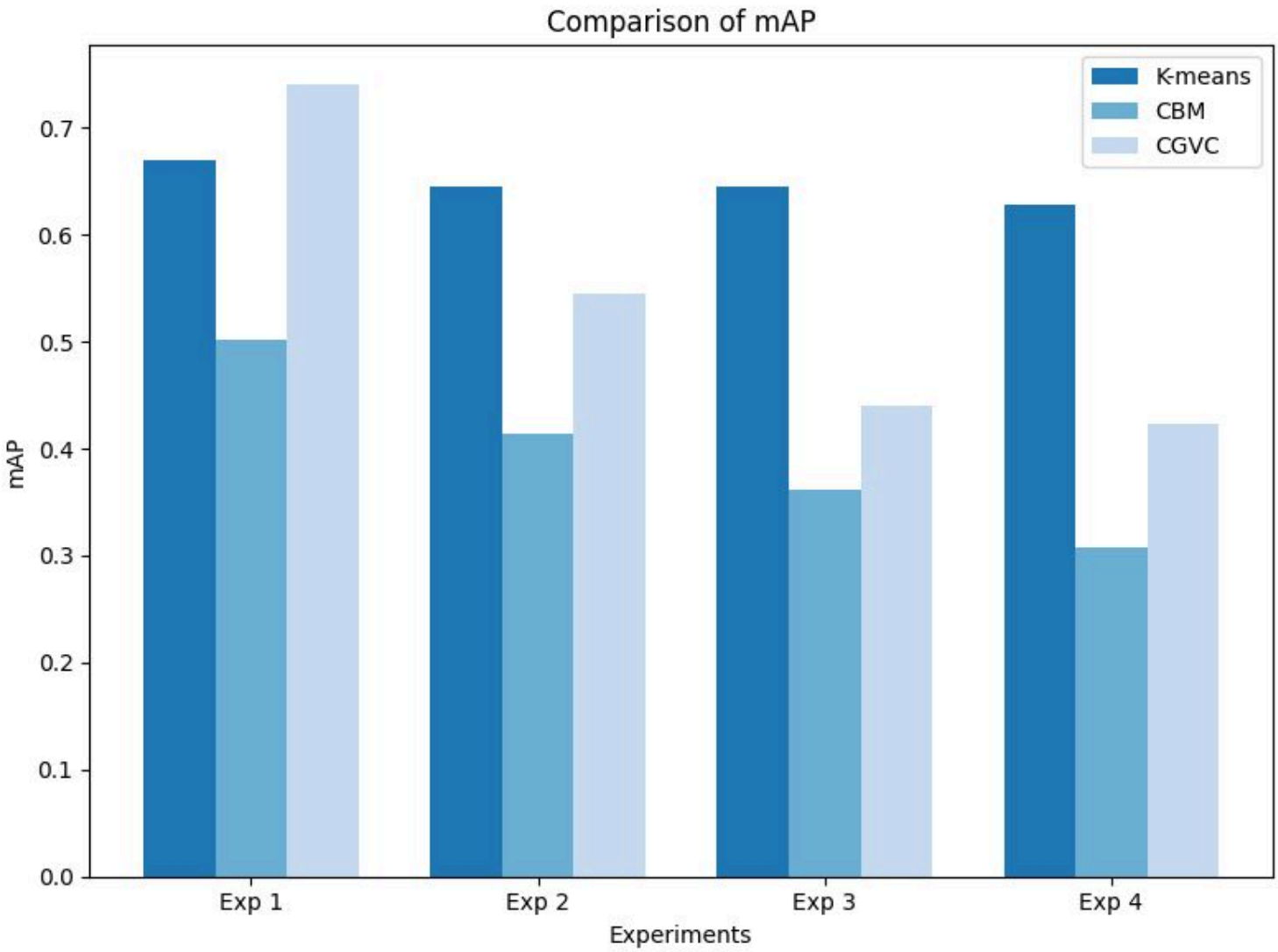
Visualizations

K-Means Clustering



Comparison of all 3 models:


Comparison of Forgetting
K-means
CBM
CGVC
Forgetting
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Exp 1
Exp 2
Exp 3
Exp 4
Experiments



Results Analysis

K-Means Clustering

The K-Means clustering algorithm was evaluated with different values of **k** (number of clusters), and the visualizations using t-SNE provide insights into the clustering performance:

- **K = 2**: The data is divided into two large clusters, showing a clear separation. However, the average clustering precision is **0.6691**, suggesting that this configuration may be too simplistic.
- **K = 4**: With four clusters, the t-SNE visualization reveals more structure in the data, but the average clustering precision drops slightly to **0.6450** with a forgetting rate of **0.0241**.
- **K = 6**: This configuration shows a more complex segmentation, with six distinct clusters. The average clustering precision is **0.6448**, and the forgetting rate is minimal at **0.0002**, indicating that this value balances precision and granularity well.

- **K = 7**: Increasing to seven clusters results in some overlap between clusters, with an average precision of **0.6283** and a forgetting rate of **0.0165**, suggesting diminishing returns beyond k = 6.

Overall, **k = 6** provides the most balanced clustering performance with minimal overlap and high precision.

The results for K-means are higher than our model since we treat the image features as vectors upon which we perform the clustering; this method may be post-hoc interpretable but lacks a clear understanding of fine-grained features of the image. This is tackled in our method CGVC which helps us to correlate the features to an understandable human term (aka concept). Another reason why the results are higher than supervised methods is because we let the model run till convergence which takes a long time to train. Furthermore, the results do not reflect the ability of the model to discover emergent behaviour which would be possible only by using large transformer-based models which can serve crucial for learning cross-class relationships (learn to map old concepts to new classes and new classes with old concepts).

CBM

In these experiments, we evaluated the performance of the Concept Bottleneck Model (CBM) across four trials, varying the number of concepts used while keeping the exemplar size fixed at 100. The two primary metrics analyzed were the final average accuracy (*mAP*) and forgetting, which measures the model's ability to retain knowledge from previous tasks as new ones are learned.

mAP Results: The CBM model showed a decreasing trend in accuracy across experiments. It started with a high mAP of **0.50** in Experiment 1, but the accuracy dropped in subsequent experiments. By Experiment 4, the mAP had decreased to **0.30**. This suggests that while the model performs well in early stages, its accuracy declines as more concepts are added and the model faces additional tasks.

Forgetting Results: The model's forgetting rate was initially **0** in Experiment 1, indicating no forgetting when learning the first task. However, in Experiments 2 and 3, forgetting increased significantly, reaching values of **0.75** and **0.82**, respectively. This shows that CBM struggles with forgetting as more tasks are introduced, and it is more prone to losing information from earlier tasks.

The results from CBM show us that the vanilla model (supervised) struggles in handling forgetting which is a crucial problem in the context of continual learning. It shows that by leveraging transformer models helps to understand multi-modal information better and give rise to emergent behaviours which enables cross concept-class relationships (ability to relate old concepts with new classes and new classes with old concepts).

CGVC

In these experiments, we evaluated a continual learning model's performance across four trials, each varying in the number of concepts used while maintaining an exemplar size of 100 to equally partition past class samples. The primary metric analyzed was the *final average accuracy*, denoted by the first term, and *forgetting*, represented by `cgvc_forgetting`.

The reason for observing high forgetting in our method stems from the size of exemplar that we use. Here we only use 50 samples for the buffer memory which makes it a bit hard to overcome forgetting. In case of k-means we use all the datapoints for each experience which makes it favorable for gaining high accuracy and in turn low forgetting. We will be experimenting our model with increased buffer size to overcome this gap in the next iteration.

Overall, the CGVC model not only offers high prediction accuracy comparable to end-to-end methods but also enhances interpretability by exposing intermediate concept scores and the learned concept-class associations. This transparency provides valuable insight into the model's decision-making process, ensuring that its predictions are both explainable and reliable.

Project Goals

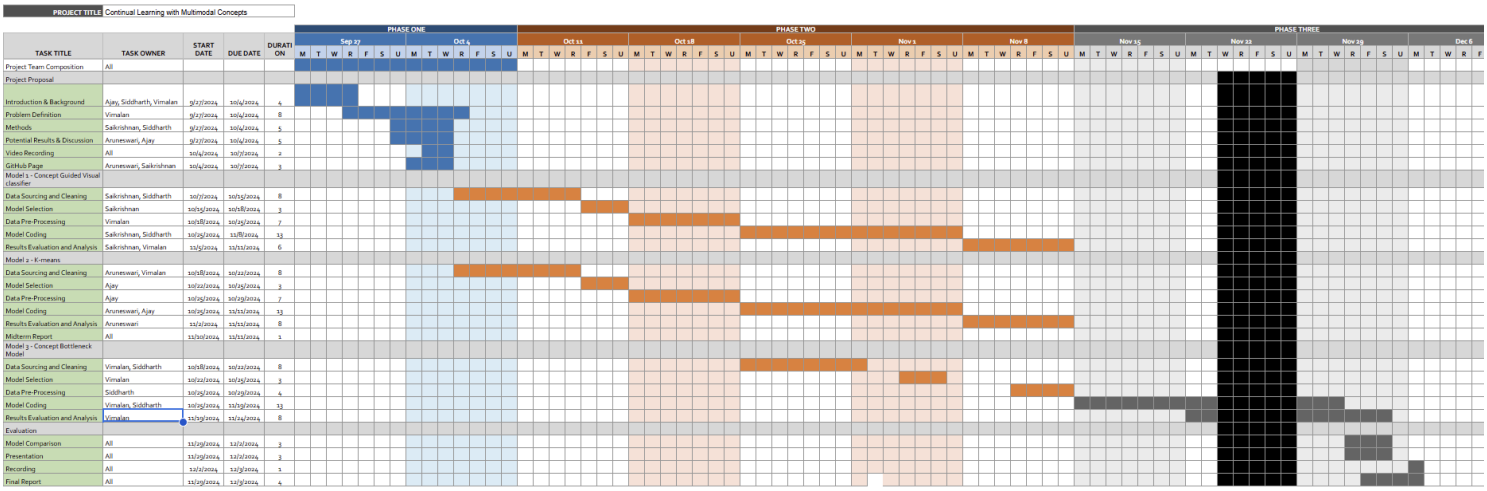
- **Develop Continuous Learning:** Create a neural network model capable of learning new information without forgetting previously acquired knowledge.
- **Enhance Interpretability:** Provide clear, human-understandable insights into decision-making processes.

Next Steps

- Experiment with strategies for exemplary replacements.
- We will try to run our models on ChestX-ray8 dataset and perform a comparison study.

Gantt Chart

GANTT CHART



PROJECT TITLE	Continual Learning with Multimodal Concepts

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Project Team Composition	All			
Project Proposal				
Introduction & Background	Ajay, Siddharth, Vimalan	9/27/2024	10/4/2024	4
Problem Definition	Vimalan	9/27/2024	10/4/2024	8
Methods	Saikrishnan, Siddharth	9/27/2024	10/4/2024	5
Potential Results & Discussion	Aruneswari, Ajay	9/27/2024	10/4/2024	5
Video Recording	All	10/4/2024	10/7/2024	2
GitHub Page	Aruneswari, Saikrishnan	10/4/2024	10/7/2024	3
Model 1 - Concept Guided Visual classifier				
Data Sourcing and Cleaning	Saikrishnan, Siddharth	10/7/2024	10/15/2024	8
Model Selection	Saikrishnan	10/15/2024	10/18/2024	3
Data Pre-Processing	Vimalan	10/18/2024	10/25/2024	7
Model Coding	Saikrishnan, Siddharth	10/25/2024	11/8/2024	13
Results Evaluation and Analysis	Saikrishnan, Vimalan	11/1/2024	11/11/2024	6
Model 2 - K-means				
Data Sourcing and Cleaning	Aruneswari, Vimalan	10/18/2024	10/22/2024	8
Model Selection	Ajay	10/22/2024	10/25/2024	3
Data Pre-Processing	Ajay	10/25/2024	10/29/2024	7
Model Coding	Aruneswari, Ajay	10/25/2024	11/11/2024	13
Results Evaluation and Analysis	Aruneswari	11/1/2024	11/11/2024	8
Midterm Report	All	11/10/2024	11/11/2024	1
Model 3 - Concept Bottleneck Model				
Data Sourcing and Cleaning	Vimalan, Siddharth	10/18/2024	10/22/2024	8
Model Selection	Vimalan	10/22/2024	10/25/2024	3
Data Pre-Processing	Siddharth	10/25/2024	10/29/2024	4
Model Coding	Vimalan, Siddharth	10/25/2024	11/19/2024	13
Results Evaluation and Analysis	Vimalan	11/19/2024	11/24/2024	8
Evaluation				
Model Comparison	All	11/29/2024	12/2/2024	3
Presentation	All	11/29/2024	12/2/2024	3
Recording	All	12/2/2024	12/3/2024	1
Final Report	All	11/29/2024	12/3/2024	4

Contribution Table

Name	Proposal Contributions
Ajay Bhat	K-Means: Model Selection, Data Preprocessing, Model Coding; Model Comparison and Presentation
Aruneswari Sankar	K-Means: Data Sourcing and Cleaning, Model Coding, Results Evaluation and analysis; Model Comparison and Report
Saikrishnan Sankar	CGVC: Data Sourcing and Cleaning, Model Selection, Model Coding, Results Evaluation and analysis; Model Comparison, Presentation
Sri Siddarth Chakaravarthy P	CGVC and CBM: Data Sourcing and Cleaning, Model Coding, Results Evaluation and analysis; Model Comparison and Recording
Vimalan Krishnan Manivannan	CGVC and CBM: Data Preprocessing, Results Evaluation and analysis; K-Means: Data Sourcing and Cleaning; Model Comparison and Report

Video & Presentation Slides

Final Submission Video

Slides

Proposal Video

References

1. Rudin, C. (2019). *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.* Nature Machine Intelligence, 1(5), 206-215.
2. Koh, P. W., et al. (2020). *Concept bottleneck models.* In International conference on machine learning (pp. 5338-5348). PMLR.
3. Yang, Y., et al. (2023). *Language in a bottle: Language model guided concept bottlenecks for interpretable image classification.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19187-19197).
4. Wang X, Peng Y, Lu L, et al. (2017). *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases.* CVPR 2017. https://nihcc.app.box.com/v/ChestXray-NIHCC
5. Esteva A, Kuprel B, et al. (2017). *Dermatologist-level classification of skin cancer with deep neural networks.* Nature, 542(7639), 115-118. https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000
