Politecnico di Torino

Politecnico di Torino

Corso di Laurea

A.a. 2024/2025

Graduation Session October 2025

Emerging Evolutionary Concepts

Clustering-Based Refinement of Concept Bottleneck Model Embeddings for Interpretable Machine Learning

Relatori:

Giovanni Squillero

Alberto Tonda

Candidati:

Peter ALHachem

Abstract

This research presents a novel approach to interpretable machine learning through the development of an evolutionary algorithm-based concept refinement pipeline for handwritten digit classification. The work addresses a fundamental challenge in explainable artificial intelligence: automatically discovering optimal granularities for interpretable concepts while maintaining or improving classification performance.

The pipeline relies on three core components: a visual concept detector, a concept bottleneck model, and evolutionary algorithm for concept refinement. The visual concept annotator transforms raw MNIST digit images into binary concept annotations for five visual concepts (loops, vertical lines, horizontal lines, diagonal lines, and curves) using computer vision techniques such as convexity analysis, Sobel edge detection, Hough line transforms, and curvature analysis. Adaptive thresholding based on 75th percentile scoring converts continuous measurements to binary annotations, achieving 25% detection rates for most concepts, even though curve detection failed due to algorithmic saturation.

The concept bottleneck model implements a true interpretable bottleneck architecture where all classification decisions flow only through the learned concept representations. The neural network uses a multi-stage design: an image encoder generating 64-dimensional concept embeddings, concept prediction layers with sigmoid activation, and a minimal digit classifier accepting only concept predictions as input. Multi-objective training simultaneously optimizes concept prediction accuracy and digit classification performance using equal-weighted binary cross-entropy and sparse categorical cross-entropy losses. Training results demonstrate excellent digit classification performance (97.8% accuracy) despite the bottleneck constraint, while concept prediction accuracy remained modest (48.6%), indicating substantial room for evolutionary improvement.

The evolutionary algorithm represents the core innovation, implementing sophisticated optimization to discover optimal concept granularities through clustering refinement. Each individual encodes complete clustering configurations specifying subdivision strategies for all concepts simultaneously. The system integrates both K-Means and DBSCAN clustering algorithms with adaptive parameter estimation and intelligent constraint handling. Fitness evaluation employs Random Forest classification on evolved binary concept features, using computational optimization strategies including subset sampling to enable efficient population-based search.

Experimental results reveal significant performance differences between clustering approaches. K-Means clustering achieved superior results with a +32.9

percentage point improvement over original concepts (from 31.4% to 64.3% accuracy), discovering an optimal configuration of 7 loop clusters, 9 vertical line clusters, 6 horizontal line clusters, and 8 diagonal line clusters. The evolutionary algorithm demonstrated excellent convergence characteristics over 15 generations with stable optimization behavior. PCA visualizations confirmed high-quality cluster separation with concept-specific granularity patterns reflecting the complexity of different visual features.

In contrast, DBSCAN clustering achieved more modest improvements (+4.9 percentage points to 36.2% accuracy) despite discovering higher granularity configurations (up to 45 horizontal line clusters). The density-based approach produced extensive noise point distributions and more volatile convergence patterns, suggesting that concept embeddings favor centroid-based over density-based clustering characteristics.

Concept distribution analysis was able to verify the level of correctness of the original concepts, with loops predominating in digits 0, 6, and 8, vertical lines in digits 1, 4, and 7, and horizontal lines in digits 7 and 4. T-SNE visualizations revealed sophisticated embedding space structures supporting meaningful clustering operations, confirming that the concept bottleneck model successfully learned discriminative representations.

The research demonstrates that evolutionary algorithms can automatically discover refined concept subdivisions that significantly enhance interpretable classification performance. The 32.9 percentage point improvement achieved by K-Means clustering represents substantial progress toward bridging the interpretability-performance gap in machine learning systems. While evolved concepts (64.3% accuracy) did not match the concept bottleneck model's performance (97.8%) or raw pixel classification (91.2%), they provide complete interpretability through human-understandable visual concepts.

This work establishes evolutionary concept optimization as a promising direction for automated interpretable machine learning, offering a systematic approach to concept granularity discovery that reduces reliance on manual concept engineering. Future research directions include addressing curve detection limitations, exploring alternative clustering algorithms, and extending the approach to more complex visual domains beyond handwritten digits.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Giovanni Squillero and Professor Alberto Tonda, for their extremely valuable guidance where they have transmitted to me the foundation of their expertise during the time of my research without forgetting their continuous support and help during this long exciting journey. Their feedback and encouragement have been highly important in shaping the project that I present today.
I am also deeply grateful for my friends and family as I sent them my heartfelt appreciation for their unconditional support, patience, and understanding during the demanding periods and challenging times as their encouragement has been my constant source of motivation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

XAI
Explainable Artificial Intelligence

CBM
Concept Bottleneck Model

TCAV
Testing with Concept Activation Vectors

CME
Concept Model Extraction

SSMTL
Semi-Supervised Multi-Task Learning

DCR
Deep Concept Reasoner

EA
Evolutionary Algorithm

NAS
Neural Architecture Search

MNIST
Modified National Institute of Standards and Technology

PCA
Principal Component Analysis

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

t-SNE

t-Distributed Stochastic Neighbor Embedding

ReLU

Rectified Linear Unit

Chapter 1

Introduction

Artificial Intelligence is experimenting a remarkable shift with the growth that deep neural networks are receiving, especially with the outstanding results noted in diverse fields of study ranging from image recognition to natural language processing. However, the progress achieved in the computation of information has created, on the hand, a rather important challenge related to the opacity of these highly articulated technological systems' decision making processes. Modern neural network models are often portrayed as a "black box" whose internal mechanisms remain largely difficult to comprehend, even to the people who designed their architectures. This limitation highlights integral problems where interpretability are essential, especially in classification tasks in domains such as medicine or autonomous vehicles.

In order to remediate the issue at hand, Explainable Artificial Intelligence became a fundamental discipline at orchestrating methods in to comprehend and clearly interpret deep neural networks system decisions. We do find among these promising methods; Concept Bottleneck Models represent significant innovative progress that focuses on enforcing all predictions to go through human-interpretable visual concepts. Nonetheless, these models are traditionally correlated with predefined concepts that are humanly annotated and may prove to be too general or incapable to encapsulate the complexity of visual patterns that are needed for an optimal classification.

The crossing between machine learning interpretability and evolutionary optimization presents a promising opportunity that helps mitigate these limitations. Although the state of the art in evolutionary algorithms have shown great success in "Automatic feature engineering" and "Neural architecture search", their application to "Concept refinement" in interpretable machine learning reside largely unexplored for this matter.

The most important obstacle addressed within this dissertation lies in the rigidity between model interpretability and classification performance in machine learning systems. Actual concept bottleneck models, though provide some interpretability, sustain important constraints that limit their practical effectiveness and strength.

To begin with, reliance on manually annotated concepts constitutes a significant challenge in the development process, which then requires specific domain expertise and notable trial-and error to distinguish appropriate concept lists from improper ones. These predefined concepts usually work on generally defined granularity basis that mis-captures distinctive visual patterns necessary for correct discriminative classification. For example, the concept "has fur" associated to an image of a cat may be insufficient to distinguish with other animals that have fur for that matter, like rabbits or hamsters. Thus, missing implicit but important variations could increase classification accuracy.

Secondly, current methods lack precise methodologies that help in finding optimal concept specifications. The choice that surrounds how to accurately detail visual concepts remains largely intuitive, potentially leading to suboptimal concept hierarchies that either oversimplify complex patterns located in visual concepts or end up creating unnecessary distributed representations that affect interpretability.

Finally, the static nature of humanly defined concepts counters the process of adaptive refinement based on learned representative embeddings, which means that, once a concept is defined, it remains fixed throughout the training and deployment phases, unable to evolve to patterns discovered during these processes. This tension reduces the possibility of discovering any compelling concepts that could represent learned embedding spaces by the neural network.

The focal point of the research conducted, thus, poses a critical question: How can evolutionary algorithms be used to automatically discover emerging concepts that enhance both interpretability and classification results in deep neural network systems? This question contains several critical sub-problems: specification on concept granularity, maintaining interpretability while rendering the model fundamentally discriminative, and implementing important evaluation decisions for evolved concept and their hierarchies.

We try and answer these limitations and restrictions in five elaborated comprehensive sections that systematically develop, implement and evaluate the proposed emerging evolutionary concept pipeline.

In the first section, literature review and theoretical foundations are provided

for a deep understanding of explainable artificial intelligence, concept bottleneck models, and evolutionary optimization in machine learning. Clear and coherent groundwork will establish necessary explanations that focus on the intersection between interpretability and evolutionary computation, while notably identifying the gaps that motivate the current research performed.

In the next section, we will present an extensive illustration of automatic concept detection based on the MNIST dataset. It will also contain a technical analysis on the design choices, adaptive thresholding that were adopted in order to present a significant concept detection accuracy on the MNIST digits.

In the following part, we will then discuss the concept bottleneck model architecture and training with its complete implementation from the concept bottleneck design to the multi objective training performed, followed by an analysis of the embedding extraction mechanism and full understanding of the training framework. An elaborated demonstration of how deep neural networks can show limitations to provide interpretability while achieving great classification performances.

In the fourth section, we will introduce the evolutionary algorithm for concept refinement which is the core methodological addition by presenting a thorough description of the evolutionary framework, that ranges from genetic representation strategies, fitness evaluation decisions and the integration of a clustering algorithm that encapsulates the newly generated concepts. This section contains a comprehensive analysis of the design choices and optimization strategies.

The final section will conclude our discussion with an explanation of the experimental results and analysis of the information provided. The results provided will present a detailed argumentation of the validity of the proposed approach going from a comparative analysis of the clustering paradigms that were utilized to a detailed evaluation of the generated concepts' quality. This part will precisely demonstrate the actual effectiveness of the evolutionary approach while providing insights that highlight the relation between clustering algorithms and concept embedding characteristics.

The thesis concludes with a comprehensive discussion of the contributions made, the limitations that were present, and directions that future research could interestingly conduct to enhance the current state that was reached within a broader context that englobes both interpretable machine learning and evolutionary computation research.

1.1 Goal

The main goal of this research that has been conducted is to study whether evolutionary algorithms can automatically discover refined visual concepts that can improve interpretability while keeping an important classification performance in deep neural networks, while specifically addressing the challenge between model clarity and the accuracy of the predictions in Explainable Artificial Intelligence (XAI).

This project aims to develop and validate a sequential three-stage pipeline that removes the need for any extensive definition of concepts manually by actually discovering optimal concept granularity though evolutionary optimization automatically. It is also extremely crucial to consider maintaining complete explanation of the model through a concept bottleneck architecture that forces all predictions to flow exclusively through visual concepts such as loops,lines and curve that are understandable by humans.

Additionally, the research performed focuses on demonstrating that evolutionary algorithms can discover new emerging concepts that can find impressive classification accuracy over baseline of manually-defined concepts, which establishes the first integrated framework that combines both concept bottleneck models with evolutionary algorithms that aims for automatic concept generation.

One of the key objectives of this study is to systematically evaluate and compare K-Means clustering, a centroid-based technique, with DBSCAN clustering which is more of density-based approach, all within the evolutionary framework. This will allow us to determine which clustering technique works better with the concept embedding spaces for optimal concept refinement. The expected results include a pipeline that is able to contain visual concept annotation, concept bottleneck modeling and evolutionary concept refinement, along with performance results that demonstrates the effectiveness of evolved concepts. The analysis of the pipeline will also represent the evidence that supports that the evolutionary algorithm works well for automated concept discovery, as well as a comparison analysis of the clustering algorithms and a foundation for future research.

By achieving these objectives, the research will demonstrate that interpretable machine learning systems do not need to sacrifice significant performance for interpretability, while also providing a practical way toward deploying AI systems in critical domains such as medical diagnosis and autonomous vehicles where both high accuracy and complete clarity on the results are extremely important.

1.2 Structure of the thesis

This thesis is structured as follows:

- **Chapter 1:** Introduces the research problem of interpretable machine learning and defines the objectives of evolutionary concept refinement.
- **Chapter 2:** Explains the previous literature present on explainable AI, concept bottleneck models, and evolutionary algorithms which then establishes the theoretical foundation for this work.
- **Chapter 3:** Describes the methodology, including visual concept detection algorithms, concept bottleneck architecture, and evolutionary optimization framework.
- **Chapter 4:** Presents experimental results comparing K-Means and DBSCAN clustering approaches, demonstrating significant performance improvements through evolutionary refinement.
- **Chapter 5:** Concludes with a summary of key findings and contributions to interpretable machine learning.
- **Chapter 6:** Discusses limitations and outlines future research directions for extending the evolutionary concept optimization approach.

Chapter 2

Background and State of the Art

2.1 Explainable Artificial Intelligence: Foundations and Evolution

The domain of Explainable Artificial Intelligence (XAI) was developed as a response to the opacity that surrounds deep learning architectures. The "black-box enigma" that comes within Deep Neural Networks interfere with the widespread utilization of such powerful instruments. This can be widely seen within industries that adopt extremely strict regulations where error margins must be immensely low. As a result, there has been a significant expansion within the field of explainable AI [1], that focuses on improving the comprehensive quadrant of deep learning systems.

The most crucial obstacles that are challenged by XAI lies in the tension between complicated models and their interpretability. The impressive revolution of artificial intelligence was represented by machine learning models that adopt traditional approaches that operate like "black boxes", where the mapping from input to the desired output is completely opaque even to the designers of such complex architectures. This "obscurity" in comprehension became especially tricky in critical applications to AI systems such as medical diagnosis, judicial decision-making, and autonomous systems, where understanding the reasoning behind the decision that is being taken is crucial for trust and accountability.

Explainable Artificial Intelligence literature has developed several taxonomies that subdivide methods that help with interpretability. The most important difference lies between **intrinsic interpretability** methods, which are the main

impetus behind a direct representation of neural network models, from **post-hoc explanation** methods that try to explain already-trained models through diverse methodologies and analysis.

Post-hoc explanation methods have been extremely important in the early stages of XAI research, with feature attribution approaches representing the most widely adopted category. Nowadays, the most publicly adopted domain in XAI is feature importance methods. For a given data point, these practices assign a score that displays the influence of each feature to the algorithm's decision. The designated features could be represented by a pixel, a patch or a word vector for instance. Nonetheless, feature importance methods have shown to be strongly affected by any perturbation done to the input vector or model parameters, experiments have shown that these practices do not significantly increase human understanding or trust in the functionality of the models in that sense [1].

The reliability problem that comes with these methods has been extensively represented, but on the other hand, the inconsistencies of these methods demonstrate their fragility to input and model perturbations. Research has shown that the 4 interpretation of a neural network is intrinsically shaky with any small change that is done to the model parameters [2] that may lead to drastically different explanations for the same identical inputs.

Intrinsic interpretability practices, on the other hand, attempt to mitigate the issues presented in the post-hoc ones by focusing on the added clarity into the model architecture. This paradigm has gained a lot of popularity especially in applications where understanding model reasoning is important.

A promising direction within intrinsic interpretability has gained a lot of attention to try and solving the problem of interpretability of neural models. **Concept-based explanation approaches** provide explanations in terms of human understandable concepts instead of low-level features. This transition defines meaningful concepts (such as "white wings" or "curved beak" in bird classification) rather than individual pixels or words that resonates slower with human's cognitive abilities.

Due to the obstacles that were discussed in the feature importance strategies, the other two types of XAI approaches have been receiving a notable amount of attention from model extraction approaches, and concept-based explanation approaches. Model extraction approaches attempt to translate complex black-box models into simple understandable models. On the other hand, Concept-based explanation approaches present explanations in terms of high-level concepts, by extracting this concept information from model's latent space.

The work performed on **Testing with Concept Activation Vectors (TCAV)** quantifies the degree to which user-defined concepts influence model predictions. It works through concept activation vectors located in the model's representation space. However, it is important to note that TCAV needs predefined concepts and operates as a post-hoc analysis tool rather than building interpretability into the model architecture itself [1].

Based on these foundations, researchers have developed awareness around "concept based explanations" that reflect on whether a given set of concepts contains sufficient information for accurate classification [2]. This work emphasized an important obstacle on how to determine the necessary concepts that are plausibly enough for a certain classification task.

2.2 Concept Bottleneck Models: Architecture and Theoretical Foundations

The presentation of **Concept Bottleneck Models (CBMs)** presents an ideal transition toward intrinsically interpretable neural networks. CBMs offer interpretability within the design architecture, meaning that all predictions flow through an intermediate layer of human-interpretable concepts [2]. This method fundamentally differentiates from post-hoc explanation methods by making sure that no information flows without passing through the concept layer.

The CBM architecture implements a two-stage prediction process: $\mathbf{X} \rightarrow \mathbf{C} \rightarrow \mathbf{Y}$, where X represents input features, C denotes the concept predictions and Y indicates final task labels [1]. This subdivision in layers showcases a clear human reasoning process, that typically involves the identification of relevant concepts, usually made through visual observations, before making any classification decisions. The intermediate concept layer C serves a bottleneck for the flow of information that constrains the model to make decisions solely on interpretable concept information.

We can clearly see that concept-based models aim to increase human trust in deep learning models by using human understandable concepts to train interpretable models such as logistic regression or decision trees. This methodology helps in the augmentation of human trust in AI predictors that have been behaving in an opaque manner rather than a clear understandable decision-making process.

Training CBMs needs concept annotations at training time, representing both

the strength and weakness of the strategy adopted. The presence of concept supervision allows for the optimization of accuracy with respect to concept prediction, ensuring that the intermediate architecture is in line with the human-interpretable concepts.

Research has shown us that CBMs can achieve important classification performances while maintaining interpretability through the concept bottleneck constraint. Experiments conducted on the CUB-200-2011 bird classification dataset showed that CBMs could achieve high accuracy while providing transparent reasoning paths through detected visual concepts [2].

However, several limitations were noticed in the original CBM methodology regarding **concept leakage** – where information bypasses the concept bottleneck due to insufficient bottleneck constraints present – which has been remediated by adding further regularization techniques and stricter architectural design modifications. Additionally, researchers have indulged into other methods to handle the incompleteness that can result from concept sets, where predefined concepts may not capture all information necessary for optimal classification performance.

The notable performance of CBMs has encouraged several extensions and variations that address specific and variation that can help mitigate the limitations presented. **Fuzzy bottleneck approaches**, for instance, allow concept activation to be numerically continuous instead of the traditional binary approach. This approach has fundamentally changed the perception of how concepts are presented in the input data from binary decision making to a "degree of confidence" that relates to the presence of the concept. Additionally, it proved to be particularly valuable in domains where concepts can exist in continuous form rather than a discrete one

Interactive CBMs, another variant of the traditional model, allow users to correct concept predictions during inference and training. This approach fixes concept prediction errors by allowing real-time correction and model refinement based on human feedback. The interactive paradigm has shown promise in medical applications, where domain experts can present specific corrections that improve both concept accuracy and classification performance.

2.3 Automated Concept Discovery and Extraction

Although CBMs provide a strong framework for interpretable classification, they do contain a fundamental limitation: they heavily depend on manually defined concept sets. This reliance creates several obstacles that stop concept-based approaches from being theoretically complete or practically applicable.

The **concept selection** problem revolves around determining which concepts are relevant for a specific classification task. Manual concept selection usually is done by experts in a certain domain or through intuitive thinking, which may actually overlook important but non-obvious concepts or even include irrelevant concepts that add noise without improving discrimination. This constraint becomes critical when dealing with complex domains where the relationship between visual patterns and classification outcomes is not directly obvious.

Another challenge lies within **concept granularity** that deals with the appropriate level of abstraction for concept definitions. Concepts that are too broad (such as "has a beak") may lack the necessary discrimination for an accurate performance in terms of classification. On the other hand, concepts that are too specific may render the representational space extremely detailed to a point where generalization is no longer effective. Extensive experimentation and domain expertise are certainly required while determining an optimal concept representation space with effective granularity.

Concept Model Extraction (CME) presents a framework for analyzing deep neural networks through concept-based extracted models. CME addresses the limitations presented in requiring extensive concept annotations by dealing with partially-labeled datasets and extracting concept information from multiple network layers [1].

The CME framework introduced above divides the neural networks into two separate functions: **an input-to-concept function** that maps raw inputs to concept representations, and **a concept-to-output function** that predicts task labels from concept information. This separation enables us to analyze how neural networks encode and utilize concept information for classification decisions.

CME exploits **Semi-Supervised Multi-Task Learning (SSMTL)** to get concept predictions from network representations, treating each concept as an independent prediction task [1]. The framework adds concept information across

multiple network layers, identifying the optimal layer for each concept prediction based on its accuracy. This multi-layer strategy makes sure that different concepts may be best represented at different levels of abstraction within the network division.

It is notable to mention that concept-based explanation approaches are usually capable of handling binary-valued concepts only, which implies that multi-valued concepts must be converted into binary first. For instance, given a concept such as "shape", with possible value "square" and "circle", these approaches must convert "shape" into two binary concepts *"is_square"* and *"is_circle"*. This undermines that such approaches are computationally expensive, since the binary representation of a concept space usually has a high cardinality, and error prone. On the other hand, CME can handle multi-valued concepts directly, without the need to binarize concepts.

Moreover, concept-based explanation methodologies are typically correlated to the representative space of a single layer when extracting concept information. DNNs have been more performant, in terms of feature extraction, when using layers closer to the output utilizing higher-level data representations, compared to layers closer to the input. This does clearly mean that choosing a single layer imposes an unnecessary trade-off between low and high-level concepts. CME is able to combine latent space information efficiently from multiple layers and thus, overcome this problem.

The extracted models give us a plethora of analytical possibilities: model precision through concept-based decision analysis, model debugging by identifying problems within inactive concepts, and most importantly the knowledge gained behind the identification of the effectiveness of key concepts in classification accuracy, with case studies showing how model can be improved by over 14%, using only 30% of the available concepts [1].

2.4 Neural-Symbolic Integration and Concept Reasoning

Looking to the limitations of traditional concept-based models, the **Deep Concept Reasoner (DCR)** presents the first interpretable concept-based model that builds on concept embeddings [2]. However, state-of-the-art concept-based models, which rely on concept embeddings to attain high performance, are not completely interpretable. In fact, concept embeddings lack clear semantics on single isolated dimensions, which loses interpretability in favor of the capacity of the model which may lead to a reduction in human trust when using such types of models for that

matter.

DCR takes into account this challenge by implementing a differentiable-based approach on concept embeddings that build a set of fuzzy rules which can then be executed on semantically meaningful concept truth degrees to provide a final interpretable prediction. This architecture heavily relies on activation functions that limit the number of concepts per rule, while making sure that learned rules remain interpretable and maintain a significantly decent performance.

Indeed, DCR showcases the capability to discover meaningful logic rules, that matches known ground truths, even if we take into consideration the absence of training concept supervision. The system, in fact, learns to apply logical operations instead of concept representations, which then enables both the learning process and reasoning process withing a single framework. This implementation fixes fundamental limitations of neural symbolic approaches by combining learned concepts with reasoning mechanism which renders the model more interpretable.

Experimental results have shown that DCR accomplish better task accuracy than state-of-the-art interpretable concept-based models, while discovering meaningful logic rules. These discovered rules present transparent representations of model behavior and can explain misclassifications at the level of tasks. For example, a task might be mis predicted because some concepts have been predicted wrongly, or the scores in selecting the set of concepts did not fully reflect the relevance of these concepts. This transparency helps in the extraction of simple counterfactual explanations without the need to integrate external algorithms [2].

A crucial characteristic of explanations is stability while having small perturbations. Users do not take into consideration explanations that change significantly while having similar inputs for which the model makes the same prediction. DCR explanations have proven to be, in fact, very consistent, especially when comparing them to local post-hoc explainers [2]. Stability is fundamental for building trust in concept-based systems.

The logic rules clearly showcase which concepts play an important factor in prediction. Following established methods, DCR generates counterexamples by first ranking the concepts present in the rule based on their relevance scores, then starting from the most relevant concept and inverting their truth value until the prediction changes. For example, knowing that the relevance score of “yellow” is the highest relevance score for a concept when it comes to predicting a “banana”, flipping its truth value will automatically render the input classified NOT a “banana”.

2.5 Evolutionary Algorithms in Machine Learning

Evolutionary Algorithms (EA) define a group of optimization techniques where natural selection and genetic processes are the basis of these techniques. The foundational work established Genetic Algorithms as the first systematic approach to evolutionary computation, introducing key concepts such as population-based search, genetic operators (selection, crossover, mutation), and fitness-based survival [3].

The EA family has grown significantly, englobing foundations such as Genetic Programming, Differential Evolution, Evolution Strategies, and Evolutionary Programming [3]. Each variant presents a solution for obstacles found in optimization through specific representations, operators and selection mechanisms.

The fundamental EA principles include population-based search, which allow the algorithm to explore potential solutions in parallel; genetic operators, on the other hand, help in balancing the exploration and exploitation through limited amount of randomness; fitness-based selections, finally, help in guiding the search toward a high-valued solution. These principles have proven to be especially effective when optimizing problems characterized by complex, non-linear, multi-modal objective functions.

EAs were able to be a significant game changer within applications in neural network optimization, addressing challenges in architecture design, hyperparameter tuning and feature engineering. Neural Architecture Search (NAS) represents one of the most successful applications, where EAs automatically discover optimal network architectures for specific tasks and datasets [4].

Additionally, EAs have shown significant success in feature engineering, where they use different combinations, transformations, and selection strategies. This application domain shares important characteristics with our target of concept refinement: both involve discovering optimal representations that can balance performance with interpretability.

Genetic Programming for Feature Engineering, in fact, uses mathematical expressions that transform raw input features into more discriminative representations. These evolved features often discover unusual relationships between input variables that improve classification performance while remaining interpretable as mathematical expressions [4].

The success of EAs in feature engineering suggests their potential applicability to concept generation and concept refinement dilemmas. Just as EAs can discover optimal feature combinations through evolutionary search, they may be capable of discovering optimal concept subdivisions and organizations that enhance both discrimination and interpretability.

2.6 Research Gaps and Opportunities

2.6.1 Concept Granularity Optimization

One of the biggest obstacles in applying current concept-based techniques practically is the manual specification of concept granularities. Finding the best concept subdivision to use is still very much done intuitively, with no systematic attempt to strike a balance between interpretability and discrimination.

In the end, the issue of concept granularity revolves around questions of how far broad concepts should be subdivided, when subdividing concepts helps or hurts classification performance, and how to preserve interpretability as the concept sets get more granular. Each of these constitutes a roadblock toward the formation of automated procedures capable of systematically exploring concept granularity spaces.

2.6.2 Dynamic Concept Refinement

Within existing CBM approaches, static concept definitions remain fixed throughout training and deployment; thus, adaptation to patterns discovered within learned embedding spaces or refinement on the basis of classification performance is prevented.

Dynamic concept refinement thus remains an under-researched avenue with the potential for greatly increasing the flexibility and effectiveness of concept-based approaches. Integration of concept refinement with training procedures can allow for discovering more effective concept organizations while still preserving interpretability guarantees.

2.6.3 Evolutionary Concept Optimization

The intersection of evolutionary algorithms and concept-based interpretability remains largely unexplored, despite the high correlation between EA optimization

capabilities and concept refinement challenges. EA has proven that it can be successful in feature engineering and neural architecture optimization which makes a case for their potential in finding emerging concepts and organization problems.

Evolutionary concept optimization could address several current limitations: automatic discovery of optimal concept granularities, exploration of concept subdivision strategies, and integration of interpretability and performance objectives through multi-objective optimization. This research direction represents a significant opportunity for advancing the state of interpretable machine learning.

The success of these research directions (concept bottleneck models, automated concept discovery, neural-symbolic reasoning, and evolutionary optimization) creates an optimistic opportunity for developing complex and automated approaches to interpretable machine learning that can discover optimal concept refinements while it can maintain correctness, and it remains understandable for humans.

Chapter 3

Experimentation

After a thorough examination of the state-of-the-art revolving around Explainable AI, concept bottleneck models and evolutionary algorithms. We start by presenting the core potential of this research that presents an innovative approach that tackles interpretable machine learning by the development of an evolutionary algorithm-based concept generative pipeline for handwritten digit classification. The pipeline sequentially encapsulates the reasoning behind neuro-symbolic strategies with evolutionary optimization to directly discover and refine visual concepts that increase both the classification performance and the model interpretability.

The core foundation of the pipeline presents the implementation of three major components: The **visual concept detector**, the **neural bottleneck model** and the **evolutionary concept algorithm** for clustering the embeddings of the concepts. Instead of considering the black-box feature learning, this approach models human-interpretable visual concepts present in digits, such as, loops and lines, and uses evolutionary algorithms to automatically discover new generated concepts.

The system described above discusses a crucial problem in Explainable AI, which revolves around the methodology used in order to discover the right level of abstraction for interpretable concepts while prevailing or improving the classification results. The traditional systems already implemented usually relied on predefined concepts or manually annotated concepts. This pipeline introduces, instead, an automated evolutionary approach that discovers refined concepts directly from learned embeddings.

3.1 Visual Concept Annotator

The visual concept annotator presents the foundation of the interpretable concept learning pipeline by transforming raw **MNIST digit images** into meaningful concept annotations with a final binary form adopted in the training phase of the concept bottleneck model. This primal phase utilizes computer vision techniques to detect five humanly understandable visual concepts that can help in distinguishing handwritten digits.

3.1.1 Loop Detection – Identifying Circular and Oval Structures

Loop detection analyses the shape properties of digit contours, especially when it comes to measuring how "round" or "closed" a shape is. The algorithm calculates the convexity of the shape, which is a mathematical that indicates whether a shape resembles a circle or oval.

Figure 3.1 below demonstrates the complete loop detection process across three different digit types:

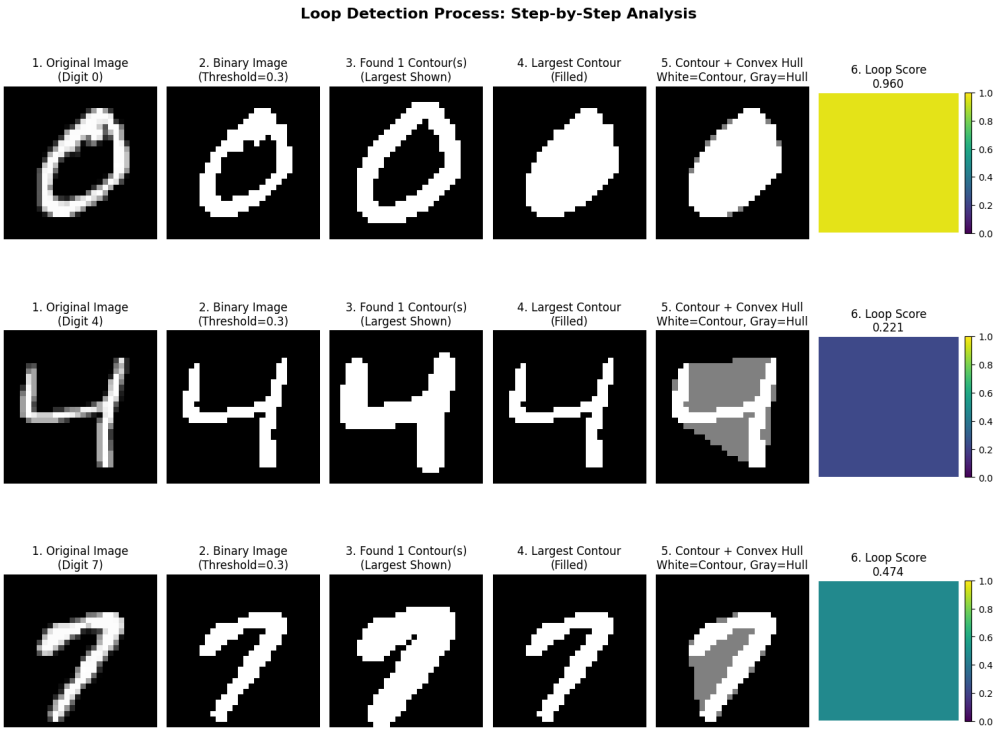

Figure 3.1: Loop Detection Analysis

- **Digit 0 (Excellent Loop Detection – Score 0.960):**The original image displays a clear oval shape with a perfect loop structure, which presents the ideal scenario to detect a loop. When converting the image to binary, we use a specific threshold of 0.3 in order to preserve the complete loop boundary without any fragmentation found. The next step utilizes a contour detection algorithm where, in that case, successfully traces the entire oval parameter continuously. Following the contour detector, a solid contour visualization algorithm reveals the filled in area of the digit in white as a completely closed area. Finally, the convex hull analysis of the digit illustrates the minimal difference between the white contour area and the grey hull area, indicating that the shape is already correctly convex-bounded. The final score obtained of 0.960, confirms the "near-perfect" loop characteristics and shows why circular digits like, 0, 6, 8, 9 consistently achieve high loop scores.

- **Digit 4 (Poor Loop detection – Score 0.221):** The structure of this digit is more related to right angles rather than convex curvatures which usually support loop-like characteristics. The comparison with the convex hull showcases a significant grey area, that shows a huge difference between the actual contour and the convex hull. This is mainly because of the angular-like behavior that is present in the digit "4" that create concave regions that must be filled in instead of convex regions, this results in a larger hull area compared to the contour area. The low score of 0.221 is realistically expected for a digit like 4 that has a dominant angular geometry.

- **Digit 7 (Moderate Loop detection – Score 0.474) :** The number "7" presents an interesting case, as we can see in the handwritten original pictures some significant curves at the corners where horizontal and diagonal lines actually meet, the system detect a partial hull that match a gray hull area that is larger than the contour but as dramatically different from the digit "4", for example. The explanation due to this phenomenon is mainly due to the junction on the corner between the horizontal and the diagonal line. The score obtained reflect that the algorithm can catch subtle curved characteristics even in a significantly dominant linear digit.

In Summary, Loop detection algorithm basically try to answer the question: "How much does a curved shape resemble a filled circle?" The results demonstrate that "0" as a digit has a near-perfect oval shape and that the other two digits have scored low results due to the predominant angular-like shape that they represent with a distinctive mention that on curvatures that are not totally linear, the algorithm was able to score a higher result. The algorithm presented above successfully distinguishes between circular or oval shaped digits and angular shaped ones, providing a robust measure for loop-like characteristics.

3.1.2 Vertical Line and horizontal Detection

The vertical line detection uses edge detection techniques to identify strong vertical gradients in the original image. The algorithm starts by applying a mathematical filter that searches for vertical edges while ignoring other types (horizontal or diagonal for that matter). The algorithm utilizes a vertical Sobel filter that calculates the intensity changes in the horizontal direction that indicates the presence of vertical edges. It then projects the edges found by the filter vertically to find columns with strong vertical features (we can understand it by considering the accumulation of the rows in a matrix for instance).

On the other hand, the algorithm used for the horizontal line detection follows a similar pattern to the one used to find vertical lines. A Sobel filter that can detect the intensity changes in the vertical directions this time, help in the indication of horizontal edges. The projection performed to find the horizontal features are performed on a row basis, which means that cumulation of the intensities of the matrix is performed by adding all the columns, we then explore any significant intensities found.

Figure 3.2 below demonstrates the complete vertical and horizontal line detection process across three different digit types, revealing how Sobel edge detection and projection analysis three different digit types, revealing how Sobel edge detection and projection analysis distinguish between different linear orientations within handwritten digits.

- **Digit 1 (Vertical Dominance - V-Score: 0.289, H-Score: 0.208):** The original image presents a clean and almost straight vertical edge that represents the most optimal case when it comes to vertical line detection. When the filter is applied, it results in a "intense" red-yellow activation that reproduces a clear vertical band that overlaps the digit edges. This is a heatmap representation of the structure of the digit. On the other hand, the horizontal edge detection reveals a drastically different response to the one found on the vertical detection, showing only a very shy presence of the red strokes on its heatmap, the red filaments also come in a scattered way with plenty of noise. The vertical projection represents another clear analysis of the detection of vertical edges in the original images with an edge strength that peaks at nearly 0.3 that spans approximately from columns 8 till 18 with a nearly bell-shaped curve distribution. Although the edge strength is moderate for that matter, it still reflects on the successful determination of the vertical edges within the intensity distribution. The horizontal projection on the contrast displays an almost perfectly flat shape that oscillates with small changes around the baseline, confirming the absence of a horizontal element

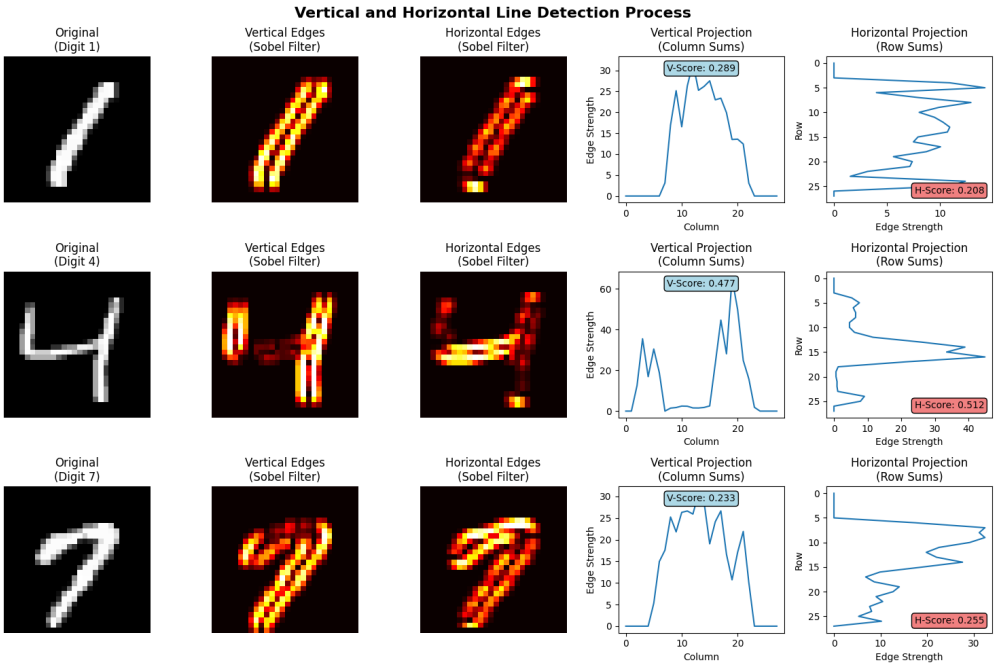

Figure 3.2: Vertical and Horizontal Line Detection Process

within the digit itself. The low score attributed to the edge strength of the horizontal projection clearly demonstrates for that instance the correctness of the intuitive visual representation of the digit "1".

- **Digit 4 (Dual Structure Excellence - V-Score: 0.477, H-Score: 0.512):** Although the original image displays a challenging architecture, the detection system was able to simultaneously identify both orientations with vertical and horizontal components. The vertical Sobel filter reveals two strong yellow responses concentrated in two diverse regions. The intense right vertical edge presents the strongest activation of the heatmap while the left vertical one showcases a secondary but significant response, creating a unique bimodal activation filter. The line that connects both edges is represented by a clear horizontal line in the center that is created by a concentrated bright yellow and red heatmap. The architecture of this digit is explained by the vertical and horizontal projection analysis, as it includes a strong peak around columns 18-22 that corresponds to the right vertical line, then a clear gap that represents the space, then a notable peak around columns 3-9 that corresponds to the left element; on the other hand, the horizontal projection shows a noticeable concentrated spike around row 15 that abruptly rises from the baseline, this offers a great evidence on the presence of the horizontal line. The algorithm provides balanced detection scores for both vertical and horizontal projections,

which represent optimal dual detection effectiveness and proves its success even though the detection of complex and overlapping structures can be challenging.

- **Digit 7 (Mixed Orientation Challenge - V-Score: 0.233, H-Score: 0.255):** The original image displays a rather intriguing test sample of an uncertain orientation of the linear edges that are different from what their usual classification would be because this handwritten digit blends both horizontal and diagonal features. When applying the vertical filter, it produces a moderate red-yellow response scattered along the diagonal edge because its complex pattern contains some vertical components that somewhat gives the filter some indication in this direction. This mix in orientational property totally coincides with the behavior of the activation of the filter, as we can see that it is clearly less intense than pure vertical edges detected but surely more important than the horizontal edges that come out through the vertical Sobel filter. On the other hand, the horizontal edge detection algorithm was able to find a strong, concentrated yellow activation precisely located at the top of the horizontal bar, that clearly identifies the prominent structural feature while remaining confined with the upper region of the digit. When it comes to the projection analysis, the graph in the figure above shows an interesting side of the mixed orientation: the vertical projection reveals a more distributed response spanning from approximately columns 8 to 20, which really projects how the diagonal edge spreads its intensity across multiple spanned out columns, going from upper left to lower right of the image space. Although it shows a little activation over the remainder of the vertical span, the horizontal projection is still greatly concentrated between rows 5 to 8 roughly, forming a tiny, sharp peak that indicates the location of the horizontal edge. The significantly low V-Score and moderate H-Score are very close and accurately reflect the mixed nature of the structure and demonstrate how the algorithm handles digits that do not present a simple and clear categorization, as neither orientation achieved dominance nor the horizontal element marginally outperforming the diagonally influenced vertical response.

In conclusion, the basic question that the vertical and horizontal line detection algorithm aims to address is "What are the primary and secondary linear orientations in the structure present in this digit?". The findings that were explored above do provide a full explanation of the vertical dominance in Digit 1 with respect to the horizontal one, the mixed orientation in Digit 7 which provided some ambiguous results in terms of scores, while Digit 4's balanced dual-orientation structure achieved high scores in both dimensions. The most important thing that this algorithm has provided is its ability to deal with dual-orientation structures (with significant scores), mixed orientations (balanced scores) and pure linear orientations (score separation).

3.1.3 Diagonal Line Detection - Capturing Angled Patterns

Diagonal detection makes use of the **Hough Line Transform**, a technique that helps in identifying straight lines in an image at any angle. The algorithm looks for lines that are between *20 and 70°* and *110 and 160°* with respect to the horizontal.

Figure 3.3 below presents how diagonal detection is done using the Hough Line Transform, this allows us to understand how this complex technique can distinguish between curved strokes and actual linear diagonal elements.

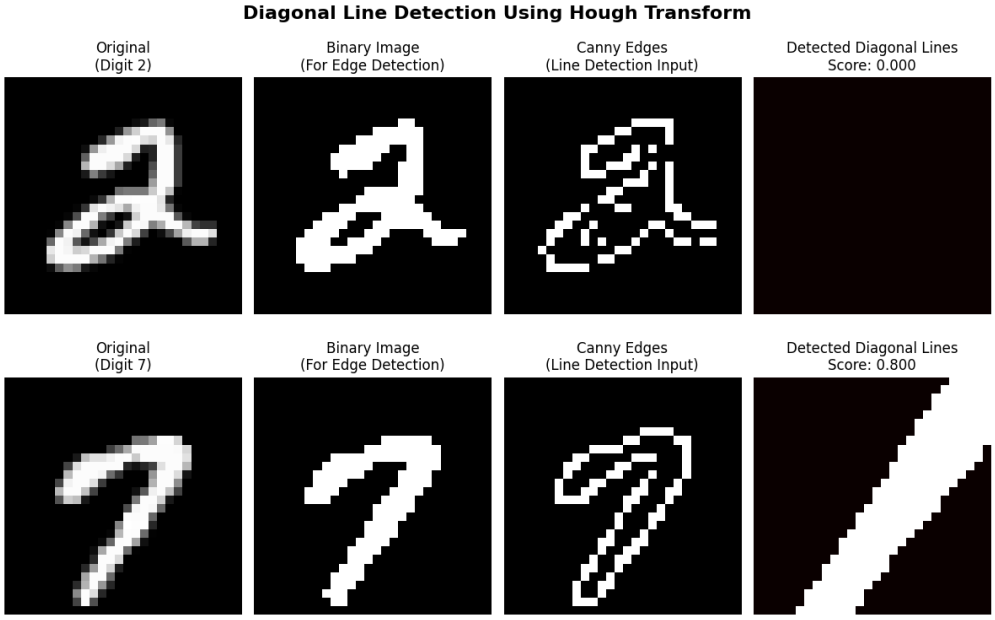

Figure 3.3: Diagonal Line Detection Analysis

- **Digit 2 (Failed Detection - Score: 0.000):** The conversion to the binary representation of the digit initially showcased that digit has a naturally curved form. As a second step, the Canny edge detection was able to capture the digit's curved limits, which provides a unique mapping that displays the form's contour. However, when the Hough Line Transform looks at these edges to find lines segments that fall within the diagonal angle ranges of 20–70° or 110–160°, no line contenders were detected. This happens because the digit is entirely composed on curvatures, which, although they appear diagonal to the human eye, they do not have the straight-line characteristics required by the Hough algorithm. The 0 score indicates that no diagonal lines were detected, and the final graph is painted black. This shows how accurate the algorithm is at distinguishing between curved diagonal-like forms and actual straight

diagonal line segments.

- **Digit 7 (Successful Detection - Score: 0.800):** The "7" is a clear illustration of a successful diagonal detection due to its diverse structural composition.The binary conversion creates a clean representation that clearly shows both the horizontal and diagonal components of the digit structure. Canny edge detection creates accurate edge maps that highlight the horizontal top bar and the downward diagonal stroke. After examining these edges, the Hough Line Transform was able to identify the diagonal stroke, as shown by the bright white line overlay in the final detection panel. As demonstrated by the highest score of 0.800, which confirms the presence of a true diagonal structure within the necessary angle range, the method reliably detects straight diagonal elements while ignoring other orientations present in the same digit.

In conclusion, the diagonal detection algorithm was able to provide impressive and intriguing performances as shown by the graphs where Digit “2” did not showcase any signs of clear diagonal line structures while on the other hand the algorithm did successfully classify the “7” Digit as written number with a diagonal stroke. The accuracy of this algorithm ensures that only truly linear diagonal elements are registered while ignoring any curvatures, horizontal or vertical lines.

3.1.4 Curve Detection - Identifying Subtle Curved Elements

Curve detection considers the most complex mathematical concept as its objective is to calculate the curvature at each point along the edge by looking at the rate at which the gradient direction changes. Low curvature indicates straight lines, while high curvatures imply a curve.

Figure 3.4 below illustrates the sophisticated curvature identification method with perfect score consistency, which makes use of mathematical concepts akin to figuring out the "steering wheel angle" when driving around the digit’s edges.

- **Digit 0 (Perfect Curve Score: 1.000):** We start by analysis the gradient magnitude of the digit where it can be shown an intense red-yellow ring with strong edges that are present around the oval boundary. This indicates that the edge strength remains constant throughout the making of the circle. The curvature map, as a next step, presents occasional bright spots that indicate the presence of high curvature scattered throughout the form, which represents a constant curvature presence in this digit. The analysis performed on the curvature of strong edges shows purple and pink regions in the final mask graph where the curvature measurements are concentrated mainly on the edge

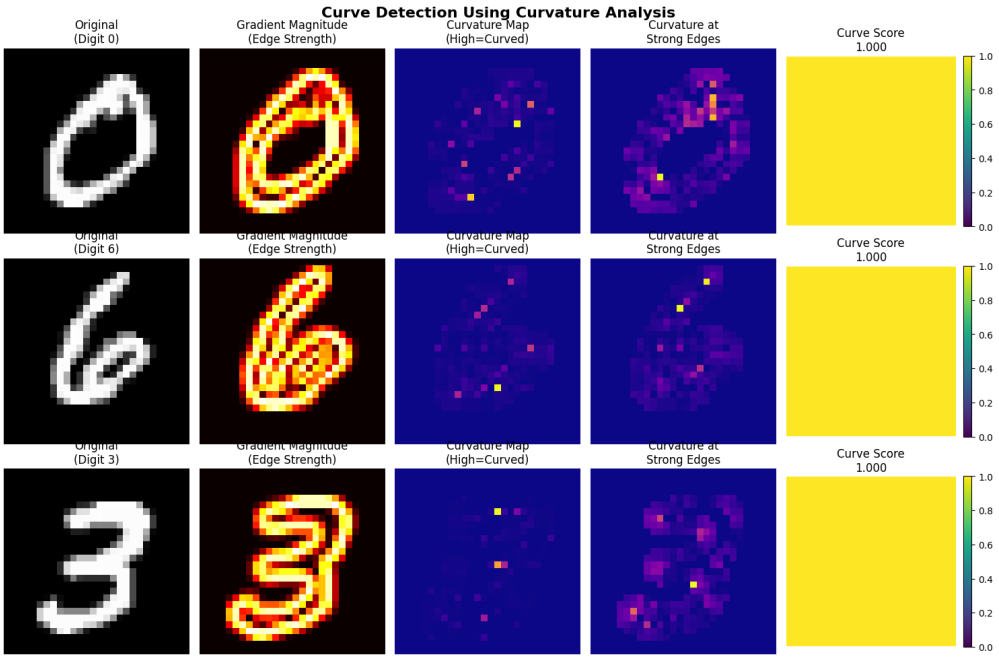

Figure 3.4: Curve Detection Process Analysis

regions to make sure that the algorithm focuses on the most reliable edge information. This actually makes sure that the curvature represented within the digit are notable characteristics throughout the boundaries.

- **Digit 6 (Perfect Curve Score: 1.000):** This digit demonstrates a more complex analysis with respect to the previous one due to the presence of a mixed structural composition of curved edges. The gradient analysis performed on the original image shows a strong edge detection on both the curved loop at the top and the straighter section at the bottom of the digit. However, the curvature analysis demonstrates a more prudent detection of the curves by showing bright curvature spots primarily concentrated in the curved regions while showing a minimal curvature receptions in the straight portion at the bottom. The detection algorithm was able to determine a shadowy figure of the curvature represented by the digit, while characterized with low intensity, but still potent enough to trace impeccable curves that delimit the digit. Despite the mixed structure, a perfect score was still attributed to the representation and that indicates that the curved portion dominates the curvature analysis more than the mere mention of straight segments within the digit.

- **Digit 3 (Perfect Curve Score: 1.000):** This digit represents, perhaps, one of the most complex curvature representations within all other digits,

with its characteristics and strongly resembles the “S” shape structure that is composed by multiple curves. The gradient detection, which shows the edges across the upper and lower curves, indicates that the algorithm will potentially analyze multiple curved regions within the digit. The measurements that are divided across multiple regions show that the algorithm can detect curvatures across multiple sections rather than a single continuous curve. The complex geometry at hand which constitutes multiple curved parts that represent the evaluation of the total curvature, demonstrates that the algorithm can handle complex curved structures.

It is important to note that the consistent obtainment of perfect scores across all examples shown above reveals that the curvature detection algorithm tend to saturate at maximum values for any digit with tendencies of having significant curved elements. This means that the algorithm is extremely sensitive to curved structures and that future modifications must include a more conditional algorithm with a more complex scoring mechanism.

3.1.5 Concept Distribution Analysis across Digit Classes

In the Figure 3.5 below, we can visualize the distribution of the digits across the initial concepts that are annotated by the concept detector, as we can interpret the popularity of each digit within each concept based on the distribution it got.

- **Loop Concept Distribution Patterns:** The concept distribution analysis reveals highly intuitive patterns that validate the original concept design. Loop concepts, as expected, show maximum predominance in digit 0 (92% prevalence) and significant presence in digits 6 (66%) and 8 (42%), with minimal presence in linear digits like 1, 2, 3, 4, 5. The near-zero prevalence in digits 2, 3, 4, 5 confirms that the loop detection algorithm is able to correctly distinguish between curved and angular edges. The moderate presence in digit 9 (17%) may reflect some different variations in handwriting style where some "9" digits contain more pronounced loop structures.
- **Vertical Line Distribution Intelligence:** Vertical line concepts display some discrimination across digit classes, with highest prevalence in digit 1 (69%) as it was expected to be the most purely vertical digit. Significant presence in digits 4 (31%) and 7 (32%) present the vertical components in these mixed-structure digits, although digits like 7 for the human eye are more likely to be associated with a diagonal line but it is mainly related to the structure of the handwriting in the dataset. We can also detect a moderate presence in digit 8 (35%) which likely captures the vertical elements in the upper and lower lines of "8" digit as a vertical line. The algorithm appropriately shows lower prevalence in primarily horizontal or curved digits.

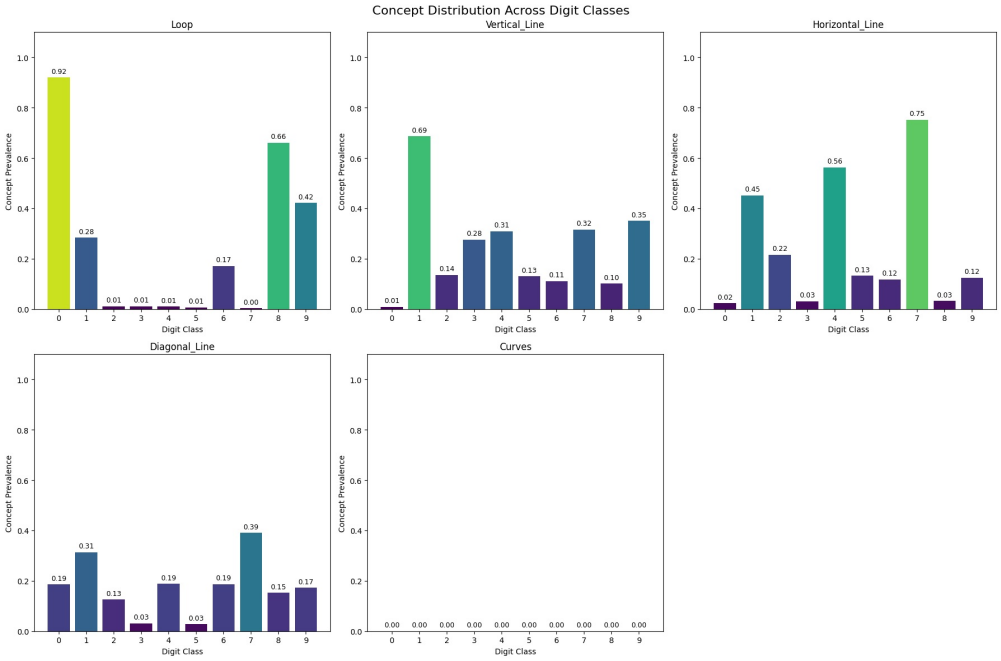

Figure 3.5: Concept Distribution Analysis

- **Horizontal Line Distribution Accuracy:** Horizontal line detection demonstrated an excellent discrimination with high prevalence for digit 7 (75%) reflecting the prominent horizontal bar in the upper part of the digit, and a moderate prevalence in digit 4 (45%) corresponding to the crossbar that connects both the diagonal edge and vertical edge together, and some shy presence in digits 2 (22%) and 3 (13%) where horizontal elements play second to none role basically. The low prevalence in pure vertical digits like 1 validates the algorithm's capability to distinguish between different linear orientations, as it was clear from the MNIST dataset the complete absence of the horizontal line within this digit for example.

- **Diagonal Line Pattern Recognition:** Diagonal line distribution shows a more uniform presence in digits with digits 2 (31%) and 7 (39%) being the most predominant of all the digits. This result correctly verifies the traditional and human-recognizable digits that contain diagonal edges. This moderate presence across other digit classes suggests that the Hough transform algorithm captures unclear diagonal elements that may not be immediately distinctive but that clearly contribute to the discriminative of the digit. The relatively lower overall presence of this concept compared to other ones shows the scarcity of clear diagonal elements in the MNIST dataset.

3.1.6 Adaptive Thresholding - How to convert scores to Binary Concepts

After each algorithm calculates the continuous score for its respective concept, the system converts them into binary (0/1) annotations using adaptive thresholding based on the dataset's 75th percentile of scores. This method notably outperforms fixed threshold approaches by automatically adapting the distribution of visual features in the data.

For every concept, all 60,000 training images are assigned continuous values between 0.0 and 1.0 before the 75th percentile thresholding is performed. Following the sorting of these scores, the system calculates the value at which exactly 75% of all values fall below it. Any image that scores higher than this 75th percentile is given a binary annotation of 1 (concept present), whereas images that score at or below the threshold are given a binary annotation of 0 (concept absent). With about 25% of the dataset displaying each concept, this ensures that each concept is present across all concept types.

Many significant detection problems are resolved by the adaptive nature of percentile-based thresholding. Loops, for example, naturally produce lower average scores than vertical lines, suggesting that different concepts should normally have different baseline frequencies and score distributions. Using percentile-based thresholds ensures that each concept captures its most representative examples, without strongly considering the score distribution. This method also gives us some robustness by automatically adapting to different datasets or image properties without the need for a manual threshold adjustment for each concept type.

3.1.7 Concept Annotator Results and Analysis

The implementation produced the following concept distribution results:

- The system was able to create distinct concept distribution patterns across the MNIST dataset that reveal both successful performances and more challenging ones. Loop detection successfully identified 15,000 training samples out of the 60,000 representing exactly the 25 percentiles of the training dataset. This was also transmitted to the test set where 2,500 out of the 10,000 images were classified as containing loop structures, maintaining the same 25.0% detection rate. Vertical line detection has shown very similar characteristics as it was able to identify 15,000 sample for the training set and 2,500 samples for the test set, both representing 25.0% detection rate with perfect consistency between training and testing datasets. Horizontal line detection also achieved very good

performances with same percentage of currently identified samples for both training and test sets, each representing 25.0% of their respective datasets and demonstrating the robustness of the adaptive thresholding mechanism for these fundamental concepts.

- Diagonal line detection has created different results with 1,693 detections out of 10,000 samples or 16.9% of the test dataset, and 10,868 training samples out the total 60,000 total or 18% of the training dataset. This drop from the target detection rate of 25% reflects how selective the algorithm related to the detection of diagonal lines can be, while it presumably rejected curved edges that might appear diagonal lines to the human eye.

- The most significant failure was presented in the curve detection, which produced zero positive detections across the training and testing datasets. This is mainly related to the saturation of the algorithm rather than the absence of any curved features in the dataset. We can blatantly notice that both training and test dataset failed to receive any positive curve annotations.

The success of Adaptive thresholding

The first three concepts have presented perfect thresholding performance. In fact, the loop, vertical line and horizontal line detection systems all achieved detection rates of exactly 25.0% across both training and testing datasets. The fact that these three algorithms were able to comply with the 25% target shows the ability of adaptive thresholding and the detection algorithms to create important score distributions that allow for correct classification between when a concept is present and absent within a sample.

Diagonal Line Limitations

The diagonal line detection algorithm did achieve a lower detection rate of 18.1% for training and 16.9% for testing, without being able to reach the intended 25% target. This limitation happened because the Hough Line Transform algorithm specifically detects straight diagonal line segments within a defined angle (20-70° and 110-160°), while many of the MNIST dataset samples that appear diagonal to the human eye have a curved edge rather than straight diagonal segments. Digits like “2” and “7” may have diagonal-appearing elements, but the “2”, for example, consists more of curved elements that fail to activate the straight-line detection algorithm, while “7”, for example contains a true straight diagonal line. This result showcases how the algorithm is being able to distinguish between curved diagonal-like shapes and actual straight diagonal lines.

Curve Detection Algorithm Failure

The curve detection algorithm could not present the proper theoretical idea behind the algorithm as it failed to detect any sort of curvatures with the images that clearly present curved edges. This was due to a saturation of the algorithm that was represented by an attribution of a maximum score of 1.0 to nearly all the images of the dataset, which produced a uniform score distribution transmitting that all samples present an ideal case of curvatures. When we try to identify the 75th percentile of a uniform distribution of values equal to 1.0, this means that the method obviously selected was 1.0, and since no sample was able to score higher than 1.0 (being the maximum value), no sample did satisfy the requirements for positive concept annotation, resulting in the 0.0% detection rate observed in both the training and testing sets.

Diving deep in the analysis of this saturation issue, the discriminative strength for the curve identification was lost because the algorithm was connotated with high sensitivity, although technically it was successfully identifying curved parts as we can see from the high scores. The mathematical curvature formula used and the normalizing technique which allows results to be maximized to 1.0, let even slightly curved elements produce maximum responses. This is a common issue in the design of computer vision algorithms, where mathematical representations need to be well calibrated to produce meaningful scores rather than saturating at some value.

Consequences on subsequent stages of the pipeline

These results have had important consequences on training the concept bottleneck model and the subsequent stages of evolutionary algorithms. The evolutionary algorithm's capacity to find new refined concepts linked to curves may be extremely difficult because the system only uses four functional concepts rather than the five that were expected. Nevertheless, the three efficient concepts – loops, horizontal and vertical lines – provide a great foundation that is great at demonstrating the evolutionary concept methodology. Although the diagonal detection could not perform as well, it still provides valuable idea annotations for a subset of the sample. Even though the curve detection did not satisfy the expectations that was perceived from it, it still presents an opportunity to show that the pipeline is robust against technical problems with specific components.

3.1.8 Final Output

Each MNIST image receives four meaningful binary concept annotations:

- **Loop:** 1 if strong circular/oval structure detected, 0 otherwise (25% of dataset)

- **Vertical line:** 1 if a strong vertical edge is found, 0 otherwise (25% of dataset)
- **Horizontal line:** 1 if strong horizontal edge is present, 0 otherwise (25% of dataset)
- **Diagonal line:** 1 if angled stroke detected, 0 otherwise (18.1% of dataset)
- **Curves:** Consistently 0 due to algorithm saturation (requires future refinement)

These binary annotations, despite the curve detection limitation, provide a meaningful foundation for the concept bottleneck architecture and evolutionary refinement processes that follow in the pipeline.

3.1.9 Concept Bottleneck Model

The pipeline central component that represents the neural-symbolic representation of this framework is the Concept Bottleneck Model. This model implements a clear interpretable bottleneck architecture in which learned concepts are the only means by which all digit classification decisions can be made. This is performed mainly because the model cannot rely on obscure, incomprehensible features to make its predictions. In fact, this design makes sure that each classification decision to be explained using the predetermined visual concepts computed above.

Structure of the Architecture

In order for this architecture to be uniquely represented with respect to pseudo-bottleneck architectures that might allow information to leak around the concept layer, the model employs what is known as direct concept bottleneck. In this implementation, the classification of the digits is only dependent on concept predictions, and for that reason the presence of a bottleneck that redirects the visual data to be encoded using the interpretable concept representation is extremely crucial. This "wall" in the architecture makes sure that the decision-making of the model is always explicable. This architecture is defined by different stages with its own properties and functionalities:

- **Stage 1 - Image Encoder for Concept Prediction:** The architecture begins with a deep neural encoder that transforms raw pixel data into rich representational embeddings specifically designed for concept detection. The encoder employs a progressive dimensionality reduction strategy, starting with a dense layer that maps the 784-dimensional input (28×28 flattened MNIST images) to a 512-dimensional hidden representation using ReLU activation. This initial expansion allows the network to capture complex pixel interaction

patterns that are essential for visual concept detection. The architecture then applies generalization capabilities, followed by a second dense layer that compresses the representation to 256 dimensions while maintaining ReLU activation for non-linear feature learning. A second dropout layer provides additional regularization before the final encoding stage.

- **Stage 2 - Concept Embedding Layer:** At the end of the encoder is a concept embedding layer that generates 64-dimensional embeddings that are especially tailored for concept representation. This dimensionality was selected to maintain computational efficiency for later processing stages while offering enough representational capacity to capture the subtleties of visual concepts. These embeddings are important elements that must keep the balance between interpretability and explaining concepts since they provide the basis for both concept prediction and the later evolutionary refinement process.

- **Stage 3 - Concept Prediction Layer:** The concept embeddings are passed directly into a concept prediction layer, which generates separate binary predictions for each of the five visual concepts using sigmoid activation. Given that visual concepts can co-occur within single digits (for example, both vertical and horizontal lines in a "4"), the sigmoid activation is essential because it enables each concept to be predicted independently. All visual data must be transformed into understandable concept representations in this layer, which serves as the explicit bottleneck.

- **Stage 4 - True Bottleneck Digit Classification:** By only accepting concept predictions as input and denying direct access to the original image data or intermediate representations, the final classification stage carries out the true bottleneck constraint. A 10-way Softmax output layer for digit classification comes after a tiny hidden layer with 32 neurons and ReLU activation. The model is directly encouraged to learn comprehensive and meaningful concept encodings by this minimal architecture, which guarantees that digit classification performance is solely dependent on the caliber and completeness of the concept representations.

Multi-Objective Training Strategy

The model uses a progressive multi-objective training method that aims at maximizing both digit classification and concept prediction at the same time. Because concept detection is multi-label and each concept is handled as a separate binary classification problem, the concept prediction task employs binary cross-entropy loss, while the digit classification task employs categorical cross-entropy loss, suitable for single-label multi-class classification. In order to make sure that balanced

optimization across both tasks without favoring either concept learning or digit classification, both loss functions are given equal weight (1.0) in the combined objective.

In addition to dual supervision that are the binary concept annotations obtained from the visual concept annotator and the ground-truth digit labels, the model is trained using flattened MNIST images. The concept prediction layer is directly supervised by the concept annotations, which are arranged as a matrix with each row representing an image and each column representing a concept. Due to the bottleneck constraint created by this dual supervision, the model must learn to encode visual information in a way that supports both accurate concept detection and efficient digit classification, which creates a different training dynamic from traditional neural networks.

Concept Embedding Extraction

The model's capability to extract learned concept embeddings, which are used as input to the evolutionary algorithm is important after training. The 64-dimensional representations that encode visual information in the feature space that are relevant to the specific concept are obtained through a "sub-model" in the extraction process, which gives us the output of the concept embedding layer. The training and the testing datasets are extracted in this way, which guarantees that the evolutionary algorithm utilizes detailed embedding data in order to evaluate and optimize new concepts.

The embeddings related to specific concepts are created by organizing the extracted embeddings based on the concept annotations. The system creates a dataset that shows the learned encoding of particular visual concepts by identifying all images that received positive concept annotations for each concept and then we extract the associated embeddings. It is important to note that index mappings are always in sync with the evolutionary algorithm so that it can exactly refine concepts by tracking embeddings back to their original images and concept annotations.

The embeddings related to each concept, as well as their source image indices and references to the entire training and testing embedding sets are all present in a structured data container. In this way, we can satisfy the requirements presented by the evolutionary algorithm to examine concept-specific patterns while also keeping track of the larger dataset structure.

Interpretability of the model

The bottleneck architecture provides a strong assurance towards the model interpretability. In fact, every classification choice is explicable in terms of the visual concept related to it because the digit predictions can only access concept predictions and not the original raw image or any intermediate representations. The whole classification process can be interpreted thanks to this architecture of the bottleneck, as it prevents the model from creating hidden decisions that do not come from the interpretable concept layer.

We can also consider it as a trade-off between classification accuracy and interpretability as the model will focus more on satisfying the condition of the latter one. This clearly allows us to understand the model behavior, increase the confidence in automated systems and comprehend the premises of classification errors. We can also rely on the model to evaluate the concept representation of the annotator, as it will give us a great direction if the automated concepts are a reliable metric to use or not.

3.1.10 Evolutionary Algorithm for concept generation and refinement

The pipeline's main innovation is the Evolutionary Algorithm (EA), which uses a complex optimization framework to automatically find the best concept granularities through clustering refinement. In order to improve interpretability and classification performance, this phase refines the learned concept embedding from the bottleneck model into concept subdivisions. The EA works on the principle that the initial broad concepts can be divided into more discriminative, specific "sub-concepts" that preserve interpretability while capturing finer-grained visual patterns.

Evolutionary Representation and Genetic Encoding

A complete clustering configuration that explains how to split all visual concepts at once is represented by each individual in the evolutionary population. Each gene that is used by the genetic representation of the individual correlates to a clustering parameter for one of the five concepts. Now in order to balance interpretability and granularity, each gene in the K-Means clustering, for example specifies the number of clusters to be formed for that concept, and the values limited to gene are between two and eight. On the other hand, each gene in DBSCAN determines the minimum sample of parameters which range from 5 to 15 and regulates the density to form a cluster. Every individual is guaranteed to define a new concept

that can be applied directly to the embedding space, thanks to this representation.

Some interesting constraints that consider the real-world limitations of concept subdivision are added into the evolutionary representation. Additionally, to keep the algorithm from going back to the initial state of having only 5 concepts, there is a constraint that obliges to subdivide the embeddings for each cluster into new concepts. And in order to avoid overfitting or loss of interpretability, we mitigate this issue by adding another upper bound constraint on the number of clusters in K-Means and DBSCAN (8 clusters for K-Means and 15 minimum samples per cluster for DBSCAN).

K-Means Clustering Integration

The concept embeddings can be divided into clusters of approximately equal size using the K-Means clustering technique, which is basically centroid based. The way K-Means clustering works in this algorithm is represented by a subdivision of the embeddings of all the samples annotated with a concept into regions, when the individual gives a certain cluster count for that concept. Each cluster becomes a separate sub-concept on its own and as a result we generate new binary features through these cluster assignments. This method gives us reliable and repeatable clustering results and allow the algorithm to optimize them using the evolutionary algorithm.

DBSCAN Clustering Integration

A radically different clustering paradigm is offered by the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) integration, which finds clusters based on local density as opposed to global centroids. Cluster shapes are automatically determined by the algorithm, which can also detect noise points that don't belong to any cluster and identify clusters of any shape. To ensure adaptive sensitivity to the real data distribution, the epsilon parameter is automatically estimated using percentile-based thresholds and k-distance analysis. This method can handle concepts with irregular or non-spherical distributions and is particularly 30 Experimentation good at finding natural cluster boundaries in embedding spaces.

Both DBSCAN and K-Means use advanced parameter estimation techniques that can be optimized based on the embedding distribution of each concept and the properties each of its own. In order to determine the local density variation for DBSCAN, the epsilon parameter examines the nearest neighbors based on a percentile scale which guarantees that the determination of cluster is based on the specific embeddings. The K-Means algorithm, on the other hand, makes sure to consider minimum cluster size constraints in order to avoid inadequate solutions

and also uses multiple random initializations to guarantee convergence to global optima.

Fitness Evaluation

The fitness evaluation procedure is an advanced machine learning pipeline that uses downstream classification performance in order to assess the quality of refined evolved concepts. To produce a complete set of refined sub-concepts, the evaluation starts by applying the clustering configuration that each individual specifies to all concept embeddings. After that, these sub-concepts are transformed into binary feature matrices, which are fed into a classification algorithm created especially for quick assessment throughout evolutionary optimization.

Because of its exceptional performance characteristics for binary feature data and its computational efficiency during evolutionary search, Random Forest classification is used as the assessment mechanism in the fitness evaluation. With 50 estimators and a maximum depth of 8, the Random Forest classifier gives a balance between computational speed and predictive accuracy, allowing for the effective assessment of thousands of individuals over several generations. There is a direct correlation between the quality of concept refinement and the improvement in performance since the classifier uses the evolved concept features as input and then outputs digit classification accuracy as the fitness metric.

Several computational optimization techniques are used for the evaluation framework to allow the evolutionary search performed to be real taking into consideration an acceptable time for running the algorithm. By restricting evaluation to 5,000 randomly chosen samples, training subset selection reduces computational load while delivering statistically significant evaluation. 2,000 samples are used for test subset evaluation in order to keep the balance between evaluation speed and accuracy. These subset sizes were established empirically to allow for population-based search across several generations to obtain an accurate estimation of the fitness function.

Evolutionary Operators

The mutation operator uses a flexible approach that abides by the limitations presented while leveraging the features of every clustering algorithm. Mutation applies small perturbations (± 1 cluster) to K-Means clusters, with bounds checking to make sure all values stay within the acceptable range of 2–8 clusters. Larger perturbations (± 2 min samples) are used for mutation for DBSCAN individuals, considering the different sensitivity properties of density-based clustering related

to the DBSCAN technique. Throughout the running process of the algorithm, the 30% mutation probability preserves population diversity while allowing for balanced exploration.

The crossover operator uses uniform crossover, which modifies clustering parameters gene-by-gene between parent individuals with a 50% chance of coming from either parent. Each gene produces children that combine the clustering property of a specified concept coming from both parents. By using this method, the evolutionary algorithm can find hybrid solutions that may use different subdivision techniques for various concepts, such as a straight-forward clustering representation for simpler concepts and a more detailed technique for complex ones.

Using **tournament selection** with a tournament size of three, the selection considers the balance between not letting populations become similar and keep selection pressure, so the algorithm doesn't end up with weak individuals. This process avoids premature convergence to weakly optimized solutions while also guaranteeing that high-fit individuals have a higher chance of reproducing through the operator stated above. A compromise that preserves the genetic diversity required for further research while retaining enough selection pressure for convergence is the tournament size of three.

3.1.11 Test Mapping and generalization

The evolutionary algorithm's method for mapping identified clusters from training embeddings to test embeddings while preserving concept coherence is among its most advanced features. To ensure that evolutionary refinement respects the original concept annotations, the mapping process works under the restriction that test samples can only be assigned to clusters of concepts that they initially possessed. This restriction stops the algorithm from reassigning samples to incorrect concept categories in order to artificially improve performance.

K-Means Mapping Strategy

Using the trained cluster centroids result from training, test mapping assigns directly the test embeddings to its appropriate cluster that corresponds to samples with a specific concept annotation. Test samples are only allocated to clusters that fall under their annotated concept categories thanks to this method, which also offers deterministic, repeatable test mappings that represent the cluster boundaries discovered during the training phase.

DBSCAN Mapping Strategy

Because the algorithm is density-based and lacks explicit cluster centroids, DBSCAN test mapping is more complex. By using spatial proximity analysis, the mapping strategy assigns test embeddings to clusters according to the distances between them and cluster centroids calculated from training data. Based on the greatest distance between training samples and their cluster centroid, the algorithm determines a cluster radius. Test samples that fall inside this radius are then assigned to the appropriate cluster. This method offers significant generalization to test data while preserving the spatial coherence of density-based clusters.

Concept Feature Extraction

The final result that is issued from the evolutionary algorithm consists of a binary feature matrix that represent the evolved concepts. Each generated cluster becomes an independent binary feature, creating a feature space that typically contains 10-30 refined concepts compared to the original 5 concepts that were perceived by the annotator. The binary nature of these features maintains interpretability while providing the discrimination necessary for improved classification performance.

The algorithm is able to generates concept descriptions that maintain the relationship between original concepts and their refined subdivisions. Each evolved concept receives a descriptive name that indicates its original concept, clustering algorithm, and cluster identifier such as *“has_loop_kmeans_cluster_2”* or *"has_vertical_line_dbscan_cluster_0"*. This naming convention allows the researchers to understand the relationship between original concepts and their evolutionary refinements while supporting detailed analysis of the discovered patterns.

The extraction procedure includes a strategy that ensures the system can produce useful results even if the evolutionary optimization runs into problems. The system automatically returns to the initial concept annotations while preserving the same data structures if the clustering process is unable to generate enough refined concepts. This fallback method guarantees that the system remains reliable while clearly identifying optimization issues that might call for changing parameters or the algorithm itself.

Chapter 4

Experimental Results and Performance Analysis

In this section, we present the experimental results obtained by both the Concept Bottleneck Model, concerning the classification of the digit or concept prediction. Furthermore, we will discuss the results gathered by the Evolutionary Algorithm using K-Means clustering from one part and DBSCAN clustering as an alternative to the previous technique. A thorough analysis on the successful performances that we were able to obtain and the critical obstacles that were perceived across the experimentation's final predictions will also be discussed using table and visuals graphs when necessary to better describe the analytical performance of the obtained results.

4.1 Concept Bottleneck Model Results and Analysis

The model was executed for 15 epochs in both the training and validation phase, and it was able to reveal some interesting dynamics that provide crucial insights into the effectiveness of the concept bottleneck architecture. The training process successfully presented two distinctive learning trajectories for the concept prediction and the digit classification tasks that highlight both the challenges and successes of the bottleneck approach, without taking into consideration the strong similarities in the convergence of both tasks. The results of the training and evaluation phase (considering both concept and classification accuracy in decimal values as well as concept and classification loss) are presented within Table 4.1 and Table 4.2

The training begins with significant challenges in the first epoch, showing concept prediction accuracy of only 50.38% and digit prediction accuracy of 37.24%,

Table 4.1: Concept Bottleneck Model Training Results for 15 Epochs

Epoch	Concept Acc.	Concept Loss	Digit Acc.	Digit Loss	Total Loss
1	**0.5038**	0.3552	0.3724	1.9937	2.3489
2	0.4380	0.4141	0.9016	0.5830	0.9971
3	0.4355	0.3705	0.9590	0.2293	0.5998
4	0.4432	0.3369	0.9692	0.1527	0.4897
5	0.4516	0.3122	0.9757	0.1187	0.4309
6	0.4641	0.2975	0.9784	0.1017	0.3992
7	0.4647	0.2892	0.9806	0.0892	0.3784
8	0.4735	0.2806	0.9821	0.0794	0.3600
9	0.4709	0.2724	0.9839	0.0743	0.3467
10	0.4844	0.2671	0.9843	0.0670	0.3341
11	0.4787	0.2590	0.9851	0.0632	0.3223
12	0.4825	0.2613	0.9841	0.0647	0.3260
13	0.4806	0.2541	0.9869	0.0536	0.3077
14	0.4827	0.2503	0.9865	0.0550	0.3053
15	0.4859	**0.2488**	**0.9877**	**0.0519**	0.3008

Table 4.2: Concept Bottleneck Model Evaluation Results for 15 Epochs

Epoch	Concept Acc.	Concept Loss	Digit Acc.	Digit Loss	Total Loss
1	0.4376	0.4014	0.8564	0.7362	1.1374
2	0.4360	0.3596	0.9604	0.2427	0.6023
3	0.4747	0.3228	0.9706	0.1545	0.4773
4	0.4474	0.3011	0.9719	0.1322	0.4333
5	0.4705	0.2880	0.9719	0.1281	0.4161
6	0.4823	0.2812	0.9748	0.1117	0.3928
7	0.4597	0.2657	0.9751	0.1093	0.3750
8	0.4708	0.2634	0.9724	0.1168	0.3803
9	0.4823	0.2549	0.9779	0.1043	0.3591
10	0.4798	0.2494	0.9775	0.1065	0.3558
11	0.4931	0.2472	0.9763	0.1160	0.3631
12	0.5027	0.2438	0.9782	0.1100	0.3537
13	**0.5141**	0.2397	0.9779	0.1043	0.3439
14	0.4927	**0.2355**	0.9788	0.1042	0.3396
15	0.4847	0.2367	**0.9803**	**0.1026**	**0.3392**

indicating that the model initially struggles to learn meaningful concept representations from the visual features. However, one of the most striking patterns emerges in the dramatic improvement in digit classification performance. By *epoch 2*, digit accuracy jumps from 37.24% to 90.16% on training data, while validation digit accuracy reaches an impressive 96.04%. This significant increase is noticeable throughout the early stages of training, with the digit accuracy reaching around 98% by the fifth epoch and then maintaining this performance for the remainder of the training period. The final epoch reaches **the highest digit accuracy in training with 98.77% and 97.88% in validation accuracy**. These impressive results in the classification task show that the bottleneck constraint does not limit the classification prediction of a digit once the concepts are learned.

On the other hand, the concept prediction task shows a markedly different and more challenging learning trajectory, with much more gradual improvement throughout the entire training process as its training begins with an accuracy of 50.38% and then slowly to reach a stable accuracy of 48.59% by the fifteenth epoch after a notable drop in accuracy that started from the second epoch. The descriptive information about the concept accuracy details how actually learning meaningful concept representations is more challenging than classification of digits. Furthermore, the validation concept accuracy follows a similar pattern as it starts at 43.76% and then reaches 48.47% by the end *epoch 15*. This slow improvement in the validation accuracy does not negate the fact that the rate of improvement is really slow and thus, concept detection presents the most important bottleneck in the architecture and mitigating its limitations may necessitate architectural modifications or different learning strategies.

The loss that resulted from the model reveals the optimization challenge faced by the model during training for both concept detection and digit classification. **The total loss decreases from 2.34 in Epoch 1 to 0.3 in the final epoch**, which ensures that the optimization of both objectives is successful overall. However, while we examine the loss metric provided, it actually presents deeper insights into the learning dynamics. **The digit prediction loss drops dramatically from 1.99 to 0.05**, representing a **reduction of over 97%**, while **concept prediction loss decreases less from 0.35 to 0.24, representing approximately a 30% reduction**. This significant disparity suggests that once the model learns basic concept representations, the digit classification task becomes relatively straightforward due to the bottleneck architecture's effectiveness, but refining concept detection accuracy remains challenging throughout the entire training process.

These results provide great validation that the bottleneck architecture is clearly effective and present a strong theoretical foundation. Despite forcing all digit classification decisions to flow exclusively through only 5 concept predictions, the

model was capable to achieve nearly 98% accuracy, demonstrating that the learned concept annotations contain enough discriminative information for a significant classification performance. This exactly coincides with our the fundamental hypothesis that visual concepts can be effective and also complete intermediate representation for digit recognition tasks. The architecture successfully maintains interpretability without sacrificing classification performance, achieving the dual objectives of explainability and effectiveness that were previously detailed as an important target to find the balance between both purposes.

Additionally, the training process maintained an impressive stability and great improvement rates across all metrics without significant oscillations or convergence difficulties even though the convergence rates that were obtained by the concept layer were modest to the say the least. The learning curves demonstrated through the results shown in Table 4.1 and 4.2 a clear and steady increase in accuracy without signs of overfitting, training instability, or premature convergence. The balanced loss weighting (1.0 for both objectives) has in fact shown its reliability when optimizing both target objective functions while preventing either task from dominating the learning process. The consistent improvement in both training and validation phases throughout the 15-epoch period suggests that further training of the model, for that matter, could potentially provide even more improvements, particularly in concept prediction accuracy.

The training results provide us with an optimal foundation for the evolutionary algorithm phase that follows while simultaneously presenting significant opportunities for improvement. The average at best concept prediction accuracy of approximately 48-51% suggests that there is plenty of room for improvement in concept representation quality, making this an ideal objective for evolutionary refinement methods. The evolutionary algorithm will operate on concept embeddings that have been trained to support high-quality digit classification with accuracy that dabbles between 97% and 98%, providing a robust basis for discovering newly refined concept subdivisions. The gap between the digit classification success and concept prediction challenges provides us with the insight that the learned embeddings contain rich representational information that may not be fully captured by the original concept detection algorithms represented by the five original concepts, which also suggests that evolutionary refinement could unlock additional discriminative power within the embedding space of the concepts. That, itself, significantly renders the expected results from the evolutionary algorithm for concept refinement to be quite exciting to analyze.

Hence, we can clearly see that the concept bottleneck model serves as the crucial foundation for the evolutionary algorithm phase as it provides both great

representative concept embeddings (each within its specific space) that will be refined and a robust evaluation technique for measuring the refinement quality of these concepts. The learned embeddings capture the model's representation of visual concepts in a 64-dimensional space that supports clustering and refinement of these concepts. The training results demonstrate that although the current concepts demonstrate reasonable performance, significant optimization potential remains present and feasible, making this an ideal target for evolutionary improvement. This dual role of digit classification and concept detection makes the concept bottleneck model an important bridge between the initial concept detection phase and the evolutionary optimization phase that will proceed.

4.2 Evolutionary Algorithm Results and Analysis on K-Means Clustering

In Figure 4.1 , we can detect that the evolutionary algorithm was able to successfully discover an optimal clustering configuration that equates between the granularity of the emerged concepts and the discriminative influence of each cluster. The best individual that was found within our framework presents **the following configuration: [7, 9, 6, 8, 0]** representing 7 refined loop clusters, 9 emerging vertical line clusters, 6 new horizontal line clusters, 8 diagonal line clusters, and 0 curve clusters (due to the curve detection failure discussed previously). This configuration showed that the algorithm is able to determine new granularities within a specific concept, which verifies the correctness of the hypothesis that presents different visual concepts to require different levels of subdivision in order to maximize their effectiveness.

4.2.1 Cluster representation and Embedding Space structure

- **Loop Concept Clustering Analysis:** The PCA visualization reveals notable clustering quality for loop concepts, with seven distinct clusters showing clear spatial separation and minimal overlap. The clustering structure demonstrates a sophisticated understanding of loop variations within the embedding space, with clusters arranged in a roughly circular pattern that likely corresponds to different loop sizes, or orientations, or completeness levels. The clean separation between clusters validates the evolutionary algorithm's decision to employ seven subdivisions for this concept, suggesting that loop embeddings contain rich structural information that benefits from fine-grained clustering.

- **Vertical Line Concept Clustering Analysis:** The vertical line concept

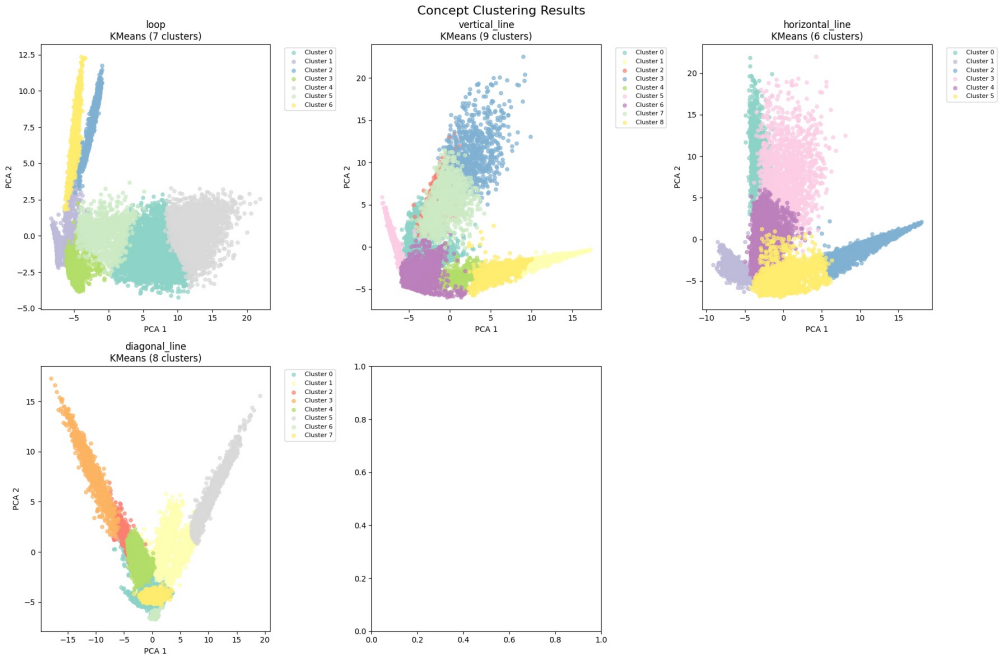

Figure 4.1: Concept Clustering using K-Means

shows the highest granularity out of all the other concepts with nine different clusters arranged in a complex configuration space. The PCA visualization shows clusters distributed across diverse regions with the vertical line concept as its unique characteristics. Some clusters appear in an elongated formations while others form compact groups, this presents several types of vertical structures (thick and thin lines or straight and slightly curved). The selection of nine clusters by the algorithm seems to connect embedding visualization to separate spaces within the embedding space.

- **Horizontal Line Concept Analysis:** The Horizontal line clustering was able to demonstrate a moderate subdivision with six clusters that show good separation and balanced population distributions. The clusters are to form distinct spatial regions as seen in the PCA space, with some clusters forming compact groups and others showing more elongated distributions. This suggests that horizontal line variations are somewhat less complex than vertical lines but still can be represented in meaningful subdivision that overlook simple binary classification.

- **Diagonal Line Concept Analysis:** The diagonal line concept shows the most fragmented clustering pattern, with eight clusters distributed across the embedding space as it presents a scatter representation of the embeddings.

This division likely shows that diagonal patterns within the MNIST dataset are complex in their general form, as they seem to appear in fewer digit types with a great number of variable ways to see the diagonals. This distributed representation of the embeddings over the eight clusters shows that diagonal lines are more complex to cluster than other concepts, especially when the diagonal line present within the dataset samples are also limited by default.

4.2.2 Evolutionary Algorithm Convergence Analysis

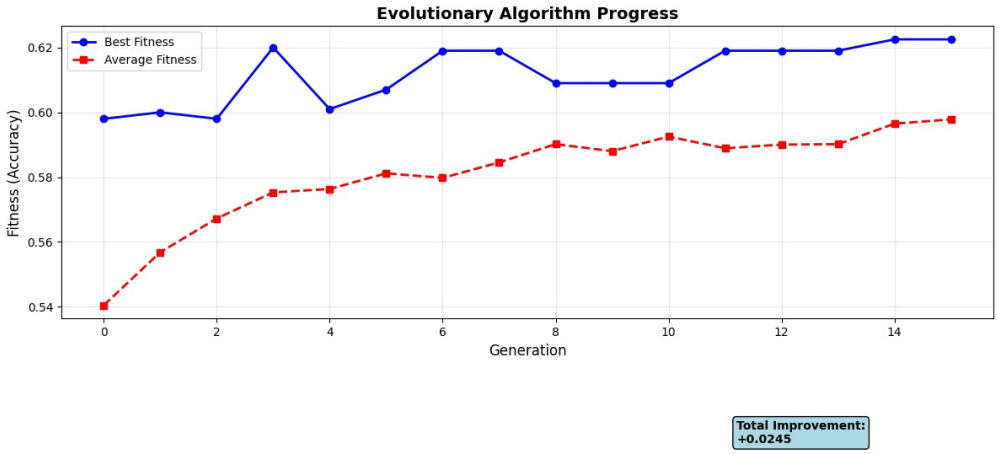

Figure 4.2: Evolutionary Algorithm fitness accuracy with K-Means as the clustering technique.

Using the graph in Figure 4.2, we can notice that the evolutionary algorithm demonstrates excellent convergence characteristics over 15 generations, achieving a total improvement of +0.0245 (2.45 percentage points) from approximately **59.7% initial accuracy to 62.3% final accuracy.** The convergence pattern shows three distinct phases: an initial exploration phase (generations 0-3) with moderate fitness improvements, a rapid improvement phase (generation 3-4) where best fitness jumps from approximately **59.8% to 62.0%**, and a final convergence phase (generations 4-15) with gradual refinement and population convergence.

The algorithm maintains healthy population dynamics throughout the evolutionary process, with average fitness steadily improving from approximately 54.0% to 59.7% while best fitness stabilizes around 62.3%. The gap between average and best fitness in the early stages shows that the average fitness is slowly converging towards the best fitness as the algorithm is able to detect good solutions throughout the population. The convergence in the final phase, where the average and best

fitness are close together, was able to detect a high-quality clustering configuration with a good convergence rate.

The consistency in the improvement found in Figure 4.2 for both the best and average fitness demonstrates that the algorithms have reached a certain level of stability without any oscillating deviations or premature convergence. This stability has also great significance to the evolutionary operators (selection, crossover and mutation) as this means that they are well-calibrated and provide enough exploration while maintaining selective pressure for improvement.

4.2.3 T-SNE Embedding Space Visualization

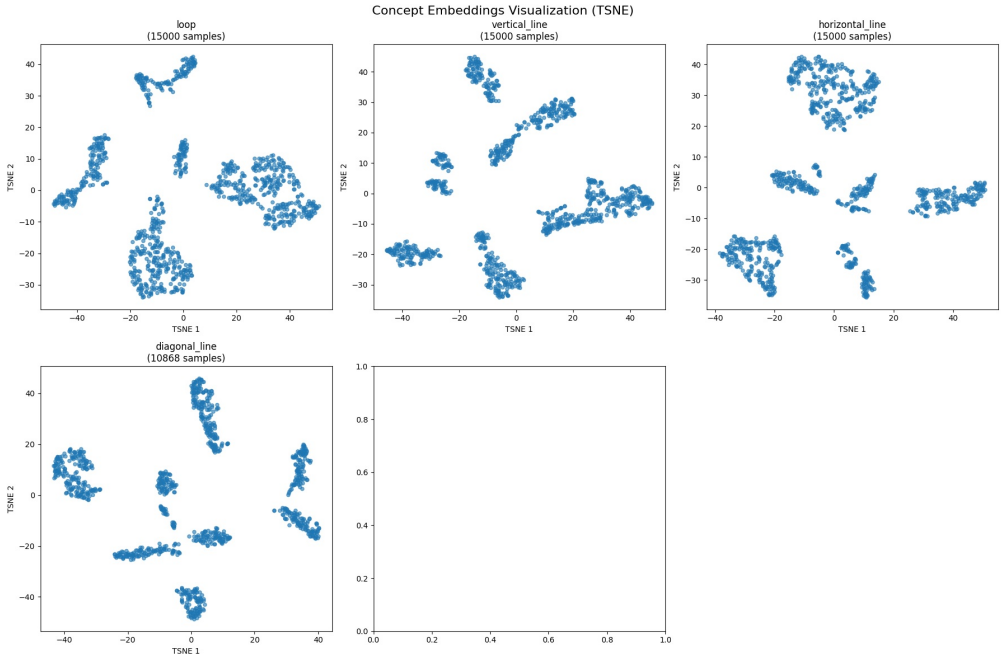

Figure 4.3: Concept Embeddings Visualization using TSNE.

The t-SNE visualizations, present in Figure 4.3, reveal a complex structure within the concept of embedding spaces. Loop embeddings show multiple distinct clusters arranged in various configurations, suggesting that the 64-dimensional embedding space captures meaningful variations in the characteristics of the loop. The clear spatial separation between different regions indicates that similar loop types cluster together while different loop variations occupy distinct regions of the embedding space.

The projection showcased in Figure 4.3 present that the concept bottleneck model

successfully learned to encode meaningful visual patterns into the 64-dimensional embedding space. Each concept has different clustering patterns that show the implicit similarities in structure, with similar samples clustering together and different patterns showing clear separation. This actually shows how effective the concept bottleneck is in learning discriminative embeddings that are valuable to clustering techniques.

The cluster structures in the t-SNE plots in Figure 4.3 verify the decisions that the evolutionary algorithm has taken when it comes to clustering. The obvious separation between different regions suggests that K-Means clustering is able to identify meaningful subdivisions within each concept's embedding space, which resonates well with the selection of cluster counts that was taken by the algorithm.

4.2.4 Classification Performance Analysis

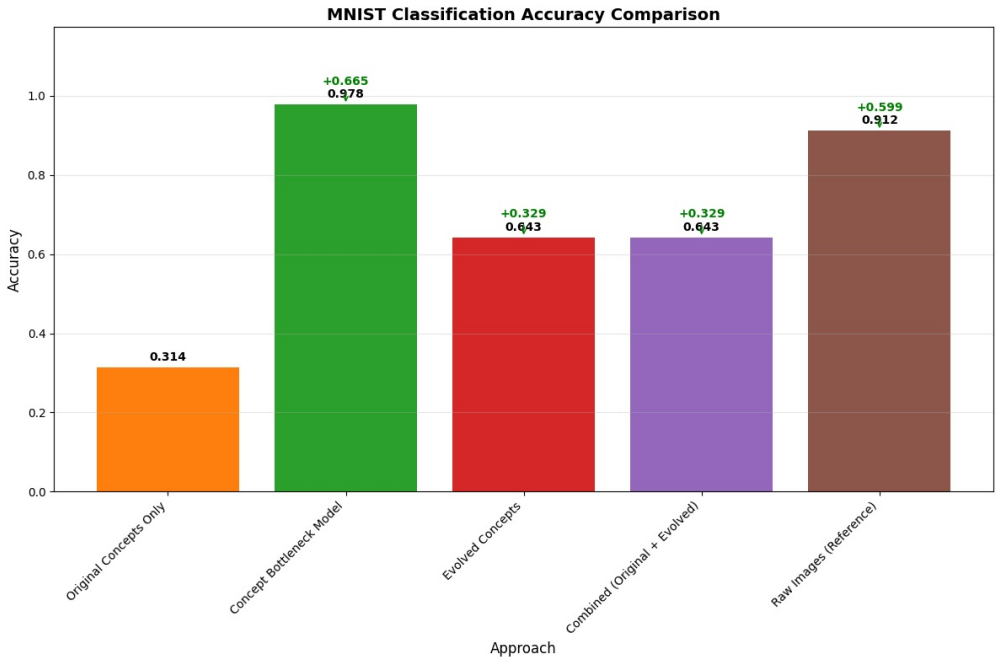

Figure 4.4: Classification Accuracy comparison.

The classification accuracy comparison, that is visualized in Figure 4.4, demonstrates the significant impact of evolutionary concept refinement across multiple evaluation frameworks. Original concepts alone achieve 31.4% accuracy, providing a baseline that reflects the limitations of broad, undifferentiated concept categories. The concept bottleneck model achieves an impressive 97.8% accuracy, validating

the effectiveness of the neural bottleneck architecture for digit classification through concept mediated predictions.

The evolved concepts were able to achieve an impressive **64.3% accuracy**, which actually represents a +32.9-percentage increase over the original five concepts alone. This improvement shows the importance of concept refinement through evolutionary optimization as well as its effectiveness. The combination of evolved and original concepts achieved an identical 64.3% accuracy, which implies that the evolved concept captures and extends the power of the original concepts in discriminating digits, without the literal need of them being present.

The raw images reference achieves 91.2% accuracy using traditional pixel-based classification, providing context for the evolved concept performance. While the evolved concepts (64.3%) do not reach the performance of raw pixel analysis, they provide the crucial advantage of complete interpretability while achieving reasonable classification accuracy. The significant improvement over original concepts (31.4% $\rightarrow$ 64.3%) demonstrate the evolutionary algorithm's success in discovering more discriminative concept subdivisions.

The result demonstrates the trade-off between interpretability and performance. As the concept bottleneck model achieves an optimal accuracy in 97.8% by learning optimal feature representations during what is transmitted as the "black box" approach, while the evolved concepts with 64.3% were able to achieve a good performance using only the predefined visual concept framework. The jump between the evolved concept prediction accuracy and the traditional ones is significant and it represents a step towards reducing the gap between interpretability and performance.

The efficiency of the evolutionary algorithm are represented in these experimental results that further validates the link between initial concepts and improved interpretable ones. The improvement in performance present that is approach is greatly reliable when it comes to the interpretable machine learning realm, while on the other hand the identified clustering configurations show an important understanding around the subdivisions of the concepts. The visuals demonstrate that the evolutionary algorithm effectively finds significant structure in the learned embedding spaces, allowing for the automatic identification of emerging concept subdivisions that improve interpretability while keeping great performances.

4.3 Evolutionary Algorithm Results and Analysis on DBSCAN Clustering

4.3.1 Clustering Configuration of DBSCAN technique

In Figure 4.5 , we can see that the DBSCAN evolutionary algorithm discovered a dramatically different optimal configuration that reflects the density-based clustering paradigm's unique characteristics. The algorithm was able to select a configuration of 12 new loop clusters, 16 generated vertical line clusters, 45 horizontal lines clusters and 8 diagonal lines clusters, meaning that the subdivision of the original concept was higher than the one that we have gotten from K-Means clustering. This increase in clustering is mainly due to the fact that DBSCAN determines clusters count based on data density rather than predetermined numbers, leading to more fine-grained subdivisions of the embeddings.

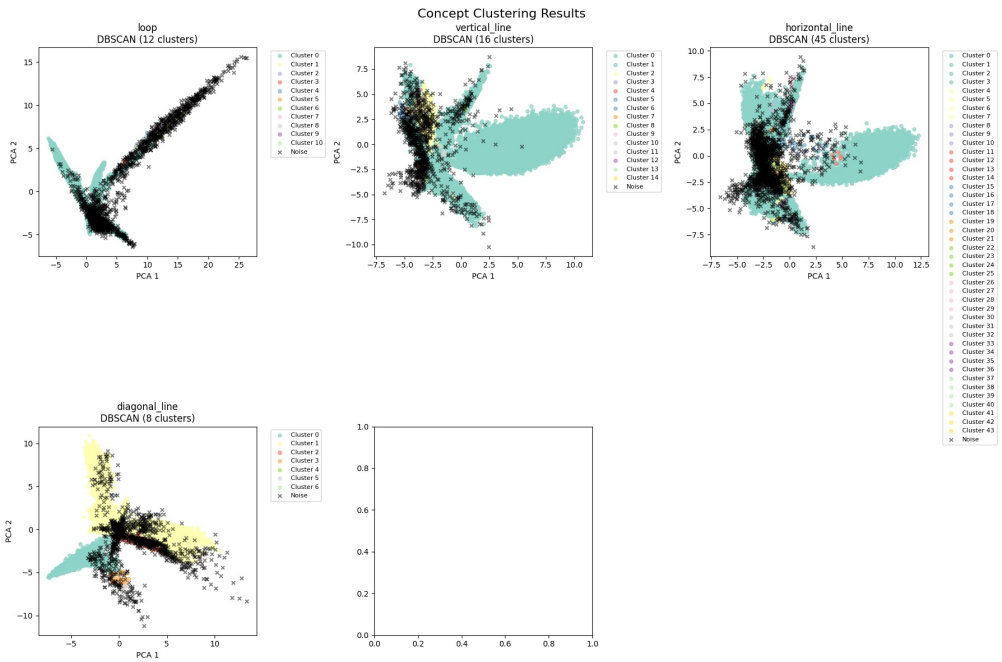

Figure 4.5: Concept Clustering based on DBSCAN technique.

- **Loop Concept DBSCAN Analysis:** The PCA graph present in Figure 4.5 present a completely different clustering structure with respect to K-Means, as it was able to identify 12 distinct density regions within the embedding space of the loop concept. The clustering pattern shows primarily linear-based regions as a lot of clusters extend diagonally across the embedding space, with

a significant number of noise points that are marked with the x symbols, that represent samples that don't fit into well-defined density clusters. This pattern demonstrates that loop embeddings contain very complex density variations that DBSCAN captures through its density-based technique, however the significant noise indicates that not all loop samples abide by these clusters.

- **Vertical Line Concept DBSCAN Analysis:** The vertical line concept presents the most complex structure with 16 clusters distributed across multiple regions of the embedding space. The graph in the Figure above shows that the clusters can have multiple sizes, as some form compact groups while others appear elongated. The distribution of noise marked by a black x shows that the embeddings are heterogeneous which defies the concept of density-based clustering. We can deduce that DBSCAN is sensitive to local density variations within the vertical line embedding distributions due to the large number of clusters.

- **Horizontal Line Concept DBSCAN Analysis:** The graph showcases a highly fragmented embedding space as the clustering granularity of the horizontal line concept presents 45 distinct clusters, which means the subdivision of the embedding space is really detailed. The visualization displays several different small clusters scattered across the embedding space with a lot of noise data points. This extreme division of the original concept demonstrates that the DBSCAN clustering technique identifies small and local density regions rather than large-scale ones, which may lead to overfitting or excessive fragmentation.

- **Diagonal Line Concept DBSCAN Analysis:** The diagonal line concept shows more positive results considering it was able to find 8 clusters arranged in different regions. The graph details a less fragmented representation of the embedding space compared to other concepts, with clearly fewer noise points. This result implies that the embeddings of this concept are more coherent in terms of density and consequently suit well the DBSCAN approach, even though the frequency of diagonal lines found in the dataset was lower than the threshold, as presented in the annotator.

4.3.2 Evolutionary Algorithm Performance for DBSCAN Clustering

Using the fitness visualization presented in Figure 4.6, we can realize that the DBSCAN evolutionary algorithm demonstrates more volatile convergence characteristics compared to K-Means, with best fitness oscillating between approximately 36.8% and 39.3% over 15 generations while achieving a total improvement of

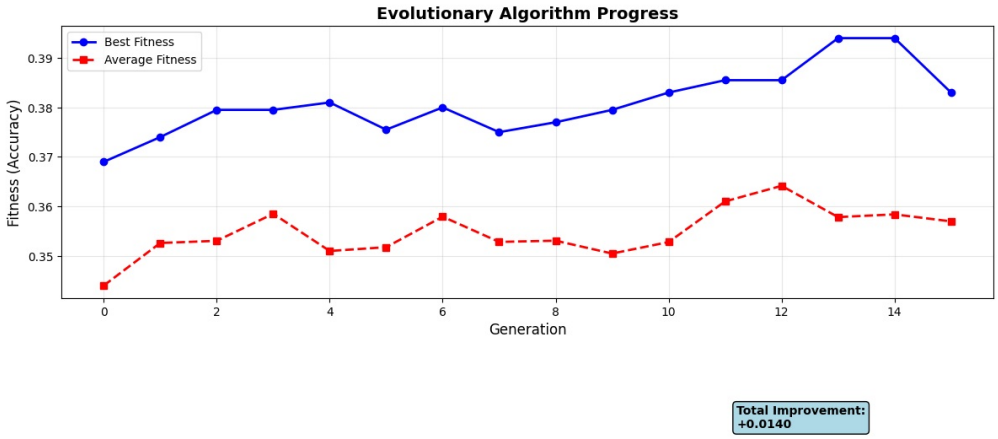

Figure 4.6: Evolutionary Algorithm Performance for DBSCAN

+0.0140 (1.4 percentage points). The convergence pattern shows less stability than K-Means, with multiple local peaks and valleys that suggest a more complex optimization landscape. The best fitness reaches a maximum around generation 13-14 before declining slightly in the final generation, indicating potential overfitting or instability in the density-based clustering approach.

In fact, the average fitness shows a limited improvement, remaining stable around 35 – 36% throughout most of the generations of the evolutionary algorithm while also presenting more oscillation than the K-Means approach. We can detect that the smaller gap between the best and the average fitness, compared to K-Means suggests a more limited optimization where DBSCAN parameters have less effect on performance. We witness that the population finds it difficult to improve with consistency, showing a few periodic setbacks that demonstrate how challenging it is to optimize density-based clustering.

The DBSCAN technique was able to achieve a smaller improvement (+1.4 percentage points) compared to the K-Means improvement that was analyzed before, and we can strongly interpret that the density-based clustering may be less effective for this specific task. And thus, we can strongly confirm that DBSCAN's density-based technique may not align well with the embedding space learned by the concept bottleneck model.

4.3.3 T-SNE Embedding Visualization

We can see through the t-SNE visualizations presented in Figure 4.7 that DBSCAN identifies different patterns to the ones that were displayed in K-Means clustering.

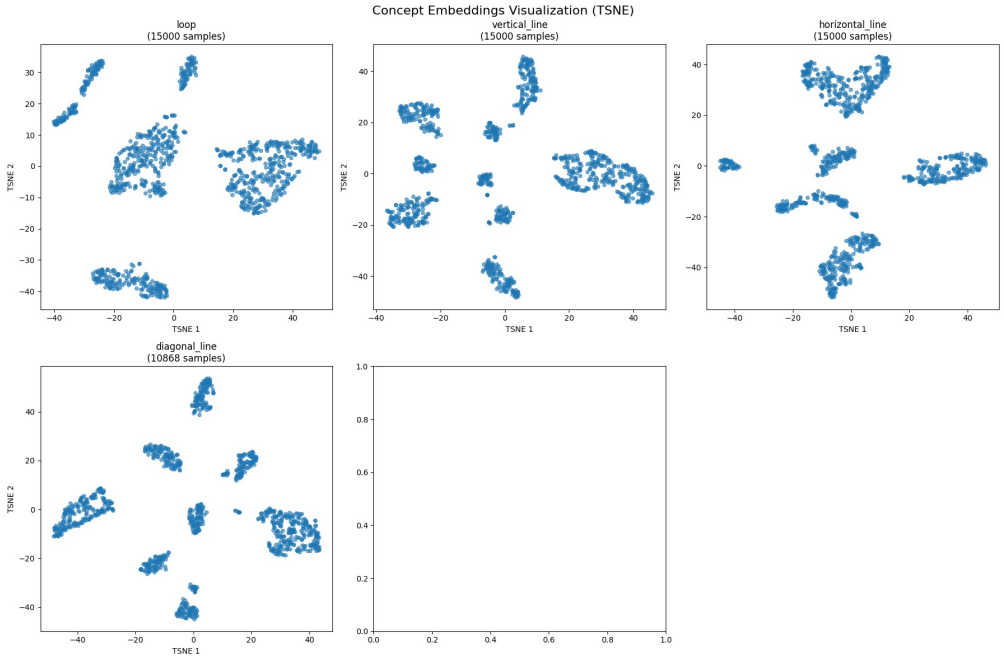

Figure 4.7: Concept Embedding Visualization using TSNE

For example, the horizontal embeddings show multiple isolated cluster regiones separated by great space, this comes as an advantage to DBSCAN clustering techniques as it bases its algorithm on density accumulation. However, the considerable amount of noise that is present in all concept visualizations shows that many embeddings do not fit well density clusters which limits the effectiveness of DBSCAN. Although there is an advantage in excluding the outliers from the clusters that do not fit them, this may also be counter efficient as it may reduce the total number of samples that have meaningful concept refinement, which may consequently limit the evolutionary optimization process.

4.3.4 Classification Performance Analysis

The DBSCAN approach demonstrates major difference in terms of performance compared to K-Means clustering. From Figure 4.8, we can notice that the classification accuracy comparison reveals that **original concepts** achieve **31.4% accuracy** as the baseline, while the concept **bottleneck model** maintains its impressive **97.9% performance** (+66.6 percentage point improvement), which confirms that the neural network architecture is consistent across different clustering approaches. However, the **evolved concepts using DBSCAN** achieve **36.2% accuracy**, representing a more modest +4.9 percentage point improvement over the original

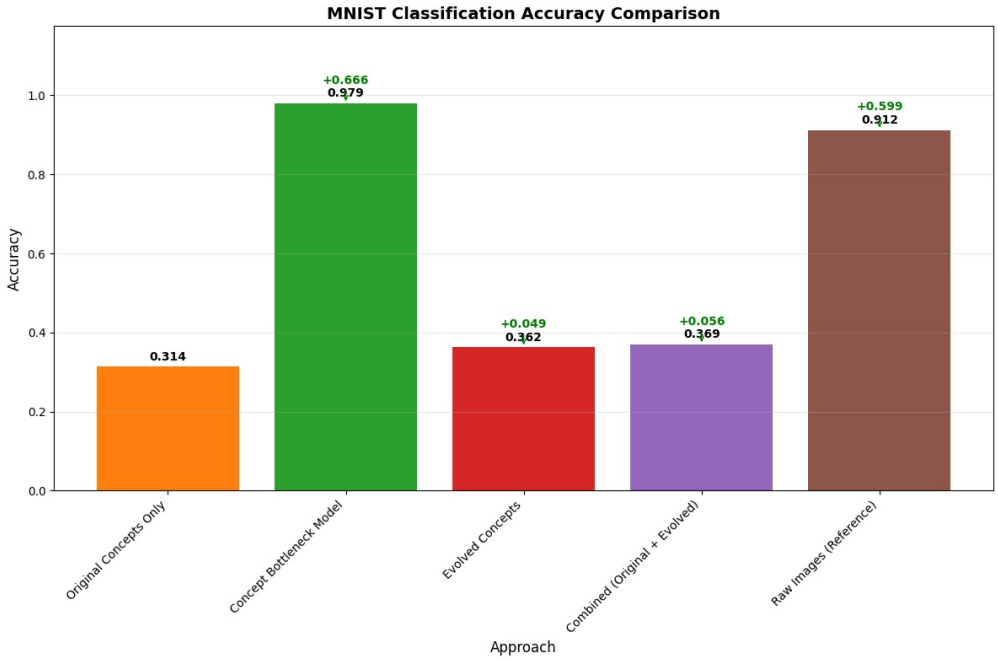

Figure 4.8: Classification accuracy based on the DBSCAN clustering embeddings

concepts baseline. The **combined approach (original + evolved)** achieves **36.9% accuracy** with a +5.6-percentage point improvement, showing minimal additional benefit from combining original and evolved concepts. The **raw images reference maintains 91.2% accuracy** (+59.9 percentage points), providing consistent performance context across both clustering methodologies, same as the concept bottleneck model.

In conclusion, these results reveal important trade-offs between the two clustering approaches. K-Means achieves superior performance improvement (+32.9 percentage points vs +4.9 percentage points for evolved concepts) and demonstrates more stable evolutionary convergence, suggesting better alignment with the embedding space characteristics. DBSCAN provides more ambiguous cluster discovery with automatic granularity determination and noise detection but achieves slight performance improvements and shows greater optimization volatility. This implies that the concept embeddings learned by the model may have some characteristics that are more suitable to a centroid-based clustering technique rather than a density-based one. Since the performance of K-Means was extremely superior to the one of DBSCAN, we can also deduce that the concept embeddings are a compact with a spherical distribution and that a density-based method for clustering can present restrictions for the embedding space structure, leading to our findings of

abundant noise points.

Chapter 5

Conclusion

The research conducted in thesis was able to demonstrate the potential that evolutionary algorithms have when it comes to discovering newly emerged concepts, where these new concepts can enhance both interpretability and classification performance in deep neural networks. Abiding by a detailed three-stage pipeline that contains visual concept annotator, the concept bottleneck model and evolutionary concept algorithm, this work presented the fundamental obstacle related to balancing interpretability while weighing on discriminative power in explainable artificial intelligence.

The research approach progressed using a systematic approach related to interconnected phases within a pipeline. The visual concept annotator was able to successfully transform raw MNIST digit images into binary concept annotations by picking five fundamental visual concepts, while achieving 25% detection rates for loops, vertical lines and horizontal ones using adaptive thresholding. Then the concept bottleneck was able to achieve a remarkable accuracy of 97.8% for digit classification and an average accuracy of 48.6% for the concept prediction accuracy. The bottleneck was able to demonstrate that its constraints preserve classification performance while ensuring complete interpretability.

Next, with the execution of the evolutionary algorithm, we were able to discover optimal emerged concepts through clustering refinement, this phase constituted the most innovative section in our research. The K-Means clustering technique was able to find an optimal configuration for the five original concepts that resulted in a substantial increase in the classification accuracy of the digits arriving at 64.3% compared to the baseline of 31.4%. While, on the other hand, DBSCAN achieved a more modest improvement of +4.9 percentage points due to fragmentation and noise in the embedding space.

Speaking of the contributions made within this project, it is important to include the integration of evolutionary algorithms with concept bottleneck to discover concepts automatically. Additionally, the implementation of a bottleneck constraint that ensures complete interpretability all while demonstrating significant performance improvements and preserving explainability. We can also consider the development of a robust pipeline that focuses on evolutionary concept optimization as a significant contribution made by this work.

In the end, we can confidently assert that this project was able to establish evolutionary concept optimization as a promising direction for interpretable machine learning, while providing an automated approach that is independent from manual concept specification. The notable performances that were detailed concerning what was achieved with K-Means clustering show the credibility around the usage of evolutionary approaches for reducing the gap between interpretability and the strength of neural networks. Although the evolved concepts did not achieve "black box" level of accuracy, they provided nevertheless great improvements over the baseline representation while keeping the interpretability and establishing a great foundation for future research in discovering concepts automatically and optimizing concepts through evolutionary algorithms for explainable artificial intelligence.

Chapter 6

Future Research

Moving forward, further research directions should address several challenges and limitations that were encountered during the diverse implementations of this project. However, we need to also consider the plethora of opportunities that can be leveraged as well in order to improve on the conducted research in order to obtain more optimal results that certainly improve the confidence in our initial hypothesis and our developed solution as a whole.

Essentially, the curve detection algorithm needs to be redesigned in order to address the saturation that was encountered and to provide an important asset that was missing which is a meaning discriminative power. This can be done by finding alternative curvature calculations methods or normalization techniques that capture detailed variations of the curves without reaching any saturation of the algorithm.

After the modest results that were found using DBSCAN as a clustering technique, it is also important to investigate alternative approaches such as Gaussian Mixture Model, spectral clustering or hierarchical clustering methods that may provide us with improved results in terms of refined concepts, as these techniques may be better aligned with our embedding space characteristics.

One of the most important targets that need to be considered in any further research is expanding the concepts beyond the structural explainable results that we have found. In fact, it is extremely important to emphasize on how semantic-level representations could improve interpretability and applicability to more complex visual domains outside of the restrictions of handwritten digits.

In order to provide our three-stage pipeline to become more generalized and scalable, it is crucial to broaden the scope of which the pipeline is applicable by

testing it on larger and more complex datasets such *CIFAR-10* or *CUB-200-2011* as it would add more validation and credibility to our experimentation and thus our pipeline implemented.

Instead of finding one optimal solution that maximizes the performance of the evolutionary algorithm based on classification accuracy, we could consider finding multiple optimal solutions that offer different balance between being accurate with comprehensive meanings, letting users to choose what is more important for their specific application and by that a multi-objective evolutionary algorithm would be useful to find a better balance in the trade-off between interpretability and performance.

Finally, rather than letting the algorithm to blindly discover concepts based only on mathematical calculation that represent the core principle of the evolutionary algorithm itself, an expert user could be shown the discovered concepts in order to distinguish what align with its logical interpretation of what the result should or should not be. This ensures that the final concepts are not just mathematically optimal but also correlated with how human naturally think and categorize visual patterns. Thus, the implementation of a human feedback mechanism, although time consuming, could make the emerged concepts more meaningful and purposeful.

Bibliography

[1] D. Kazhdan, B. Dimanov, M. Jamnik, P. Liò, and A. Weller. «Now You See Me (CME): Concept-based Model Extraction». In: *CIKM 2020 Workshops* (2020). Available: `https://arxiv.org/abs/2010.13233` (cit. on pp. 6–8, 10, 11).

[2] P. Barbiero, G. Ciravegna, and A. Tonda. «Interpretable Neural-Symbolic Concept Reasoning». In: *Proc. International Conference on Machine Learning (ICML)*. 2023 (cit. on pp. 7–9, 11, 12).

[3] S. Luke. *Essentials of Metaheuristics*. Available: `https://cs.gmu.edu/~sean/book/metaheuristics/`. Lulu, 2009 (cit. on p. 13).

[4] A. Slowik and H. Kwasnicka. «Evolutionary Algorithms and Their Applications to Engineering Problems». In: *Neural Computing and Applications* (2020). Available: `https://doi.org/10.1007/s00521-020-04832-8` (cit. on p. 13).