ALGERIAN DEMOCTATIC AND POPULAR REPUBLIC

MINSTIRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

KASDI MERBAH UNIVERSITY OUARGLA

FACULTY OF NEW TECHNOLOGIES OF INFORMATION AND CUMMUNICATION

DEPARTMENT OF CUMPUTER SCIENCE AND INFORMATION TECHNOLOGY

ACADEMIC MASTER thesis

Domain: Computer science and information technology

Field: Computer Science

Specialty: Artificial Intelligence and Data Science

By Mohammed Aymen Benzine & Issam Eddine Kiouer

THEME

Improving Brain Tumor Classification Using Knowledge Distillation

Publicly Discussed on: 12/06/2025

Before the Jury:

Dr. Khadra Bouanane	Examiner	UKM Ouargla
Dr. Oussama Aiadi	Supervisor	UKM Ouargla
Dr. Aicha Korichi	President	UKM Ouargla

Academic Year: 2024/2025

Acknowledgment

First and foremost, we praise and thank Almighty Allah, the Most Gracious, the Most Merciful, for granting us the strength, wisdom, and perseverance to complete this work. Without His blessings, this achievement would not have been possible.

We express our deepest gratitude to our supervisor, Dr. Oussama Aiadi, for his invaluable guidance, continuous encouragement, and insightful critiques throughout this research journey. His unwavering support and confidence in our abilities motivated us to overcome challenges and refine our work. We are truly honored to have learned under his supervision.

Our sincere appreciation goes to the distinguished committee members for dedicating their time to review this thesis. We are humbled by their participation and grateful for their constructive feedback, which will undoubtedly enhance the quality of our research.

We would also like to extend our thanks to all our professors and teachers who have shaped our academic journey, imparted knowledge and inspiring us to strive for excellence. Their mentorship has been instrumental in our growth.

No words can fully capture our gratitude to our beloved parents may Allah bless them with long, healthy lives filled with happiness. Their endless love, sacrifices, and prayers have been our greatest source of strength. We pray that we may always honor and make them proud.

A heartfelt thanks to our family and friends for their unwavering emotional support, patience, and encouragement during this demanding yet rewarding process. Their presence has been a constant motivation.

Lastly, we acknowledge everyone who contributed, directly or indirectly, to the completion of this thesis. May Allah reward them abundantly.

Dedication

It is with deepest love and appreciation that I dedicate this work to my beloved parents my father, may Allah protect him and grant him good health, and my dear mother, may Allah prolong her life with happiness. Their unwavering support, endless sacrifices, and constant prayers have been my guiding light throughout this journey.

I also extend my sincere gratitude to all my professors and the department faculty whose knowledge and mentorship have shaped my academic growth. A special acknowledgment goes to my supervisor, Dr. Oussama Aiadi, for his patient guidance, valuable insights, and steadfast encouragement during every stage of this research.

To my siblings and friends your emotional support and belief in me have been invaluable. This accomplishment belongs as much to you as it does to me.

May Allah accept this humble effort and reward all those who contributed to its realization.

Abstract

The increasing demand for deploying deep learning models on mobile and edge devices has brought the need for lightweight and efficient neural networks to the forefront of research. This is particularly crucial in medical imaging, where real-time and accurate diagnostic support is required. However, most high-performing models rely on large architectures, making them impractical in constrained environments.

This thesis proposes an enhanced knowledge distillation (KD) framework that aims to build compact student models while preserving high diagnostic performance for brain tumor classification. The approach leverages a training-free student selection method based on the DisWOT score and introduces two key enhancements: the use of Layer-wise Relevance Propagation (LRP) instead of Grad-CAM for fine-grained semantic supervision, and cosine similarity instead of L2 loss for robust and scale-invariant feature alignment.

Our experimental evaluations on a publicly available brain MRI dataset demonstrate that our improved KD pipeline significantly lessens the quantity of parameters and computation cost with minimal impact on accuracy of the teachers. Our best student model achieved 95.35% classification accuracy, while reducing over 85% of the parameters in the teacher model. Our results substantiate the use of high-order distillation signals for generalization and training stability.

In conclusion, the proposed framework provides an effective and efficient solution for interpretable lightweight model development, and has great potential for use in real-world contexts such as in medical or other constrained resource scenarios.

Keywords: Knowledge Distillation (KD), Lightweight Models, Brain Tumor Classification, Layer-wise Relevance Propagation (LRP), Cosine Similarity, DisWOT, Deep Learning, Medical Imaging.

Résumé

La demande croissante pour le déploiement de modèles d'apprentissage profond sur des appareils mobiles et embarqués a mis en avant la nécessité de réseaux de neurones légers et efficaces. Cela est particulièrement crucial dans le domaine de l'imagerie médicale, où un support diagnostique en temps réel et précis est essentiel. Cependant, la majorité des modèles performants reposent sur des architectures volumineuses, ce qui les rend peu pratiques dans des environnements contraints en ressources.

Ce mémoire propose un cadre amélioré de distillation des connaissances (Knowledge Distillation - KD) visant à concevoir des modèles étudiants compacts tout en préservant des performances élevées pour la classification des tumeurs cérébrales. L'approche repose sur une méthode de sélection des étudiants sans entraînement, basée sur le score DisWOT, et introduit deux améliorations clés : l'utilisation de la propagation de pertinence par couche (LRP) au lieu de Grad-CAM pour une supervision sémantique fine, et la similarité cosinus à la place de la perte L2 pour un alignement robuste et invariant à l'échelle.

Nos évaluations expérimentales sur un ensemble de données d'IRM cérébrales accessible publiquement démontrent que notre pipeline KD amélioré réduit significativement le nombre de paramètres et les coûts de calcul, tout en maintenant une précision de classification élevée. Le meilleur modèle étudiant a atteint 95,35 % de précision tout en réduisant plus de 85 % des paramètres du modèle enseignant. Ces résultats confirment l'efficacité des signaux de distillation de haut niveau pour améliorer la généralisation et la stabilité de l'entraînement.

En conclusion, le cadre proposé constitue une solution évolutive pour la conception de modèles légers interprétables, avec un fort potentiel d'application dans des contextes réels, notamment en médecine ou dans d'autres environnements à ressources limitées.

Mots-clés : Distillation des Connaissances (KD), Modèles Légers, Classification des Tumeurs Cérébrales, Propagation de Pertinence par Couche (LRP), Similarité Cosinus, DisWOT, Apprentissage Profond, Imagerie Médicale.

الملخص

إن الطلب المتزايد على نشر نماذج التعلم العميق على الأجهزة المحمولة والحواسيب الطرفية قد سلط الضوء على الحاجة إلى تصميم شبكات عصبية خفيفة وفعالة. وتعد هذه الحاجة أكثر إلحاحا في مجال التصوير الطبي، حيث تعتبر الدقة والسرعة في التشخيص من العوامل الحيوية. ومع ذلك، فإن معظم النماذج ذات الأداء العالي تعتمد على هياكل معمارية كبيرة، مما يجعلها غير مناسبة في البيئات ذات الموارد المحدودة.

يقترح هذا البحث إطارا محسنا لـ تقطير المعرفة(Knowledge Distillation - KD) ، يهدف إلى بناء نماذج طلابية مدمجة تحتفظ بأداء تشخيصي عالٍ في تصنيف أورام الدماغ. وتعتمد الطريقة على اختيار الطالب بدون تدريب باستخدام مقياسDisWOT ، وتقدم تحسينين رئيسيين: استخدام طريقة LRP (انتشار الصلة حسب الطبقات) بدلا من Grad-CAM لتوفير إشراف دلالي دقيق، واعتماد تشابه جيبي (Cosine Similarity) بدلا من فقدان L2 لتحقيق توافق أكثر استقرارا وغير حساس للحجم.

أظهرت التقييمات التجريبية على مجموعة بيانات صور رنين مغناطيسي للدماغ (MRI) مفتوحة المصدر أن الإطار المقترح يقلل بشكل كبير من عدد المعاملات وتكلفة الحوسبة مع تأثير طفيف فقط على الدقة. حيث حقق أفضل نموذج طلابي دقة تصنيف بلغت 95.35%، مع تقليل عدد المعاملات بأكثر من 85%مقارنة بالنموذج المعلم. وتؤكد النتائج فعالية إشارات التقطير عالية المستوى في تحسين التعميم واستقرار التدريب.

في الختام، يوفر هذا الإطار المقترح حلا قابلا للتوسع لتطوير نماذج خفيفة وقابلة للتفسير، وله إمكانيات كبيرة للتطبيق في السياقات الواقعية مثل المجالات الطبية أو البيئات ذات الموارد المحدودة.

الكلمات المفتاحية: تقطير المعرفة(KD) ، النماذج الخفيفة، تصنيف أورام الدماغ، انتشار الصلة حسب الطبقات (LRP)، التشابه الجيبي(Cosine Similarity) ، DisWOT، التعلم العميق، التصوير الطبي.

Contents

List of Figures

List of Tables

General Introduction

1. Introduction

In recent years, the rapid growth of artificial intelligence (AI), especially deep learning, has transformed many areas such as healthcare, autonomous systems, finance, and natural language processing. One of the greatest advances has been the use of Convolutional Neural Networks (CNNs) for image classification and image pattern recognition with outstanding results in medical image analysis, which is a field that values speed, efficiency and accuracy.

However, even with their great accuracy, CNNs are very computational and memory intensive, which leads to challenges with real-time deployment, especially in resource-limited environments like mobile devices, embedded devices or edge computing systems, These limitations are even more pronounced in medical settings, where fast and accurate diagnostic tools are needed at the point of care. In such contexts, deploying high-performance models without relying on powerful computing infrastructure becomes a critical necessity.

To meet this demand, there has been a growing interest in the development of lightweight neural networks that aim to reduce model complexity without sacrificing performance. Models like MobileNet, ShuffleNet, and EfficientNet represent key milestones in this direction. Yet, even these compact architectures can struggle to match the performance of their larger counterparts, particularly when trained on small or imbalanced datasets an issue commonly encountered in the medical domain.

A promising solution to this problem lies in Knowledge Distillation (KD) a model compression technique that allows a large, high-capacity model (the "teacher") to transfer its learned knowledge to a smaller, more efficient model (the "student"). Originally proposed as a method to transfer the soft output probabilities from the teacher to the student, traditional KD has evolved significantly. While early methods relied on mimicking the output logits or hard labels of the teacher model, modern approaches explore richer and more informative strategies, such as transferring intermediate feature representations, attention maps, or relational structures between layers.

These advanced KD techniques go beyond surface-level outputs and delve into the internal behavior and semantic understanding of the teacher model. By guiding the student model to imitate not only what the teacher predicts but also how it processes and understands the input data, KD enables the student to achieve performance levels close to or even surpassing the teacher, all while maintaining a lightweight structure. This is especially beneficial in medical image classification, where a balance must be struck between model interpretability, inference time, and accuracy.

2. Problematic

Although KD has proven effective for model compression, existing approaches still suffer from several limitations:

- **Limited semantic guidance**: Techniques such as Grad-CAM used in semantic supervision often produce coarse and noisy activation maps, which may misguide the student model during training.

- **Scale sensitivity in similarity computation**: Common metrics like L2 loss are highly sensitive to magnitude differences, penalizing the student even when semantically aligned with the teacher.
- **Underperformance in small-data regimes**: Many KD frameworks rely on extensive training, making them impractical for few-shot learning scenarios commonly encountered in the medical domain.

Hence, our work addresses the following central question:

How can we improve the quality of knowledge transfer in KD to train high-performing lightweight models, while enhancing semantic alignment and reducing computational overhead and keeping the same performance?

3. Overview of The Related Techniques

To address the limitations of traditional knowledge distillation methods, several approaches have been proposed in recent years. One of the most foundational methods is Traditional Knowledge Distillation, introduced by Hinton et al. [1], which relies on softened probability distributions from the teacher model to guide the training of a smaller student model. This method effectively transfers class-level knowledge but does not account for deeper internal representations.

Another category is feature-based KD, which attempts to enhance supervision by transferring intermediate layer outputs from the teacher to the student using similarity losses such as L2 or Mean Squared Error (MSE) [2]. Although this method leverages deeper feature knowledge, it is sensitive to scale and may mislead optimization if the activations differ in magnitude.

A more recent line of work explores training-free knowledge distillation, such as DisWOT [3], which uses attention-based similarity metrics like Grad-CAM combined with L2 alignment. This allows the selection of student candidates from a defined search space without training, based on semantic and relational similarity to the teacher. However, such approaches often produce coarse and noisy maps that may not offer sufficient semantic precision for effective knowledge transfer.

To optimize the selection of student models, Neural Architecture Search (NAS) [4] is employed. NAS automates the discovery of suitable student architectures by evaluating candidates across various configurations, balancing between accuracy and efficiency. Although powerful, NAS can be computationally intensive and requires careful design to be practical.

Moreover, research on lightweight architectures such as MobileNetV2 has introduced structural innovations (e.g., depthwise separable convolutions and bottleneck layers) to reduce the total number of parameters and computational cost for maintaining performance. These models are very suitable for situations where resources are limited.

Despite significant advancements, the methods presented above still have limitations, including their inability to use fine-grained semantic alignment and unstable supervision when applied to deep feature spaces. To overcome these limitations, we offer a new framework that substitutes Grad-CAM with Layer Wise Relevance Propagation (LRP) [5], for improved semantic clarity and interpretability, and replaces L2 with cosine similarity leading to scale-invariant and stable feature

alignment. Together, these enhancements serve as the foundation for our extended knowledge distillation strategy.

4. Contributions

The main contributions of this thesis can be summarized as follows:

- A refined knowledge distillation framework that integrates LRP-based semantic supervision and cosine similarity to enhance the student model's feature alignment with the teacher.
- A training-free evaluation approach using DisWOT scores to select optimal student models based on structural and semantic similarity.
- Design and evaluation of lightweight student architectures, showing significant reductions in FLOPs, MACs, and parameters with minimal accuracy drop.
- Comprehensive experimentation on a brain tumor classification dataset, demonstrating the superiority of the proposed KD pipeline over baseline and standard KD setups.

5. Structure of the Thesis

This thesis is structured as follows:

- Chapter 1: Provides background on deep learning, computer vision, lightweight architectures, and an overview of Knowledge Distillation and its variants.
- Chapter 2: Details the proposed KD framework, including the enhancements via LRP and cosine similarity.
- Chapter 3: Presents experimental results, including dataset description, evaluation metrics, model comparisons, and lightweighting performance.
- Finally, we will draw the general conclusion of the thesis, highlighting the main outcomes and discussing potential future work and research directions.

Chapter 1

Work Background

1.1 Introduction

In recent years, there have been significant technological advancements in artificial intelligence, as well as new applications emerging across sectors, with deep learning and computer vision at the forefront. Deep learning has revealed outstanding performance in modeling complex representations from large volumes of data, making it a useful tool for a wide variety of tasks in machine learning applications like image classification, segmentation, and pattern recognition. This chapter takes an overview of the principles of deep learning and machine learning, emphasizing, in particular, convolutional Neural Networks (CNNs), computer vision methods, and the importance of lightweight models. In addition, the chapter explores learning paradigms (e.g., supervised, unsupervised, and reinforcement learning), while focusing on modern methods of optimizing deep models for performance in general, and embedding for efficiency in applied scenarios (e.g., on mobile or edge devices, and low-resource situations, e.g., radio or cellular).

1.2 Machine Learning

1.2.1 Definition

Machine learning is a branch of Artificial Intelligence that permits computers to learn from information and improve themselves without having been directly programmed to perform the given job. In other words, ML makes it possible for systems to predict or make a choice based on data.

The concept of machine learning was introduced by Arthur Samuel in 1959 [6], who described it as “the field of study that gives computers the ability to learn without being explicitly programmed”. Later, Tom M. Mitchell (1997) provided a more formal definition, describing ML as: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E” [7].

1.2.2 Paradigm of Learning

The paradigm of learning refers to the approach through which a model acquires knowledge from data. Four primary paradigms are widely recognized:

Supervised learning is a form of machine learning where models learn from an existing dataset where the correct answers, or labels, are already known. In the supervised learning process, each example has an input plus the expected output. The idea is to teach how to predict correctly by learning from the data and the associations between the input and outputs. To determine how well a model is performing we employ a loss function, or loss measure, that shows how far off a prediction is from the correct answers [8]. Supervised learning is widely used for classification

projects (such as identifying objects in images) along with regression tasks (such as predicting housing prices).

Unsupervised learning is the method of educating the algorithms using datasets that are not labeled overtly; hence, it involves no supervised training. Data without a labeled output [9] can be defined as unsupervised learning. Unsupervised learning necessitates algorithms to recognize patterns, structures, or correlations in data autonomously. Clustering stands as one of the core unsupervised learning themes of machine learning because it serves as the key way by which data dimensionality and complexity can be minimized through a process of categorizing and grouping relevant data pieces. Such a method can help to discover the underlying structure of data spread, find consistent patterns, and identify multiple types of correlations. A huge number of modern applications and data analysis benefit from this method by which engineers of various domains can easily uncover these relations and even derive actionable knowledge from them. Clustering is related to the use of data analysis, the construction of the visual model, and the setting up of the decision-making process by all these actions.

Semi-supervised learning is a different matter altogether in the area of machine learning, as it takes place between supervised and unsupervised models. This method involves training a model with a small number of labeled data and a large number of unlabeled data. [10] The basic objective of semi-supervised learning is the same as that of supervised learning; it is to create a function that can give a reliable prediction of the output variable from the input variables. However, what sets semi-supervised learning apart is the use of a dataset that contains both labeled and unlabeled data of which only a few examples of domain are known [10] .

Semi-supervised learning is particularly beneficial if a vast number of unlabeled data exists but it is not practical to provide labels for them all. Semi-supervised learning is an attempt to employ the advantages of both labeled and unlabeled data, for example, these unlabeled data were merged with a smaller labeled set of data, so as to gain a compromise between data efficiency and model accuracy. This approach mainly offers many real-world machine learning problems with a feasible and cost-effective choice.

Reinforcement learning occupies a niche in the area of machine learning which is concerned with how agents take their turns in interactive situations. During this process, the agent learns to improve his/her behavior in order to maximize a reward received in the future, over time.

The RL is defined as a framework for learning optimal behavior through interaction with one's surroundings [11] has been extracted from the literature. Where the agent learns to make optimal decisions by interactions with its environment. The agent aims to maximize cumulative rewards by exploring and exploiting its surroundings.

1. **Agent:** the decision-maker who engages with surroundings.
2. **Environment:** The external source, which is characterized by agent's functions
3. **State:** Representations of the current circumstance or environment setting
4. **Actions:** The set of possible activities or decisions to be taken in each stage by the agent.
5. **Rewards:** Numerical indications that tell the agent how well it is performing in an area which are produced in response to its actions.

Each of these paradigms plays a crucial role in shaping how artificial intelligence systems evolve and improve over time, serving as foundational strategies for training modern machine learning models.

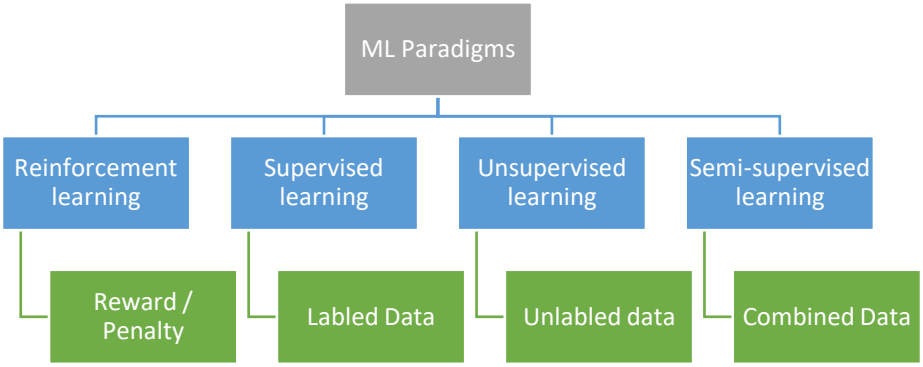


Figure 1.1: Machine learning Paradigms

1.3 Deep Learning

Deep learning (DL), is a branch of machine learning that employs artificial neural networks, which are imitation for the man's brain. The goal of the networks is to learn through improving with data [12]. In contrast to previous machine learning forms, which almost always requires human feature selection identifying key value indicators, deep learning approaches make use of models which scan the raw data and extract features automatically. This results in lower human intervention and domain expertise.

The "deep" in DL refers to the multi-layered architecture of these networks, a key factor that allows them to learn and extract features from large amounts of data, enabling them to perform complex tasks such as image recognition, natural language procession (NLP) [13], and speech recognition.

An example of such type of neural network is an artificial neural network with one or more hidden layers known as multi-layer perceptron. These layers are fully connected and sit between the input and output. The capability of the model to detect various patterns from a given set of data solely relies on the number of neurons (the width of the hidden layers) and the number of layers (the depth of the network) [14]. Presence of a wide layer is tremendous when it comes to the model recognizing more complex features, while deepening the level of networking assists the model in understanding the structure of the given data.

1.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs), unlike traditional algorithms, are specifically designed to handle visual data such as images and videos. As a type of deep learning model, CNNs have transformed the field of computer vision by automatically extracting hierarchical features

directly from raw pixel data, enabling more accurate and efficient analysis of visual content [15].

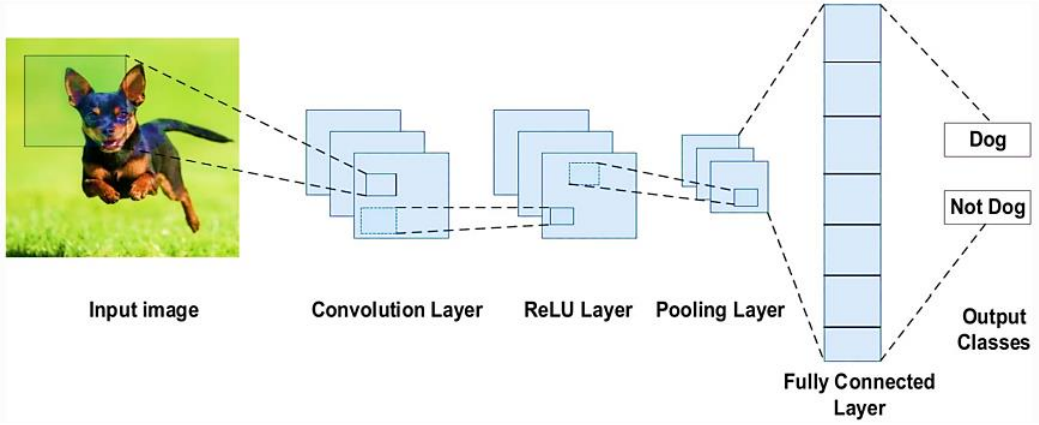


Figure 1.2: Convolutional Neural Network (CNN) architecture [2].

1.3.1.1 Convolution Layers

In CNNs (Convolutional Neural Networks), convolutional layers are the foundation for identifying spatial patterns in data, especially images. These layers apply small filters called kernels (typically of size 3×3, 5×5, or 7×7) that slide over the input with a defined stride (commonly 1 or 2) and may use padding (like "same" or "valid") to control output size. Each kernel is trained to look for specific features such as edges, textures, or shapes in an image. As a convoluted layer becomes deeper in the network, more abstract and complex features are identified by using the output of previous convoluted layers in the identification process. In the end, we end up with a set of feature maps, which indicate how often we have captured one of the unique patterns learned in the input image.

$$Y_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{i+m,j+n} . K_{m,n} + b \tag{1.1}$$

Where:

- Y: the output feature map.
- X: input feature map.
- K: kernel weight.

1.3.1.2 Pooling Layers

Pooling layers are responsible for reducing the height and width of feature maps while keeping the most important information. When feature maps keep feature information and discard the rest it leads to a model that is more efficient, reduces overfitting chances, and is easier to train. The most common type of pooling uses max pooling, typically using a 2×2 window with a stride of two pixels, and recording the maximum value of that pool region. The other type is average pooling, where the average of the pooling region is taken. Pooling helps produce translation invariance, in that when the input is shifted some pixels the output is relatively unchanged. This enables the network to learn high-level features from spatially-correlated pixels, rather than literal pixel locations.

1.3.1.3 Fully Connected Layers

After the convolution and pooling stages, CNNs use one or more Fully Connected (FC) layers to perform high-level reasoning. In these layers, each neuron is connected to every neuron in the previous layer. Generally speaking, FC layers convert 2-dimensional feature maps obtained from the convolutional and pooling stages into a one-dimensional vector generated from all of features maps corresponding neurons, which can then be used for classification and/or regression. The fully connected neurons may include their own activation functions (e.g. ReLU or Softmax), and fed to, after the FC layer, dropout layers are commonly placed in order to prevent overfitting. For example, in the case of image classification tasks, the length of the output vector from the FC layer is usually equal to the number of classes with each value representing the probability of the corresponding class.

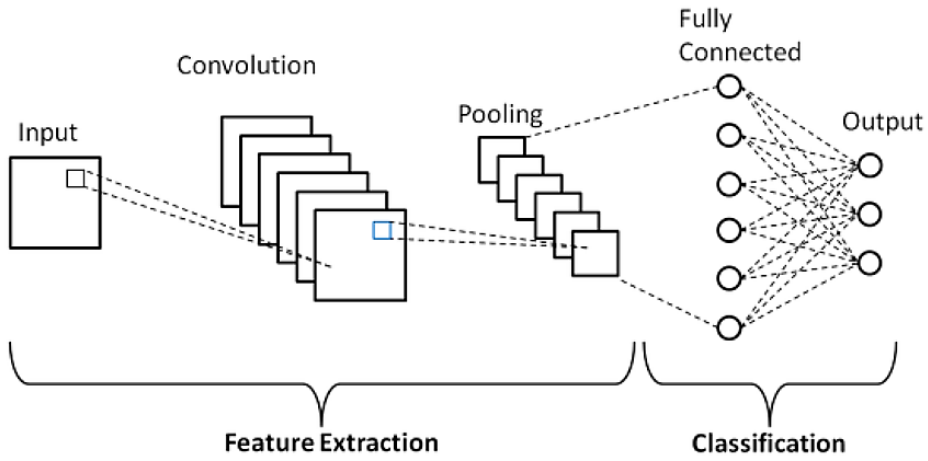


Figure 1.3: Convolution Neural Network Architecture [16]

1.3.2 The Vision Transformer (ViT)

The Vision Transformer [17], or ViT, introduces a new transformer-based framework into the ongoing advancements in neural networks for computer vision tasks. Taking inspiration from the application and success of Transformer architectures in natural language processing [13], the ViT employs an architecture literally based upon Transformers for the purpose of working with 2D image data. In order to effectively work with image features, the ViT reshapes the input image into a sequence of flattened 2D patches. Each patch is then projected into D-dimensional embedding space via a learnable linear projection forming the patch embeddings. To this sequence of patch embeddings, a learnable class token is added. This token state at the output of the Transformer is the representation for the image. In order to add positional context, learnable 1D position embeddings are added to the patch embeddings. The sequence of embeddings is then passed through a transformer encoder which contains multiple layers of multi-head self-attention and feed-forward networks using the architecture shown in Figure 1.4.

Self-attention allows the model to weigh the importance of different patches, enabling it to focus on relevant parts of the image simultaneously. This mechanism captures long-range dependencies and global context [18], crucial for effective image recognition. The encoder's output is then passed through a classification head for final predictions. This approach leverages the transformer's strength in handling sequential data and modeling complex relationships within

the image. The ViT also demonstrated great scalability and performance gains when pre-trained on large datasets. Overall, the ViT is an effective alternative to conventional convolutional neural networks in numerous computer vision tasks.

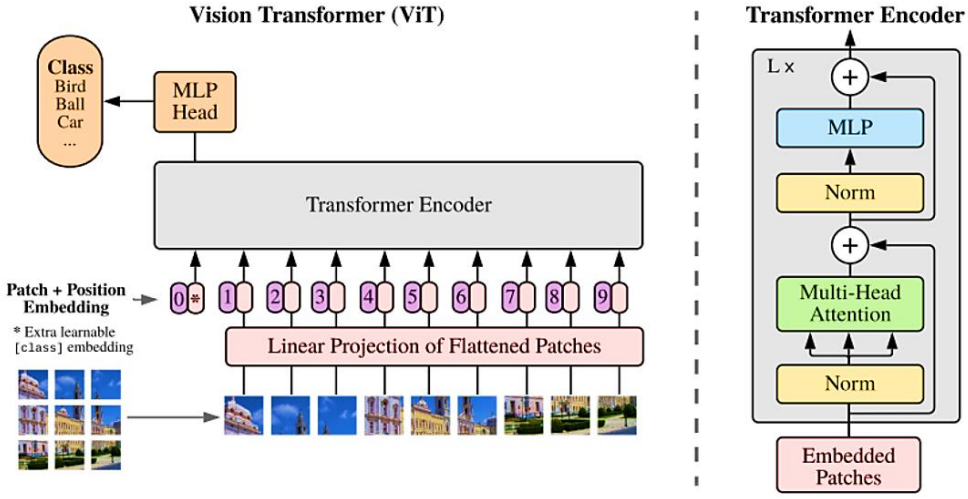


Figure 1.4: Architecture of the Vision Transformer (ViT) [17]

1.3.3 Loss Functions

The loss functions, is used to measure the distance between the model's predicted values and the actual target values (ground truth). The loss function that you use may depend largely on the task you are performing (i.e. regression or classification). Below you can find some common loss functions to use [15].

1.3.3.1 Mean Squared Error (MSE) Loss

MSE loss is commonly used in regression problems because it computes the average of the squares of the differences between the predicted values and true values. In MSE loss, the squaring step means that larger differences count even more (are emphasized) meaning that MSE loss is sensitive to outliers [19].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.2}$$

1.3.3.2 Cross-Entropy (CE) Loss

Cross-entropy loss is commonly used for classification-based tasks. It calculates the distance between the predicted probability distribution and the actual label distribution. For binary classification it is expressed as:

$$CE = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{1.3}$$

For multi-class classification, the loss extends to:

$$CE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{c} y_{ic}\log(\widehat{y_{ic}}) \tag{1.4}$$

1.3.4 Regularization

One of the greatest difficulties in training deep neural networks, such as CNNs, is to check their generalization ability on data that it has not seen before. Generalization is a model's ability to accurately predict for new inputs from the same distribution as the training data. A common mistake is overfitting, where the model achieves very high accuracy on training data, but is unable to maintain accuracy on the test data because it learned noise, or other spurious patterns in the data. [15] To mitigate overfitting, and improve generalization, there are a number of common regularization techniques that can be used:

1.3.4.1 Dropout

Dropout is a form of regularization, where certain neurons (or neurons associated connections) are randomly deactivated during training. This method reduces dependence on certain neurons and forces the model to learn more generalized features during training [20].

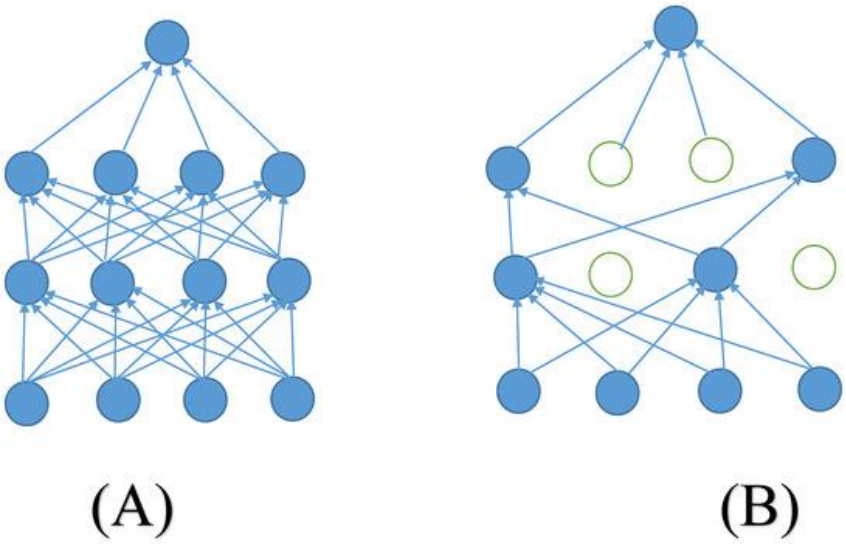


Figure 1.5: An illustration of a standard neural network (A) and after applying dropout (B) [21].

1.3.4.2 Batch Normalization (BN)

Batch Normalization normalizes the input of each layer to have zero mean and unit variance to help stabilize the training process. BN helps with stabilizing the training process by addressing issues, such as vanishing gradients and exploding gradients [20]. It significantly accelerated convergence and increases performance. BN reduced sensitivity to weight initialization and reduced neurons becoming inactive in ReLU layers.

1.3.4.3 Data Augmentation

Data Augmentation is a different regularization technique in which new training samples are generated from the initial input images using random transformations like rotations, translations, flipping, zooming, cropping, brightness changes, and noise [22]. Data augmentation increases the

size of the dataset and allows the model to encounter more varieties of input, which helps minimize overfitting and motivates the model to learn invariant features. It is an implicit form of regularization because the model is trained on images in which it cannot memorize the training data, but can process the input more generically.

1.4 Computer Vision

Computer vision [23] is a sub-field of artificial intelligence that focuses on acquiring, processing and analyzing images and videos. Image acquisition includes the process of acquiring two-dimensional (or three-dimensional) images from any device capable of capturing images. This image may come from a camera, sensor, or other devices. Image processing will enhance image quality and remove noise and other artifacts and allows the processing of the acquired images so that the extracted features are suitable for analysis based on characteristics such as color, texture, and shape. Image analysis is ultimately the last step, which takes place after image processing and involves extracting meaningful information through techniques such as image classification, object detection, segmentation, or recognition.

1.4.1 Image Classification

Image classification is a central task in computer vision which involves the association of one or several predefined labels given to an input image based on its visual content. This task consists in the features extraction and analysis of the image that can be, for instance, texture, color, and edges shape; the conclusion is then drawn on the most proper category by using these features. The major purpose is to make the systems be able to carry out the visual data recognition and categorization by themselves and thus achieve a high accuracy level. Image classification underlies the success of a great number of real-life applications in different fields, e.g., facial recognition, medical image diagnosis, autonomous driving, and surveillance systems. Furthermore, image classification acts as an initial stage for high-level tasks of object detection, segmentation, and activity recognition. In the last 10 years, the field has been advancing thanks to the introduction and wide usage of Convolutional Neural Networks (CNNs).

CNNs have dramatically transformed the area by facilitating the models to learn representations of features in increasing order of hierarchy from the raw image pixels automatically and consequently, eliminating the need for manual feature engineering and thus improving the classification accuracy for large image datasets. These deep learning [24] based approaches have become a touchstone category in visual recognition assignments and are still the main impetus for AI systems that are aimed at interpreting visual objects and scenes.

1.4.2 Image Segmentation

Image segmentation [25] is a core component of computer vision and digital image processing that involves dividing an image into meaningful and coherent regions. The primary goal is to isolate regions of interest (ROIs) based on pixel similarities in color, texture, or intensity, which enhances object detection, scene understanding, and analysis efficiency in various visual tasks.

There are three main types of image segmentation:

Semantic Segmentation: Groups all pixels belonging to the same class into one category, but does not distinguish between separate instances (e.g., all cars in an image are labeled “car”) [26].

Instance Segmentation: Extends semantic segmentation by identifying different instances of the same class, making it suitable for applications requiring detailed object-level analysis, such as autonomous vehicles and robotics [27].

Panoptic Segmentation: Combines the above two, assigning each pixel both a semantic label and an instance ID, thereby providing a holistic understanding of scenes [26].

Applications of image segmentation are vast and include:

- **Medical Imaging**: For delineating anatomical structures or tumors, aiding diagnosis, treatment planning, and disease tracking.
- **Robotics:** For real-time object recognition and navigation.
- **Autonomous Vehicles**: For detecting lanes, pedestrians, and obstacles.
- **Agriculture**: For distinguishing between crops and weeds to support precision farming.
- **Smart Cities**: For surveillance, traffic control, and urban safety through real-time analysis from CCTV cameras.

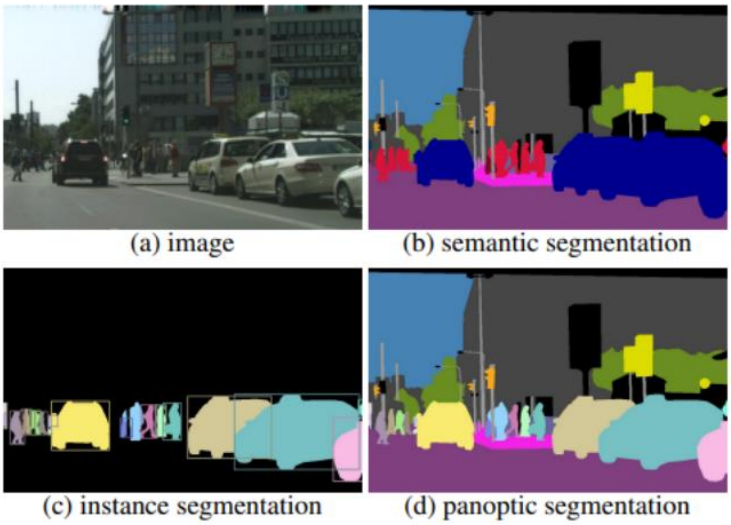


Figure 1.6: Comparison of image segmentation techniques [28]

1.4.3 Object Detection

Object detection is one of the most critical tasks in computer vision. It consists of classifying the semantic objects in an image and localizing their instances. In contrast, image classification only provides a single label for the image as a whole; object detection returns the category and a location (typically bounding boxes) for every object in the image or video. This is a unique case in that we can not only classify the content in a certain scene, but we can also localize where it appears! This dual-tasking capability is fed into many applications including video surveillance, autonomous driving and robotics, augmented reality experiences, and so on.

Modern object detection approaches are typically categorized into two main families:

Two-stage detectors, such as R-CNN and its successors (Fast R-CNN, Faster R-CNN), which first generate region proposals and then classify each region. These models tend to offer high accuracy and are well-suited for tasks where precision is prioritized [29].

One-stage detectors, such as YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector), which perform detection in a single forward pass, making them faster and more suitable for real-time applications [30].

These models rely primarily on deep convolutional neural networks (CNNs) to learn feature representations and to predict object classes and bounding box coordinates. Recently, transformer-based architectures have emerged (e.g., DETR: Detection Transformer) that model object detection as a set prediction problem, which simplifies the pipeline while producing better performance in complex scene.

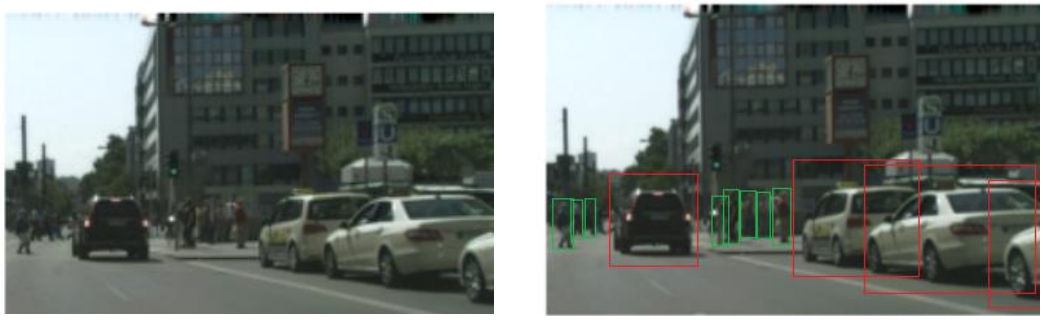


Figure 1.7: Object Detection Example

1.5 Medical Imaging

Medical imaging [31] is a method used to acquire physiological and functional data on bodily parts or organs from their images to detect and monitor diseases and conditions. The image showcases the application of intended imaging techniques and employs particular image processing methods Figure 1.8 that preserve the morphological details and features of the image.

Medical imaging techniques vary by the particular approach used to acquire a scientific image, including positron emission tomography (PET), single photon computed tomography (SPECT), X-ray computed tomography (X-ray CT), ultrasound, photoacoustic imaging, spiral computed tomography (spiral CT), magnetic resonance imaging (MRI), functional near-infrared spectroscopy (FNIR), and numerous others.

These techniques are known for their high effectiveness as a radiological tool in diagnostic and medication assessment. the recent evolution of deep learning (DL) and Machine learning (ML) fields have led to a high development of multi analytical techniques that discover and detect the anomalies by using more than just image processing only and sound waves too. Several artificial intelligence (AI) techniques, deep learning techniques specifically has contributed to the advancement of medical imaging methods which include recognition, enhancement, segmentation, visualizations and feature extraction [31].

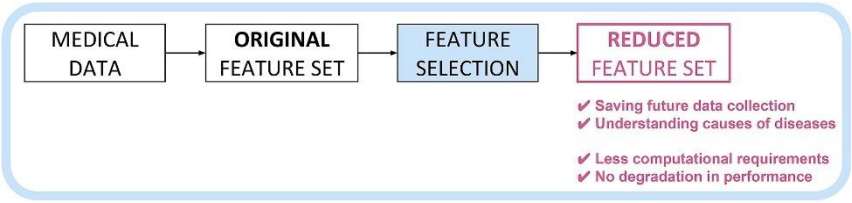


Figure 1.8: General steps of the feature selection (preprocessing technique) approaches and their main benefits [32].

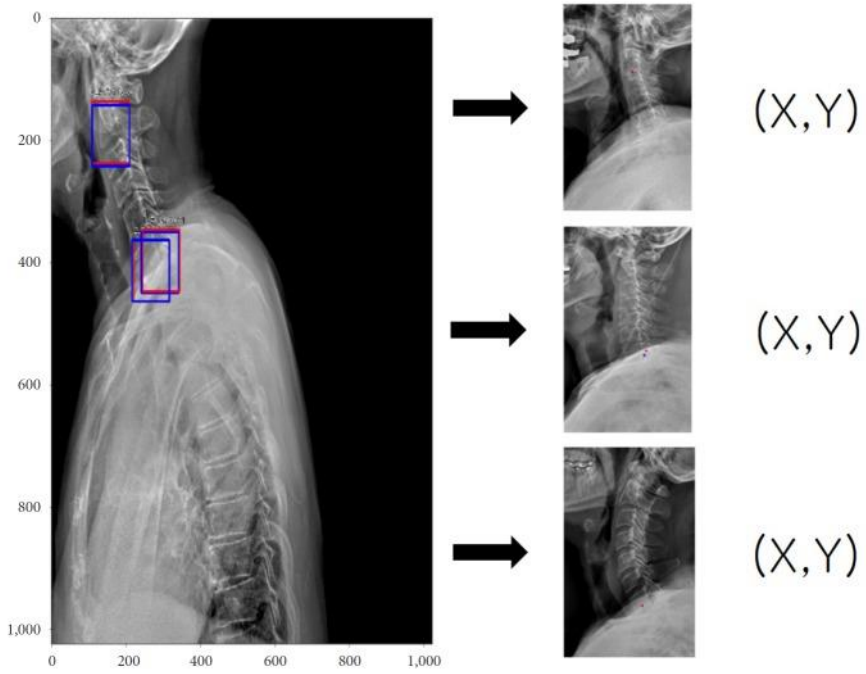


Figure 1.9: A typical example for detecting of region of interest (ROI) images (left; red, gold standard; blue, prediction ROI) and their coordinates (x and y) of corresponding landmarks in spine sagittal X-ray [33].

1.5.1 Overview of different medical imaging modalities

1.5.1.1 X-ray imaging

Is using electromagnetic radiation known as X-rays, which can get through many various materials even biological tissues (human body), to create an image that reveals internal structures such as bones [34].

1.5.1.2 Computed Tomography (CT) imaging

CT uses X-ray to create detailed cross-section images. which would be used for diagnosing and supervising a wide list of conditions, including heart disease, internal organs injuries and cancer [35].

1.5.1.3 Magnetic Resonance Imaging (MRI)

MRI scans create a very detailed images of internal anatomical structures such as biological tissues, the brain and muscles, with using a strong magnetic field and radio waves [36].

1.5.1.4 Ultrasound imaging

Ultrasound utilizes high-frequency acoustic waves to generate images of internal structures, including the uterus and ovaries during pregnancy. It is utilized to assess the organs and blood vessels within the abdomen and to direct needle biopsies [37].

1.5.1.5 Nuclear medicine imaging

Nuclear medicine uses small amounts of radioactive material to create images of internal structures and to identify abnormalities, such as tumors or infections. This modality is also used to diagnose and monitor certain diseases such as cancer, heart diseases, and thyroid diseases [38].

1.5.1.6 Electrical Impedance Tomography (EIT)

EIT imaging images internal structures, including the heart and lungs, using electrical impulses and tracks changes in their electrical characteristics [39].

Modality	Working Principle	Applications	Advantages	Limitations
X-ray	Using ionizing radiation to produce images of the internal structure of a body	Detecting broken bones, monitoring treatment of conditions such as pneumonia, monitoring the healing of fractures	Inexpensive, widely available, quick results	Low-resolution images, ionizing radiation exposure
CT scan	X-ray technology combined with computer processing to produce detailed images	Detecting cancers, identifying blood clots, assessing organ damage, diagnosing spinal problems	High-resolution images, non-invasive	Ionizing radiation exposure, high cost
MRI	Combination of powerful magnetic fields and radio waves allows for the creation of high-resolution photographs of hidden structures	Detecting tumors, brain and spinal cord injuries, joint problems, and monitoring the progression of conditions such as multiple sclerosis	Non-ionizing radiation, detailed images	Long examination time, high cost, not suitable for patients with metal implants
Ultrasound Imaging	Using high-frequency sound waves to produce images	Monitoring the growth and development of a fetus, evaluating organs and tissues, detecting tumors and cysts	Non-invasive, no ionizing radiation exposure	Operator dependent, limited view of deep structures
Nuclear Imaging	Using radioactive tracers to produce images	Detecting diseases and conditions such as cancer, heart disease, and neurological conditions	High specificity for certain conditions, non-invasive	Limited view of the structure, exposure to ionizing radiation
Electrical Impedance Tomography	Using electrical currents to produce images	Monitoring changes in tissue, measuring organ function	Non-invasive, portable	Limited spatial resolution, operator dependent

Table 1.1: Comparative Analysis of the medical imaging modalities [40].

1.5.2 Brain tumors

A tumor (also called a neoplasm) is abnormal tissue that grows by uncontrolled cell division. Normal cells grow in a controlled manner as new cells replace old or damaged ones. for reasons not fully understood, tumor cells reproduce uncontrollably.

Brain tumor is named after the cell type from which they grow. They may be primary which their grow starts in the brain or metastatic (secondary) starts as a cancer outside the brain area then begin to spread into the brain from the blood streams [41].

Brain tumors are pretty common worldwide, and that emphasizes how important it is to have really accurate ways to diagnose them. Since the symptoms of brain tumors often overlap with other neurological issues, we need a diagnostic tool that can catch even the tiniest or earliest signs with high accuracy. The usual methods we use are helpful, but they can sometimes be invasive or miss small tumors early on [42].

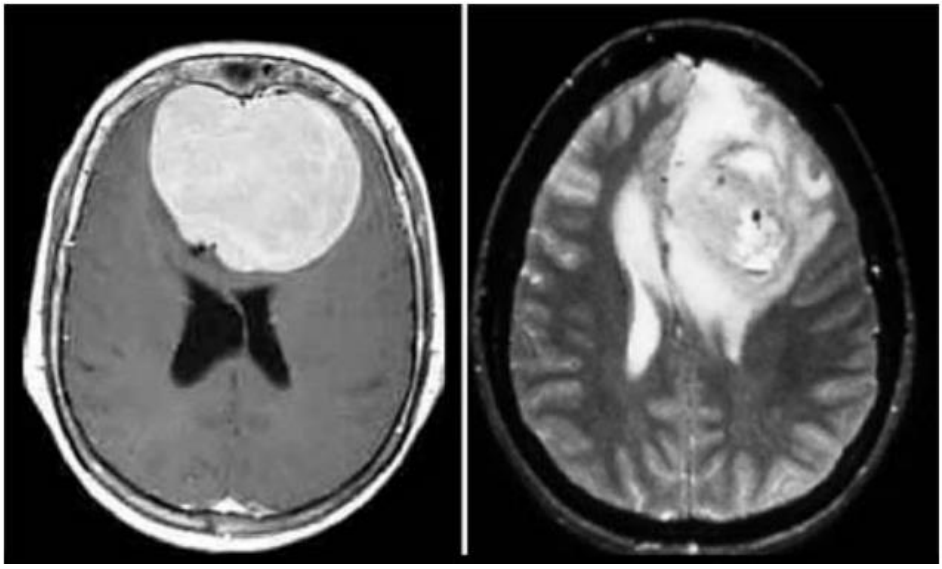


Figure 1.10: MRI scans for a primary brain tumor type (benign and malignant), the 1st type has well defined edges and are more easily removed surgically. Malignant tumors have an irregular border that invades normal tissue making surgical removal more difficult [41].

1.5.3 Medical Diagnosis

Medical Diagnosis is the process of forecasting the patient's disease in early stages such as tumor , by observe the symptoms .This process involve collect the required evidence (X-ray imaging , MRI ,etc.) and exanimations .This called diagnosis path .In order to get to the right diagnosis [43], but sometimes the diagnosing face some serious challenges to achieve the correct diagnosis due the lack of careful observation and the overlap of the diseases , the variability in medical expertise which make this process high risky and difficult .

With the evaluation wave of artificial intelligence specifically deep learning models such as Convolution Neural Network (CNN) and Recurrent Neural Networks (RNNs), which can analyze high complex medical data. The contribution of deep learning has helped in enhancing accuracy and efficiency of identifying the diseases [44].

1.6 Deep learning model compression

1.6.1 What are the Lightweight Models

Deep learning frameworks are categorized as lightweight [45] when they are capable of completing sophisticated tasks while consuming significantly low computational and energy resources. This classification typically applies to models that can operate within constraints such as under 100MB of memory, less than 1 billion FLOPs (floating point operations), or inference times under 100 milliseconds on edge devices. These devices include mobile phones, embedded systems, and more recently, IoT devices, all of which generally have limited processing power and strict energy constraints [46]. To meet these lightweight requirements, it is essential to reduce the number of model operations intelligently. Techniques such as model pruning, quantization, the use of low-power architectures like MobileNet and ShuffleNet, and the adoption of faster hardware all contribute to making deep learning feasible on resource-constrained platforms.

1.6.2 Evaluation Metrics for Lightweight Deep learning

In Deep Learning (DL), the three most commonly used metrics for evaluating model compression and computational efficiency are Floating Point Operations (FLOPs), Multiply-Accumulate Operations (MACs), and Memory Access Cost. These metrics serve as indicators of the complexity and feasibility of deploying models in resource-constrained environments [47]. FLOPs refer to the total number of floating-point arithmetic operations such as addition, subtraction, multiplication, and division that the model performs during inference [47]**.** This metric helps assess the computational workload and is commonly used when comparing architectures.

For a convolutional layer, the total number of FLOPs can be calculated as:

$$FLOP_S = 2.H.W.C_{in}.C_{out}.K.K \tag{1.5}$$

where:

- H and W are the height and width of the output feature map
- C_{in} and C_{out} are the number of input and output channels, respectively
- K is the kernel size,
- and the factor of 2 accounts for both the multiplication and addition operations in each MAC [48].

MACs (Multiply-Accumulate operations) represent the number of operations involving one multiplication followed by one addition, typically used in layers such as convolution and fully connected layers. While similar in scale to FLOPs, MACs only count these specific pairs of operations. Typically, FLOPs are approximately twice the number of MACs, due to the separate counting of multiplications and additions in FLOPs [48].

For a convolutional layer, MACs are estimated by:

$$MAC_S = H.W.C_{in}.C_{out}.K.K \tag{1.6}$$

Memory Access Cost (also abbreviated as MAC) is distinct from Multiply-Accumulate operations. It refers to the number of memory accesses required to read inputs, write outputs, and retrieve weights. This cost is significant in edge and embedded systems, where memory bandwidth and energy are constrained.

An approximation of the Memory Access Cost is:

$$\text{Memory Access Cost } (MAC) = H \cdot W\ (Cin + Cout) + k \cdot k\ (Cin \times Cout) \qquad (1.7)$$

This equation estimates the memory required to store and access feature maps and kernel weights during convolution.

In addition to computational and memory costs, two other critical performance metrics are throughput and latency. Throughput refers to the number of inferences the model can complete per second, representing processing capacity. Latency measures the time taken to produce an output from the moment an input is received, i.e., the time per inference. The inverse relationship between throughput and latency is well established and is formally discussed in [49]**.**

1.6.3 Compression Techniques

1.6.3.1 Depthwise separable convolution

It's an effective variation of standard convolution operation designed to reduce the computational cost and number of parameters in convolutional neural networks (CNNs). It was introduced by Chollet in the Xception architecture [50].

In the usual 2D convolutional, the operation jointly applies filters across both the spatial dimensions height and width and the input channels. In contrast, a depthwise separable convolution splits this process into two steps:

Depthwise convolution: aims to captures the spatial features within each channel, by applied a single filter to each input channel independently.

Pointwise convolution: $A\ 1 \times 1$ convolution is then applied to combine information across channels, learning relationships between them.

The split reduces computing resources by a lot and also lessens the amount of memory that a model may require. As a whole, the computation costs are about:

$$\text{Cost}_{\text{depthwise separable}} \approx \frac{1}{N} \cdot \text{Cost}_{\text{standard convolution}} \qquad (1.8)$$

Where N is the number of the output channels (filters)

This makes depthwise separable convolution particularly effective for mobile and embedded applications, where computational efficiency is critical [50] [51].

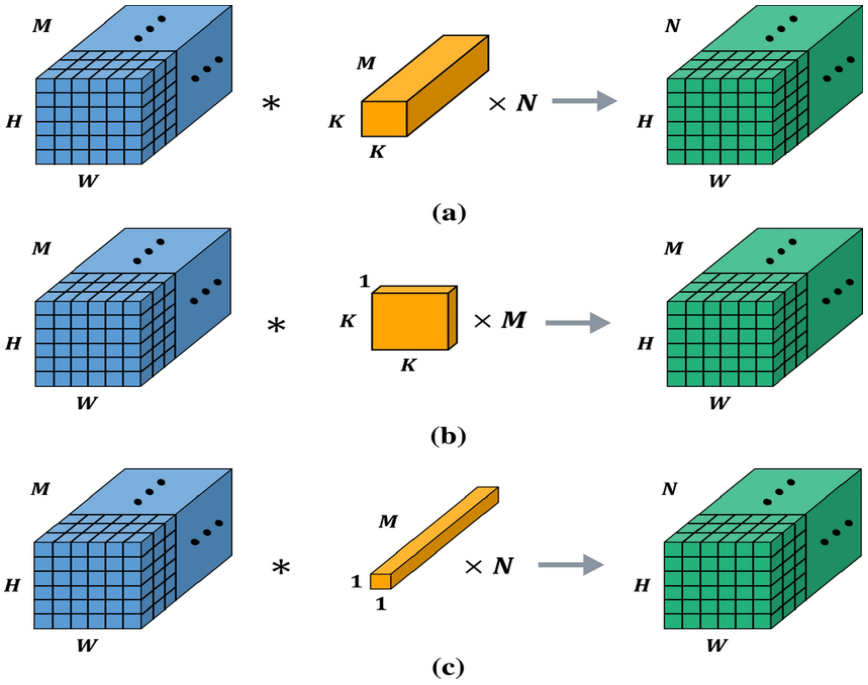


Figure 1.11: Process of convolutions. (a) Standard convolution; (b) depthwise convolution; (c) pointwise convolution

1.6.3.2 Group convolution

Group convolution is a variant of standard convolution that aims to reduces computational complexity and the number of convolutional neural networks as the depthwise convolution method but instead of applying the filter across all input channels, it divides the input and the output channels into separate groups. Each group of inputs is convolved independently with its own set of filters, the output for mall the groups are concatenated to form the final output feature map.

This structure lowers the number of the operations because it avoid the full connection between input and output channels presents in the usual convolution .the first introduce was in AlexNet to overcome hardware memory limitations [52],since then GC become a core component of modern efficient architectures such as ResNeXt [53].

1.6.3.3 Pruning:

Pruning is a compression technique that reduces the size of deep neural networks by removing unnecessary parameters or structures, which helps improving the model efficiency without effecting the model performance because it makes the model suitable to run on limited resource devices.

Puring methods are classified into two types weight pruning and filter pruning.

Weight pruning: is the process of sparse a set of connection weights at each layer or just setting their value to zero, the eliminated weights are usually the those with small magnitudes. this process reduces the footprint and accelerates the model training because the result of this process is a sparse network that have the same original structure with less weights, only the effective ones.

This method was explored in many studies m such as the pioneering work by Han et al [54] ,which introduced iterative pruning the high compression rates without losing the efficiency of the model especially the fully connected layers in large networks [55] .

Filter pruning: Rather than the individual weights, the FP prunes at the high structural level such as channel, filter and layers. removes an entire filter which leads to a smaller and thinner model with fewer feature maps and channels. The usual filter-pruning pipeline comprises three steps:

- Baseline training: which aims to train a large teacher model on a specific dataset.
- Filter selection or removal: is removing the filters that have the low rank according to a chosen importance or a selected criterion m such as L1-norm, learned importance scores.
- Fine-tuning: aims to recover the lost accuracy by retraining the pruned network.

Filter pruning stands with several strategies such as receptive filed criterion (RFC), learned filter importance, stripe-wise pruning and layer-specific pruning based on RFC [56] [27] .

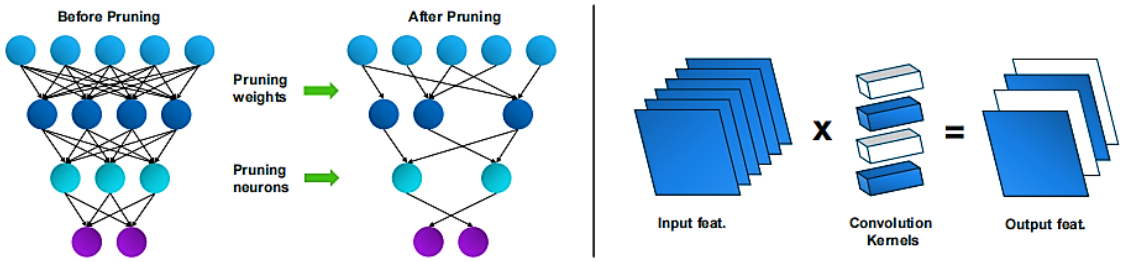


Figure 1.12: Pruning techniques are shown as organized pruning (Filters Pruning) and unstructured pruning (Weights Pruning). The components that have been pruned are displayed in white. Observe how the output dimensions of the trimmed component have changed. [57]

1.6.3.4 Quantization:

A wide used model compression technique the helps reduces the numerical precision weights and activations in a neural network which leads to enhancing efficiency of the memory and a high speed. the main idea is replacing high precision 32-bits (floating point FP-32) with lower precision, 8-bit integers (INT8). this allows the deep learning models to consumes less resources so more efficiency on limited hardware. with minimizing performance degradation [58].

Applying INT8 quantization reduces the model's memory footprint significantly speed up the inference due the faster integer arithmetic in the processors [59], the wreak handing with this strategy can lead to a losing in accuracy and decreasing the model performance which opened the door for more robust strategies:

Quantization-Aware Knowledge (QAT):

In this method both weights and activations are subjected to simulated quantization operations, by using fake quantization modules that imitate lower-bit precision arithmetic while keep the high precision for gradient updates. This allows the model to learn to compensate for quantization-

induced errors, resulting in higher accuracy when the model is later deployed with actual low-precision inference [58] [60].

Post-training quantizing (PTQ):

This method involves quantize a fully trained model (pre-trainer) and reducing the precision of weights and activations from FP32 to INT8 only, no other changes in the original training process [58]. Because the original model isn't trained to account for quantization noise. this may affect the PTQ to degradation in the model accuracy.

1.6.3.5 Weight clustering:

A technique aims to reduce the number of unique weight values in the network structure, it works by grouping the similar weight values into clusters and give each group a shared representative value which referring to the centroid, the reducing of the memory footprint comes from storing the centroids and cluster assignments [61].

By reducing the number of distinct parameters, weight clustering is more efficient and faster computation when the model's accuracy is preserved and the method has been applied appropriately.

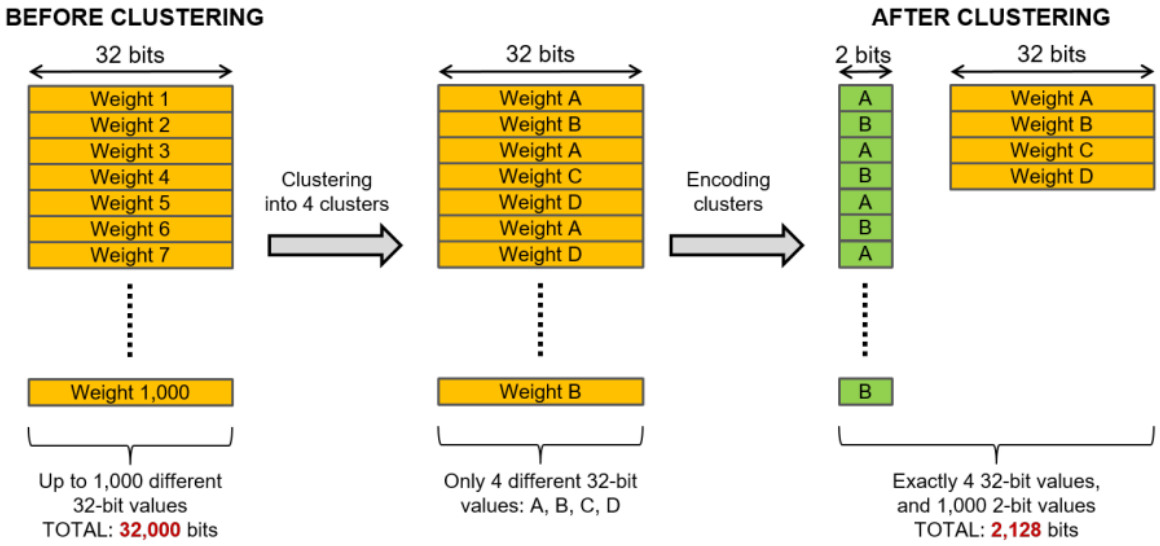


Figure 1.13: Example where 1,000 32-bit weights are clustered into 1,000 2-bit weights [62].

1.6.3.6 Knowledge Distillation (KD)

Knowledge Distillation is a wide adopted model compression technique that facilitates the transfer of knowledge from a large a model called teacher to a smaller lightweight model called student [63] [1] figure 1.14 illustrates the concept. This allows the student model to reach the teacher's performance while requiring a very fewer parameters and computational resources. In the approach originally proposed by Hinton et al [1] , teacher is trained first on the full dataset, and then used to produce soft targets. The student is subsequently trained on combined of the true labels and teacher's soft predictions. This dual-supervision enables the student to learn not only to correct class labels but also the subtle class similarities encoded in the teacher's output. The result is the ability of the student to achieve an accuracy extremely close to the teacher. With having a huge reducing in the number of total parameters.

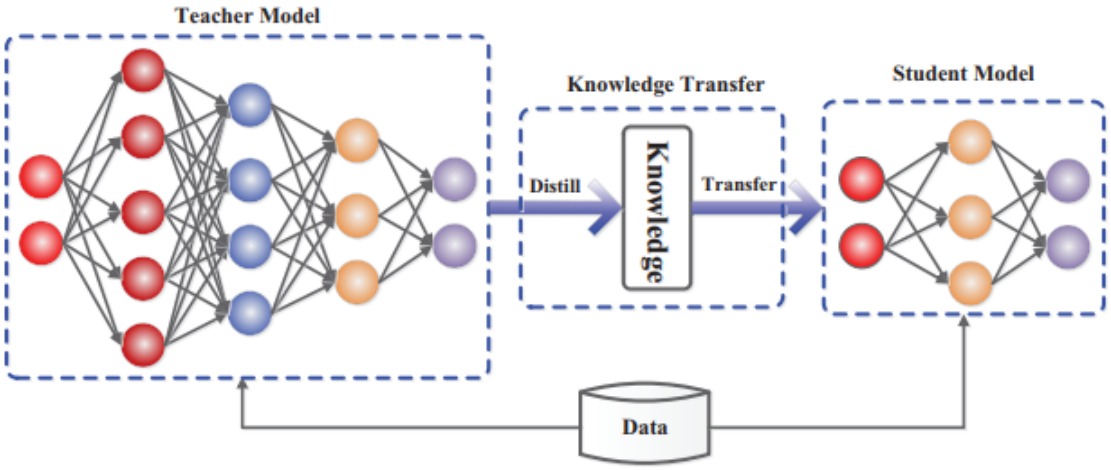


Figure 1.14: The generic teacher-student framework for knowledge distillation [64].

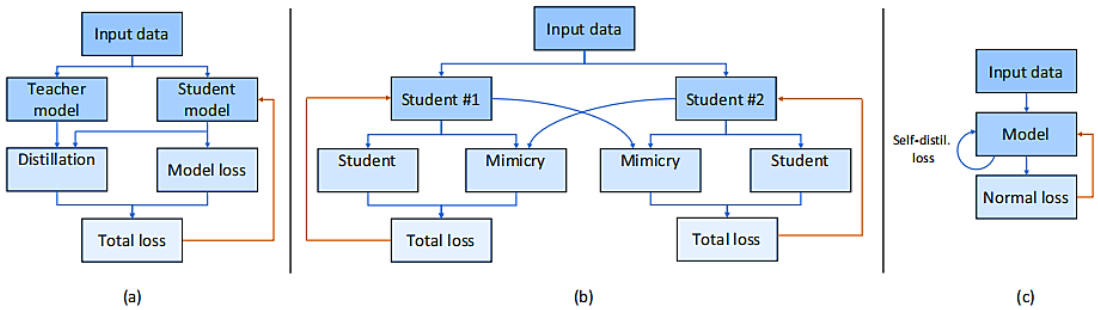


Figure 1.15: (a) Offline Distillation [1]. (b) Online Distillation [67]. (c) Self-Distillation [68] . We use orange lines to indicate the gradient update.

The knowledge distillation strategies can be categorized into three types as illustrated in figure 1.15.

- **Offline distillation:** the teacher is a pre-trained and fixed model (high-performing model) and student is trained sequentially after the teacher [63]. figure 1.15(a)
- **Online distillation:** both the teacher and the student are trained simultaneously, where the teacher can be either a fixed model or an evolving ensemble. The student continues learning while the teacher is still active during training [65]. figure 1.15(b), the teacher and the student both are referred as student 1,2.
- **Self-distillation:** the student model learns by itself either by the earlier epochs or from deeper layers acting as a teacher. With this paradigm the student can avoid use a separate teacher model [66].figure 1.15(c).

Each categorize from these defines a role of the teacher and the supervision process differently, but with same goal which is mimics the performance of a larger, more resource-intensive teacher.

1.6.4 State-of-the-Art in Knowledge Distillation:

Knowledge Distillation has become a prominent method in deep learning for model compression and knowledge transfer, KD enables the smaller student model to learn form a large high-preforming model by mimicking its output, distribution or internal representation. This allows the student to be extremely efficient with a few numbers of parameters.

The recent development of KD research has led to a various of methods, which can be organized based on the type of knowledge transferrin, the distillation paradigms, and the target tasks.

Type of knowledge transferrin:

Respond-based KD (Logits): in the classic approach where the student learns from the teacher's soft output probabilities. This was the original formulation by Hinton et al [1].

Feature-based KD: knowledge is transferred from intermediate layers such as activations, attention heatmaps (attention transfer) [69] and hits (FitNets) [70] .

Relation-based KD: Rather than individual features, the relationships between samples in feature space are distilled such as preserving the pairwise and triple-wise between embeddings (Relational KD) [71] , or using teacher supervised contrastive loss to transfer class separation knowledge (Contrastive KD) [72].

Structural or Graph-based KD: Transfers structural representations of data using graphs or attention-based modeling [73].

In the following, we summarize the state-of-the-art methods concerned with KD in the context of medical imaging and brain tumors detection (see table 1.2):

Study	Teacher Model	Student Model	Dataset	KD Method	Performance (Dice/Accuracy)	REF
Chen et al. (2021)	ResNet-50	MobileNetV2	BraTS 2019	Logits + Feature Map	Dice: 0.87 (T) / 0.83 (S)	[74]
Tang et al. (2022)	Deep U-Net	Shallow U-Net	BraTS 2020	Response + Intermediate Features	Dice: 0.88 (T) / 0.86 (S)	[75]
Wang et al. (2020)	DenseNet	Custom CNN	Private MRI Dataset	Response KD	Accuracy: 94.2% (T) / 92.5% (S)	[76]
Iqbal et al. (2023)	VGG16	Tiny CNN	Kaggle Brain Tumor Dataset	Logit + Feature KD	Acc: 98.5% / 96.1%	[77]
Elazab et al. (2022)	ResNet-50	MobileNet	Kaggle Brain Tumor Dataset	KD + Pruning	Acc: 97.9% / 95.3%	[78]

Ahmad et al. (2024)	Multi-Teacher Ensemble (U-Net variants)	Unimodal U-Net	BraTS 2018–2021	Cross-Modal Distillation	Improved Dice over baseline	[79]
Dong et al., CVPR 2023	Ensemble of Local Models	Lightweight Student Model	Figshare Brain Tumor Dataset	Ensemble KD in Federated Learning	High accuracy with reduced communication cost	[80]
Qin et al. (2021)	Deep Segmentation Network	Lightweight Network	LiTS17, KiTS19	Semantic Region Distillation	Up to 32.6% improvement in segmentation	[81]
Dong et al. (2023)	ResNet50 / ResNet34	DisWOT-searched ResNet18	ImageNet-1k	DisWOT + KD variants	72.30% (vs. 69.75% vanilla), Top-1 ↑ 2.55%	[3]
Dong et al. (2023)	ResNet56	DisWOT-searched student	CIFAR-100	CRD, KD, AT, FitNet, etc.	75.25% (CRD, 1M params)	[3]
Dong et al. (2023)	ResNet110 / ResNet56	DisWOT-searched students	NAS-Bench-201 (CIFAR-10/100/ImageNet-16-120)	DisWOT + KD	74.21% (CIFAR-100), 180× speedup	[3]

Table 1.2: SOTA of Knowledge Distillation in Brain Tumor Datasets

1.7 Conclusion

To summarize this chapter, a detailed discussion has been presented about the fundamental elements that make modern AI application possible (machine learning, deep learning, computer vision, and image classification). It has highlighted the increasing reliance on Convolutional Neural Networks to process visual data, and the growing concern for lightweight architectures for use when there are resource constraints. The aspects of learning paradigms and model compression methods - like pruning, quantization, and knowledge distillation - have also emphasized how the search for optimal accuracy and efficiency remains an active area of research. This underlying knowledge will help inform the motivation and methods for lightweight deep learning models that are examined in the following chapters of this book.

Chapter 2

Proposed Method

2.1 Introduction

The increasing demand for efficient deep learning models in real-world applications has driven the development of lightweight neural networks through knowledge distillation. While traditional KD transfers knowledge via soft and hard labels, recent training-free approaches like DisWOT (Distillation without training) aim to identify optimal student models from a defined students search space without training for distillation, using Grad-CAM and L2 similarity for semantic and relational guidance. However, these methods often suffer from coarse localization and sensitivity to activation scale, limiting their effectiveness.

To address these challenges, we propose an enhanced lightweighting strategy by replacing Grad-CAM with Layer-wise Relevance Propagation (LRP) for fine-grained semantic supervision and substituting L2 loss with cosine similarity for robust, scale-invariant alignment. This refined framework improves the selection and training of compact student models, offering better generalization and training stability in knowledge distillation.

2.2 Knowledge Distillation (KD)

Knowledge Distillation (KD) is a model compression technique proposed by Hinton et al. (2015) [80], where a small student model is trained to imitate the behavior of a larger and more accurate teacher model. The objective is to enable the student model to achieve comparable performance while being computationally efficient.

During training, the student learns from two types of supervision:

Hard labels, which represent the ground truth class labels. The loss used here is the Cross-Entropy (CE) loss, defined as:

$$L_{CE} = -\sum_{i} y_i \log \sigma(z_{si}) \tag{2.1}$$

where z_{si} are the student's logits (output scores produced by the model before the softmax layer), and $\sigma(.)$ denotes the softmax function.

Soft labels, which come from the teacher model's softened output. These are obtained by applying a softmax function with a temperature $T > 1$, which produces smoother probability distributions:

$$\sigma(z_i, T) = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j exp\left(\frac{z_j}{T}\right)} \tag{2.2}$$

Here, T is the temperature parameter, and it plays a critical role in controlling the softness of the output distribution. A higher temperature (e.g., $T > 1$) spreads the probability mass more evenly across classes, highlighting the similarities between them and making the output distribution less confident. This helps the student model learn nuanced inter-class relationships that are not captured by hard labels alone.

Soft labels provide extra information about the teacher's confidence across all classes, helping the student learn inter-class relationships, not just the correct answer. This richer supervision is captured using the Kullback-Leibler (KL) divergence, which measures the distance between the teacher's and student's softened outputs:

$$L_{KL} = T^2 \sum_i \sigma(Z_{ti}, T) log(\frac{\sigma(Z_{ti}, T)}{\sigma(Z_{si}, T)}) \tag{2.3}$$

Finally, the total KD loss is a weighted combination of both losses:

$$L_{KD} = \alpha . L_{CE} + (1 - \alpha) . L_{KL} \tag{2.4}$$

where $\alpha \in [0, 1]$ is a balancing hyperparameter.

2.3 Neural Architecture Search (NAS):

Neural Architecture Search (NAS) [4] is an automated machine learning technique used to design neural network architectures tailored for a specific task (figure 2.1). In the context of Knowledge Distillation (KD), NAS aims to discover optimal student models also referred to as student variants that can effectively learn from a larger teacher model.

To define the student search space, multiple architectural configurations are generated using lightweight operations such as global average pooling (GAP), dropout layers, depthwise separable convolutions, and variations in network depth and width. Each student variant represents a unique architecture within the search space.

Once the space of student candidates is defined, each variant is trained individually using the KD framework described previously. Specifically, the training process incorporates both the hard labels and the soft logits produced by the teacher model. The goal is to assess how well each student performs in distillation-based learning.

After all variants are trained, their performances are evaluated and ranked based on relevant metrics such as accuracy, number of parameters, and inference speed. The best-performing architecture i.e., the one that balances accuracy and efficiency is selected as the final student model.

It is important to note that this process is computationally expensive and time-consuming, as it requires training numerous architectures separately. Therefore, applying NAS in KD scenarios demands significant computational resources and careful design of the search strategy.

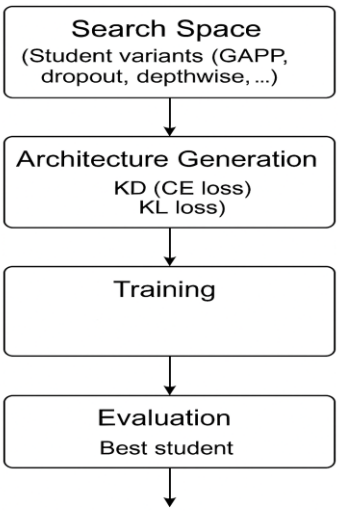


Figure 2.1: Neural Architecture Search strategies

2.4 Distillation Without Training (DisWOT)

DisWOT [3] is a training-free neural architecture search method designed specifically for Knowledge Distillation (KD). Unlike traditional NAS methods that rely on expensive training cycles, DisWOT selects the best student model without training, based on carefully designed similarity metrics computed from randomly initialized networks.

2.4.1 Search Stage

After defined the search space that are generated using lightweight operations, DisWOT's search phase relies on two main similarity metrics (figure 2.2):

2.4.1.1 Semantic Similarity

Semantic information is extracted using Grad-CAM [82] which highlights class-discriminative regions in the feature maps of a model. Given a feature tensor that represent the activations or outputs of convolutional layers A^c for class C, the purpose of this process is to calculate the similarity or correlation between the class-specific attention maps of the teacher and student models, providing insight into how well the student replicates the teacher's focus on relevant image regions. The Grad-CAM for the teacher and student is computed as:

$$G_T = \sum_{c=1}^{C_T} w_{n,c}^T . A_T^c \tag{2.5}$$

$$G_{S_i} = \sum_{c=1}^{C_S} w_{n,c}^S . A_{S_i}^c \tag{2.6}$$

where w^T and w^S are weights from the last fully connected layer, and A_T^C, A_S^C are the activation maps of the teacher and student.

The normalized correlation matrices are then computed:

$$M_T = \frac{G_T . G_T^T}{\| G_T . G_T^T \|_2} \tag{2.7}$$

$$M_S = \frac{G_S . G_S^T}{\| G_S . G_S^T \|_2} \tag{2.8}$$

Finally, the semantic similarity loss is measured as:

$$L_{Sem} = \| M_T - M_s \|_2 \tag{2.9}$$

Where $\|.\|_2$ stands for the L2 norm.

2.4.1.2 Relation Similarity

This metric measures the relational structure between samples in a mini-batch. First, the feature maps are reshaped as $\tilde{A}$, and the sample-wise correlation matrices are computed:

$$R_T = \frac{\tilde{A}_T . \tilde{A}_T^T}{\left\| \tilde{A}_T . \tilde{A}_T^T \right\|_2} \tag{2.10}$$

$$R_S = \frac{\tilde{A}_S . \tilde{A}_S^T}{\left\| \tilde{A}_S . \tilde{A}_S^T \right\|_2} \tag{2.11}$$

Where $\tilde{A}$ denotes the features map tensor extracted from a specific layer, T refers to teacher model, and S for student, and $\tilde{A}^T$ refers to transpose of $\tilde{A}$.

The relation similarity loss is then defined as:

$$L_{rel} = \| R_T - R_s \|_2 \tag{2.12}$$

Where $\|.\|_2$ stands for the L2 norm.

2.4.1.3 Evaluation and Selection

The overall DisWOT score is the sum of the two metrics:

$$DisWOT_{Score} = argmin(L_{sem} + L_{rel}) \tag{2.13}$$

An evolutionary algorithm is then applied to search for the student architecture that minimizes the DisWOT score.

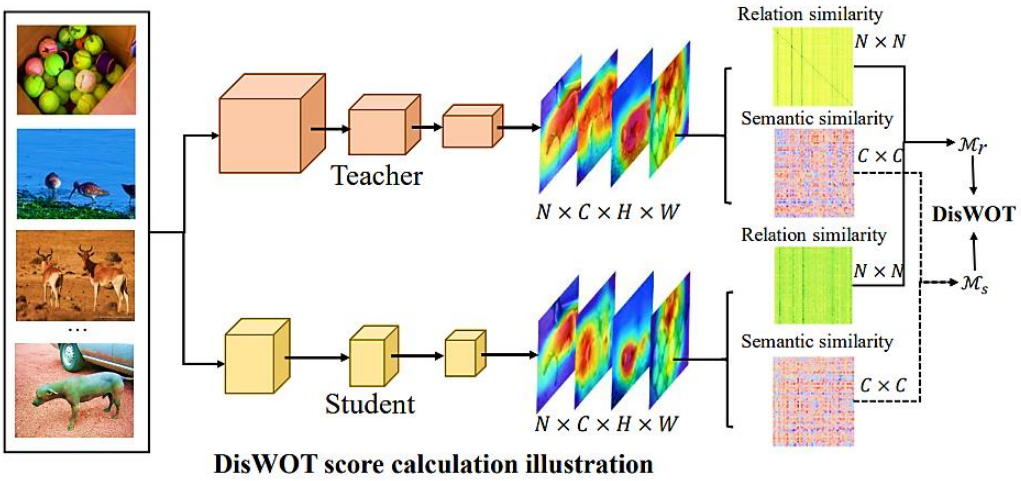


Figure 2.2: An overview of DisWOT strategies that shows how the scores are calculated [3].

2.4.2 Distillation Stage

Once the best student architecture is selected, the initial training is conducted using standard KD loss as showing in (2.4) However, DisWOT proposes to go beyond traditional KD by integrating high-order knowledge (semantic and relational similarity) directly into the training objective:

$$L_{DisWOT+} = \alpha.L_{CE} + \beta.L_{KL} + \gamma.L_{sem} + \delta.L_{rel} \tag{2.14}$$

subject to the constraint:

$$\alpha + \beta + \gamma + \delta = 1 \tag{2.15}$$

This composite loss captures richer information transfer from teacher to student and has demonstrated superior performance in experiments (as will be shown in Chapter 4 of Experimental results).

2.5 Analysis DisWOT

DisWOT originally included Semantic and Relation Similarity techniques: Grad-CAM creates a semantic relevance map and L2 similarity measures how closely aligned the teacher and student models were. However, results showed this combination did not work very well, especially during lengthy training. We describe the limitations of each of these components in the following sections.

2.5.1 Grad-CAM Limitations

Grad-CAM (Gradient-weighted Class Activation Mapping) [82] is one of the most used methods to visualize which parts of an input image are the most relevant to a model's prediction. This was accomplished by calculating the gradient of the output class score relative to the convolutional feature maps and then used these gradients to obtain a class-specific importance heatmap. The Grad-CAM heatmap for class C is given in (2.5) and (2.6).

Hereafter, we mention the main limitations of this technique:

a. Coarse Representations

Grad-CAM [82] generates maps with limited resolution because it averages the gradients over the spatial dimensions of the feature maps. The process of averaging creates a single weight for each channel. This weight is then combined with the activating map (using a linear combination) to form the final heatmap. Like previous methods, Grad-CAM's low-resolution approach has some limits as follows:

- **Loss of Fine-Grained Information:** The averaging at a fixed global scope collapses any local variations and effectively removes localized spatial cues that could be important for tasks where precise localization or segmentation is crucial.
- **Coarse Localization**: Heatmaps are predominantly large areas where the model predicts a certain class but are ill-equipped to localize intricate or overlapping features that are class-specific (figure 2.3).

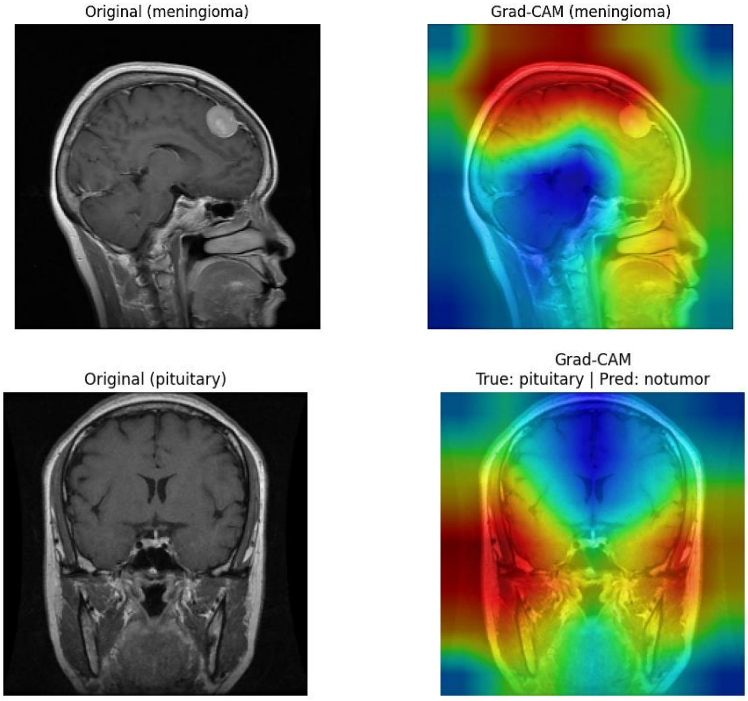


Figure 2.3: Illustration of Grad-CAM Limitations: Coarse Localization and Noisy Supervision Signals

Overall, these maps are well suited for human interpretability but poorly suited as a guidance mechanism for training student models in knowledge distillation. This is particularly problematic in deep networks when relevant high-resolution and class-discriminative features are necessary for effective knowledge transfer.

b. Weak and Noisy Supervision Signal

Grad-CAM maps identify important locations in the model's decision but class-specific or structurally coherent features may not be included. Specifically, Grad-CAM maps may retrieve general high-activation regions, which are not always truly discriminative for classification.

On the basis of knowledge distillation, the presence of such weak supervision leads to two significant phases:

- **Initial Mimicry of Coarse Patterns**: During the early phases of training, the student may maintain a good imitation of the general spatial guidance in the teacher's Grad-CAM maps, which helps align low-level features.
- **Degradation Due to Noise**: As we continue the training process, it is clear this weak supervision related to noise arose from the consequences of Grad-CAM. The lack of fine structure and semantic detail creates opportunities for the student to overfit on noisy activations. The Grad-CAM [83] maps were not reliable in identifying truly informative regions, and therefore the student model may have trained by focusing on non-discriminative features. Ultimately, there are unstable gradients and weak convergence due to non-informative features of the original images, which ultimately diminish the generalization performance of the distilled model (figure 2.4).

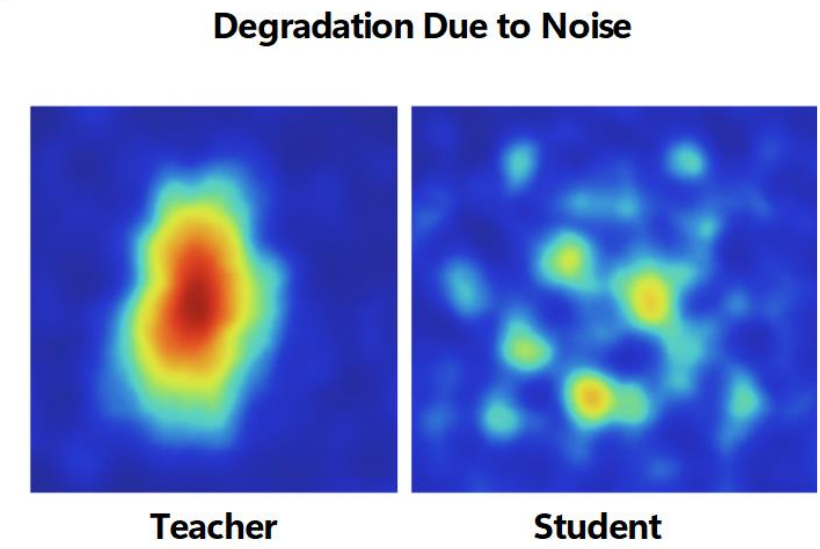


Figure 2.4: an example of degradation due to noise between Teacher model and student model.

The outlined consequences suggest that although Grad-CAM can have interpretability values, it is not an effective way to directly supervise fine-grained feature alignment in distillation frameworks - which is especially problematic with complex or hierarchical tasks.

This figure shows two examples highlighting the limitations of Grad-CAM in the context of brain tumor detection. (Top) For a meningioma case, the Grad-CAM highlights a broad area, missing the precise tumor location illustrating the issue of coarse spatial representations. (Bottom) Grad-CAM produces a misleading activation map for a no-tumor image, resulting in a false prediction of a pituitary tumor demonstrating weak and noisy supervision.

2.5.2. Limitations of using L2 Similarity

In many situations of knowledge distillation, especially the feature-based supervision ones, L2 loss (squared Euclidean distance) is widely used to represent the distance between the feature maps

or relevance maps of the teacher and student processes. Specifically, it is defined mathematically as follows:

$$L_{l2} = \|M_T - M_S\|_2^2 \tag{2.16}$$

Where:

- M_T represents the feature or relevance map of the teacher model.
- M_S represents the corresponding feature or relevance map of the student model.
- $\|.\|_2^2$ denotes the squared L2 norm, which calculates the sum of squared differences between corresponding elements in the two maps.

This loss function encourages the student model to create internal representations that are numerically close to the representations of the teacher. While it is intuitive and simple to implement, the L2 loss function has several important drawbacks that may hinder the training process and final performance of the student model.

a. Sensitivity to Scale

One of the most significant drawbacks of L2 loss is its high sensitivity to the magnitude (or scale) of the feature activations. Even if the spatial structure or directional pattern of the student's feature map is similar to that of the teacher's, a small scaling difference such as slightly higher or lower values can lead to a disproportionately large L2 loss.

Imagine that the teacher's feature map values are all twice as large as the student's. Even though the spatial arrangement and overall semantic pattern are similar, the L2 loss will be high because it measures absolute differences in values. This sensitivity can result in two key issues:

- **Penalty for nearly worthless activations**: Components of a feature map with slight differences in intensity or scaling that do not change the semantic meaning of the feature map are also penalized. This could result in the optimizer treating the minor differences as serious penalty errors in the optimization process [1].
- **Imposition to the learning process**: The model is unable to discover alternative internal representations that may still be effective since the model is obliged to reproduce exact values. It is forced to reproduce the teacher's outputs as best it can, when it could perhaps perform equally well using different internal representations [69].

b. L2 loss as over 'regularization' in later early epochs

More experienced students will also suffer when using L2 loss, as it is over-regularization. The last half of the epochs, especially once the student model has learned to reproduce the teacher at a sufficiently high level of accuracy, the differences between the teacher's and student's feature maps become quite small and are often semantically insignificant.

Nevertheless, the L2 loss function continues to penalize even the smallest of differences leading to a number of disadvantages:

- **Higher Training Loss**: the loss will be unreasonably high since there exist small numerical differences that do not impact model behavior and will lead to emanating misleading signals to the optimizer.

- **Limited Generalization Capability**: the student might be too biased to learn internal representations that elephant the teacher rather than generalizable correlations and patterns. By reducing the focus away from replicating the teacher's internal representations in favor of seeking generalizable patterns, the student will likely fail to generalize to data it receives in the future.
- **Potential for Underfitting or Unstable Training**: the model may be penalizing itself too aggressively for small L2 errors and potentially cause the model to update parameters on a broader section of the sample incorrectly, leading to unstable gradients, an improper convergence path, or even total collapse of performance (the model accuracy will decrease after a specific number of effects leading to serious underfitting problem). This underfitting effect would be even more pronounced when it's paired with relevance maps from techniques such as Grad-CAM, which are already coarse and noisy.

While L2 loss is a well-known and mathematically simple choice, it is not always ideal to optimize deep representation alignment, particularly with knowledge distillation when semantic knowledge, rather than just a numerical value, needs to be transferred. A more robust solution is to use direction-based similarity metrics such as cosine similarity, which only compares the angle between feature vectors, rather than their magnitude. This means that semantic structure will be preserved in the features, while irrelevant differences in scale won't matter, improving convergence, improving generalization, and more meaningful knowledge transfer.

2.6 Proposed Improvements

2.6.1 Layer-wise Relevance Propagation (LRP) for High-Order Semantic and Relational Supervision

To directly address Grad-CAM's limitations: coarse spatial resolution, lack of fine-grained localization, and increased sensitivity to magnitude of features, we propose a far more robust and precise extraction of semantic information: layer-wise relevance propagation (LRP). More so, to enhance the alignment of source and target models, we replace L2 similarity (also an approach to represent the distance of the teacher and student model) with cosine similarity, which is more stable and scale-invariant. Both the search stage (model to use) and the training stage (high-order knowledge distillation) will use each of the modifications above. Therefore, together as a complete and principled framework.

While Grad-CAM is often used to provide visual interpretability, the class activation maps produced by Grad-CAM are based on the gradients and pooled activations in the convolutional layer(s), resulting in spatial coarseness. These maps localize a general region of interest but will likely fail at providing the precise location of which neuron or certain spatial location is mostly responsible for making the decision. This spatial coarseness is particularly detrimental in an educational context such as knowledge distillation as proper semantic guidance is essential for making it possible for the student model to receive meaningful knowledge from the teacher model.

The advantage of LRP is that it offers a more fine-grained and class-specific way of explaining model predictions. Rather the average gradient values over the feature maps (like Grad-CAM), LRP iterates the final output score of the model backwards through the network with respect to the

output of each neuron, hence developing relevance scores for each neuron or feature according to their contribution to the final output decision (figure 2.5).

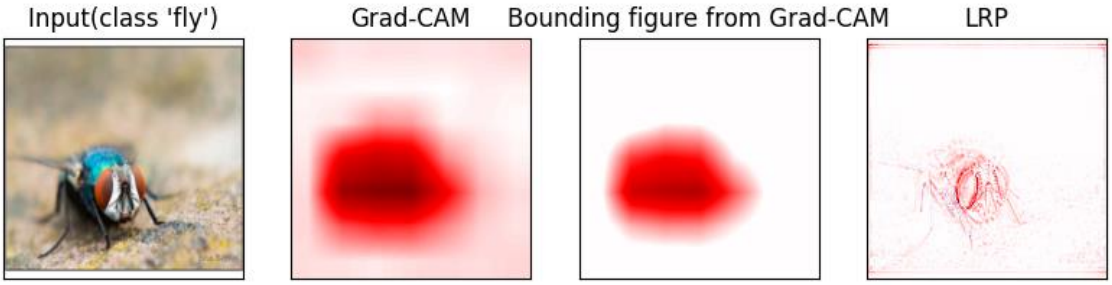


Figure 2.5: An example of the working of Grad-CAM and LRP [84]

The core principle of LRP is to decompose the prediction score S_c for class C by assigning to each neuron i a relevance score R_i, which indicates how much that neuron contributed to the final prediction. The basic rule can be expressed as:

$$R_i = a_i . \frac{\partial S_c}{\partial a_i} \tag{2.17}$$

Where:

- a_i is the activation of neuron i.
- $\frac{\partial S_c}{\partial a_i}$ is the partial derivative of the class score with respect to the activation.

This equation expresses the product of how strongly the activation is activated with how sensitive the score is to that activation, so it provides a reasonable measure of importance.

In practice, to generate relevance maps over feature maps A^k (channels of output from convolutional function), we realize the following LRP approximation using ReLU to ignore negative attributions:

$$R = ReLU\left(\sum_k A^k . \frac{\partial S_c}{\partial A^k}\right) \tag{2.18}$$

This yields a spatial relevance map that:

- Identifies pixels or features that positively contributed to the prediction.
- Provides higher spatial resolution and is class-specific, compared to Grad-CAM.
- Maintains the structural integrity of the feature representations, increasing distillation reliability.

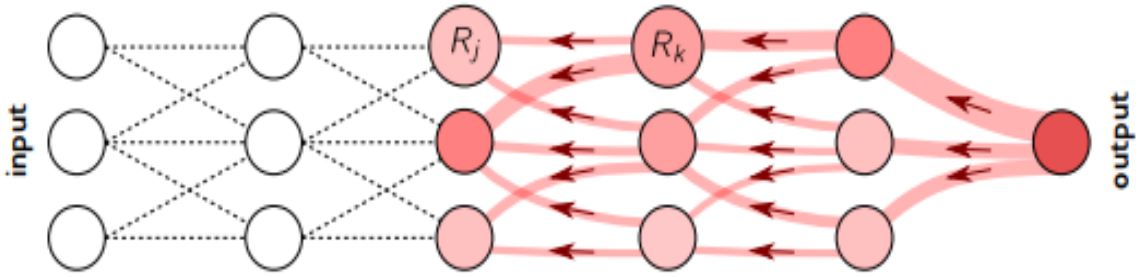


Figure 2.6: Illustration of the LRP procedure. Each neuron redistributes to the lower layer as much as it has received from the higher layer (relevance is propagated layer by layer) [85].

2.6.2 Cosine Similarity for High-Order Semantic and Relational Supervision

In the original framework, the similarity between the teacher's and student's relevance maps was computed using the L2 distance:

$$L_{l2} = \|R_T - R_S\|_2^2 \tag{2.19}$$

However, as explained Section 3.2 above, L2 loss is sensitive to the scale of activations and this could confuse training as long as the student's structured activations are aligned but they differ in scale.

To remedy this, we utilize cosine similarity in both semantic and relation framework which treats the angular distance between a flattened relevance vector. The new loss is:

$$Cosine_{sim}(R_T, R_S) = \frac{R_T . R_S}{\|R_T\|_2 . \|R_S\|_2} \tag{2.20}$$

$$L_{Semantic}^{LRP} = 1 - Cosine_{sim}(R_T, R_S) \tag{2.21}$$

Where:

- R_T and R_S are the flattened relevance maps from the teacher and student respectively.
- The result lies in the range [0,2] where lower values indicate stronger alignment.

The key benefit of cosine similarity compared to L2 is that cosine similarity is invariant to the magnitude of the vector, so it highlights orientational congruency. This can allow the student model to represent similar semantics and structure similar representations even if the magnitudes of the internal representations differ from the teacher. This is particularly important when optimally classifying parameters (improving generalization), since in deep networks, internal representations may vary amplitudes through layers or models but can encode the same semantics.

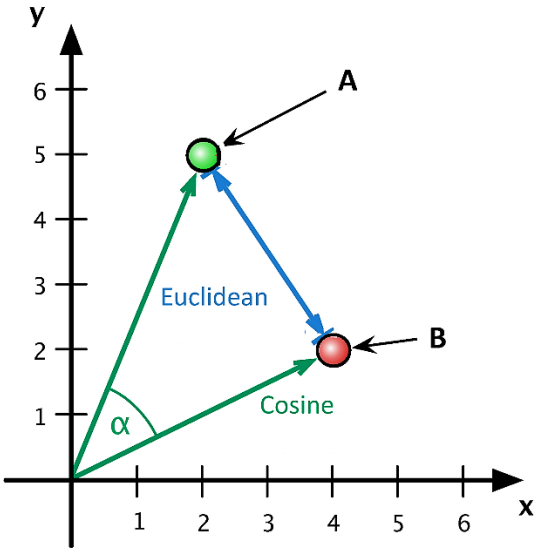


Figure 2.7: L2 Distance Vs Cosine

Utilizing cosine similarity allows us to build strong alignment of information in high-level representations so that the student grabs hold of the essence of the teachers' knowledge while also removing the penalty for inconsequential scale.

This metric is concerned primarily with directional consistency, rather than matching values exactly, allowing the student to effectively learn semantically similar representations even if the internal scaling is different. This is important for good generalization.

Criteria	**Grad-CAM + L2**	**LRP + Cosine**
Spatial precision	Low (coarse maps)	High (fine-grained)
Sensitivity to scale	High	Low (scale-invariant)
Class specificity	Moderate	Strong
Stability during training	Unstable in late stages	More robust
Feature alignment quality	Strict but noisy	Semantically accurate
Interpretability	General localization	Detailed, pixel-level attribution

Table 2.1: Key Difference between Grad_CAM + L2 Vs LRP + Cosine

2.6.3 Algorithm to avoid redundancy

To improve the quality and strength of knowledge transfer from teacher to student, we introduce a more refined supervision strategy - cosine similarity - within a layer-wise relevance propagation (LRP) context. This approach is an extension of our previous findings showing that cosine-based loss functions outperformed traditional L2 distance in describing distances between deep neural relevance representations.

Cosine similarity measures the angle between two vectors, revealing their similarity in direction rather than in magnitude. This property has specific utility in the context of deep neural representation learning, where two relevance maps can have semantically consistent structures, but differ/rescale in magnitude due to architectural aspects (e.g., batch normalization, dropout, "internal" parameter scaling, etc...). Given that cosine similarity prioritizes directional agreement over similarity in absolute value, it can be conceived as a more semantically faithful and generalizable supervision signal. The proposed loss function is defined as follows:

For Semantic loss:

$$L_{Semantic} = 1 - \frac{R_T . R_S}{\|R_T\|_2 . \|R_S\|_2} \tag{2.22}$$

Where R_T and R_S are the flattened relevance maps from the teacher and student respectively, this loss is computed layer-wise, and then averaged across selected layers.

For Relation loss:

First Compute relational matrices

$$M_T = A_T . A_T^T \tag{2.23}$$

$$M_S = A_S . A_S^T \tag{2.24}$$

Then normalized M_T and M_S by the corresponding norm:

$$M_T = \frac{M_T}{\|M_T\|} \tag{2.25}$$

$$M_S = \frac{M_S}{\|M_S\|} \tag{2.26}$$

To become the relation loss as:

$$L_{Relation} = 1 - \frac{M_T . M_S}{\|M_T\|_2 . \|M_S\|_2} \tag{2.27}$$

Where A_T and A_S are the features maps from the teacher and student respectively, and this also averages across selected layers.

In addition, we use Cross entropy loss function for supervised loss from ground truth labels and KL Divergence Loss (Knowledge Distillation Loss) for soft target loss between teacher and student outputs define as follows:

Cross-Entropy Loss (CE Loss):

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{2.28}$$

Where:

- y_i is the ground truth one-hot label.

- $\hat{y}_i$ is the predicted probability from the student model.

KL Divergence Loss:

$$L_{KL} = T^2 . \sum_{i=1}^{C} P_T^{(i)} . \log(\frac{P_T^{(i)}}{P_S^{(i)}}) \tag{2.29}$$

Where:

- C: number of classes
- T: temperature parameter (e.g., 3 or 4)
- $P_T^{(i)} = softmax(\frac{Z_T^{(i)}}{T})$: softened probability for class 3 from teacher logits
- $P_S^{(i)} = softmax(\frac{Z_S^{(i)}}{T})$: softened probability for class 3 from student logits
- $Z_T^{(i)}$ and $Z_S^{(i)}$ the logits (pre-softmax outputs) for class 3 from teacher and student respectively

The total proposed loss keeps as same in (2.14).

Setup	Accuracy	Stability	Remarks
Grad-CAM + L2	Good	Poor	Diverges after certain number of epochs
LRP + L2	Good	Good	Stable and interpretable
LRP + Cosine	Best	Best	Robust, best convergence

Table 2.2: Comparison between different setup approach combinations on accuracy and stability

2.7 Conclusion

In this chapter, we have proposed a novel knowledge distillation framework that supports a reliable approach to student model selection and training, as well as the establishment of a reliable distillation framework. We found that while DisWOT can be used as training-free based on Grad-CAMs and L2 similarity metrics, our exploration highlighted major limitations with both of these approaches. Grad-CAM produced coarse and noisy subjective spatial maps of attribution, and in similarity calculations, L2 is easily misled by scale... in this case, we conclude that a distance is too sensitive. To overcome these challenges, we proposed the use of Layer-wise Relevance Propagation (LRP) for generating fine-grained, class-specific semantic maps, and cosine similarity for stable, scale-invariant relational alignment.

These improvements were integrated into both the search and distillation stages, leading to a more robust and interpretable student selection and training process. Comparative evaluations demonstrated that the LRP + Cosine framework yields higher accuracy, better generalization, and more stable convergence than traditional Grad-CAM + L2 setups. Thus, our method offers a principled and efficient advancement in knowledge distillation for lightweight model design.

Chapter 3

Experimental Results

3.1 Introduction

Following the detailed presentation of our proposed framework and its implementation in the previous chapter, this chapter focuses on the experimental validation of our contributions. We will present the dataset used for training and testing, outline the evaluation metrics employed to assess performance, and provide a comprehensive analysis of the experimental results.

The goal of this chapter, which should be apparent in our experimental evaluations, was to exhibit support for our enhanced knowledge distillation framework, including the use of Layer-wise Relevance Propagation (LRP) and cosine similarity, to allow students to obtain additional supervision through semantic and relational information. This chapter also has interest on the lightweighting objective, where students are evaluated on more than accuracy, we also consider the compute performance (FLOPs, MACs, and parameters).

3.2 Experimental Dataset

To assess the performance of our method, we used the publicly available Brain Tumor MRI [86]Dataset from Kaggle, which contains a diverse and labeled collection of MRI images categorized into four tumor types: glioma, meningioma, pituitary, and no tumor, figure 3.1 shows the distribution of classes in training set. This dataset provides a reliable and representative benchmark for evaluating classification models under realistic medical imaging conditions.

The dataset includes a total of approximately 7,000 images, split into training and testing subsets with the following structure:

- Training set: 5,170 images
- Testing set: 1,370 images
- Image format: JPG
- Image dimensions: Resized to 224×224 pixels
- Color mode: Converted to RGB to match pretrained model requirements

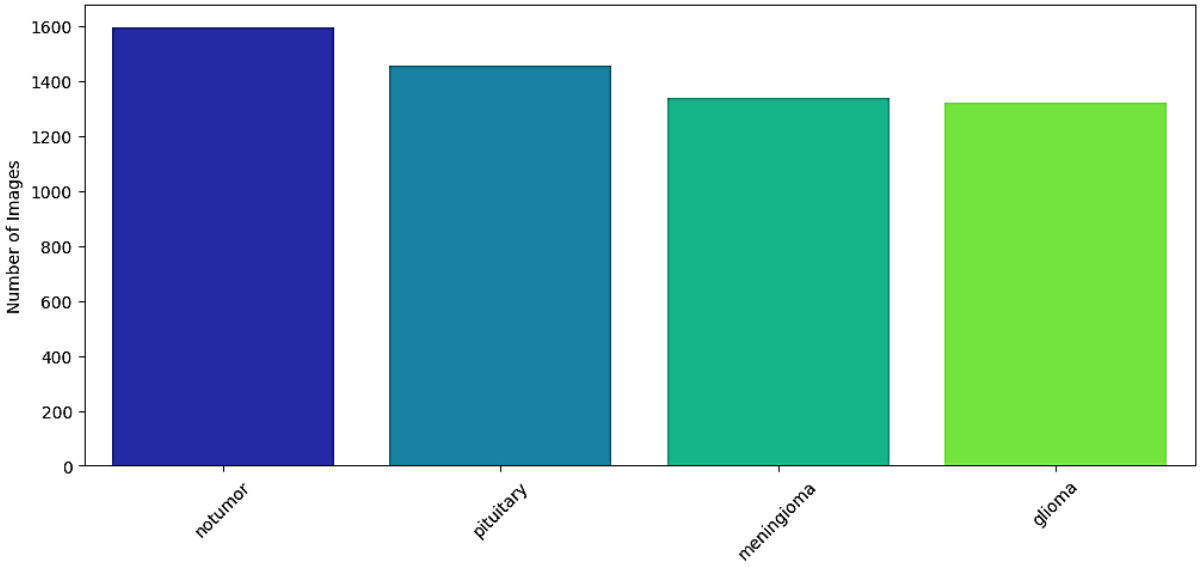


Figure 3.1: Training Distribution of Classes in the dataset

Each image is assigned a ground truth label corresponding to one of the four classes as it represents in figure 3.2. The dataset is relatively balanced and allows robust model evaluation across all categories.

Prior to training and inference, a standard preprocessing pipeline was applied to ensure uniformity and compatibility with deep learning architectures:

- Resizing all images to 224×224×3 resolution
- Pixel normalization to [0, 1] range
- One-hot encoding of categorical labels
- Data augmentation including random flips, rotation, and zoom to enhance generalization and reduce overfitting.

We selected this dataset for the variability and medical significance, which make it appropriate for assessing the performance of various strategies for knowledge distillation, and the performance of lightweight models under the intensive setting of clinical practice. Also, the multiple tumor types offered in this dataset align well with our intention to explore multi-class classification performance for both teacher and student models.

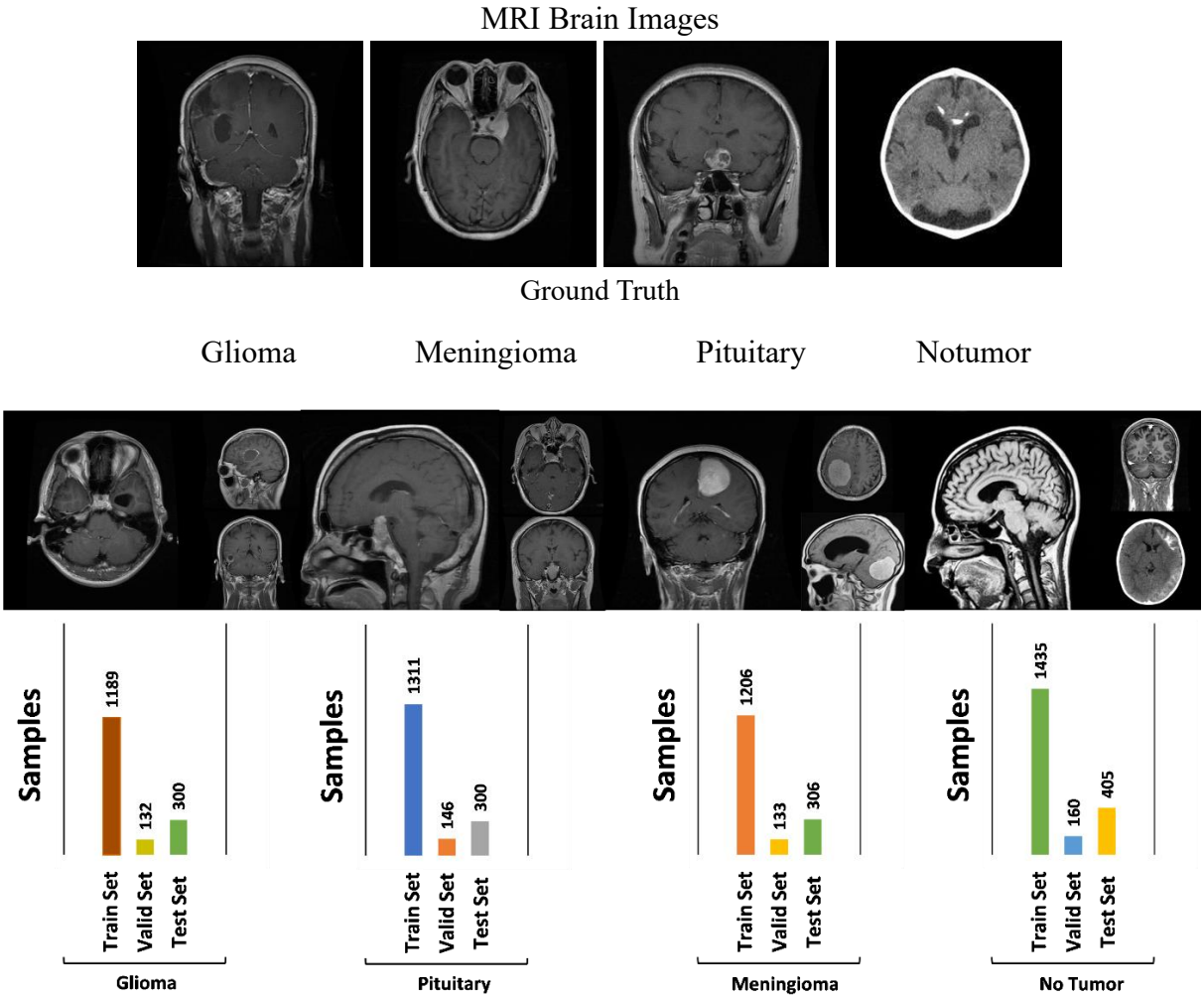


Figure 3.2: Examples of data from the Brain Tumor MRI dataset

3.3 Evaluation Metrics

To assess the proposed framework comprehensively, we evaluated both the prediction quality and computational efficiency of the models. The classification performance was measured using standard metrics including accuracy, precision, recall, and F1-score, while the lightweight nature of the student networks was evaluated using FLOPs, MACs, number of parameters, and model size.

3.3.1 Classification Metrics

The following metrics were used to evaluate model performance on the brain tumor classification task:

3.3.1.1 Accuracy

In machine learning evaluating each task is crucial and the accuracy metric is particularly useful across various applications such as classification detection and segmentation. Accuracy measures the proportion of correctly classified data points out of the total number of input samples. It is calculated as follows:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (3.1)$$

Accuracy provides a straightforward measure of a model's performance making it especially effective when classes are balanced. It reflects the overall effectiveness of the model in assigning the correct labels to the input data.

3.3.1.2 Confusion matrix:

In brain tumor classification, the confusion matrix is computed at the image level rather than the pixel level, as the task involves assigning a single diagnostic label to each MRI scan. For multi-class classification involving four tumor categories glioma, meningioma, pituitary, and no tumor the confusion matrix provides a detailed overview of the model's predictive performance across all classes. Each entry in the matrix reflects the number of images from an actual class that were predicted as a specific class, thereby enabling the identification of true positives, false positives, false negatives, and true negatives for each category. This structure is particularly valuable in medical diagnosis, where misclassification of tumor types can lead to significant clinical consequences.

Actual/Predicted	Negative	positive
Negative	TN	FP
Positive	FN	TP

Table 3.1: Confusion Matrix

- **True Positive (TP):** The number of brain MRI images correctly classified as belonging to a specific tumor class (e.g., correctly predicted as glioma).
- **True Negative (TN):** The number of images correctly classified as not belonging to a specific tumor class (e.g., images not of glioma that were also not predicted as glioma).
- **False Positive (FP):** The number of images incorrectly classified as belonging to a specific class, while they actually belong to another (e.g., non-glioma images misclassified as glioma).
- **False Negative (FN):** The number of images incorrectly classified as not belonging to a specific class, while they actually do (e.g., glioma images misclassified as another tumor type).

From these four components, a range of derived performance metrics can be calculated, providing deeper insights into the diagnostic reliability of the model. These include precision, recall, F1-score, which are discussed in detail in the following subsection.

$$Accuracy = \frac{TP + FN}{TP + FN + TN + FP} \quad (3.2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.3)$$

$$Recall(Sentivity) = \frac{TP}{TP + FN} \tag{3.4}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.5}$$

3.3.2 Efficiency Metrics

To validate the lightweight nature of the student architectures, we employed the following metrics:

3.3.2.1 FLOPs (Floating Point Operations)

Measures the total number of floating-point operations during a forward pass. Lower FLOPs imply lower computational cost.

$$FLOP_S = 2.H.W.C_{in}.C_{out}.K.K$$

where:

- H and W are the height and width of the output feature map
- C_{in} and C_{out} are the number of input and output channels, respectively
- K is the kernel size
- and the factor of 2 accounts for both the multiplication and addition operations in each MAC.

3.3.2.2 MACs (Multiply-Accumulate Operations)

Multiply-accumulate operations are a common benchmark for hardware efficiency. MACs are equivalent to roughly half the FLOPs for CNNs

$$MAC_S = H.W.C_{in}.C_{out}.K.K = \frac{1}{2} \times Flop_s \tag{3.6}$$

3.3.2.3 MAC (Memory Access Cost):

is distinct from Multiply-Accumulate operations. It refers to the number of memory accesses required to read inputs, write outputs, and retrieve weights. This cost is significant in edge and embedded systems, where memory bandwidth and energy are constrained.

An approximation of the Memory Access Cost is:

$$\text{Memory Access Cost } (MAC) = H \cdot W\ (Cin + Cout) + k \cdot k\ (Cin \times Cout) \tag{3.7}$$

3.3.2.4 Number of Parameters

Total learnable weights in the model. Lower parameter count indicates a smaller and more efficient model.

$$Parameters = (k^2 \times C_{in} \times C_{out}) + C_{out} \tag{3.8}$$

3.4 Implementation details

Teacher Model For the teacher network, we employ MobileNetV2, a lightweight yet powerful convolutional neural architecture designed for efficient performance on resource-constrained devices. The model is trained on a dataset consisting of 5,170 training images and 1,370 testing images, with each input resized to 224 × 224 × 3 to match the network's requirements. Training is conducted using a batch size of 32, and the model is optimized over 50 epochs using a learning rate of 0.0001. This teacher achieves **98.02%** validation accuracy as the high-capacity reference for subsequent student model selection and knowledge distillation tasks. Its combination of strong representational capacity and moderate computational footprint makes it an ideal candidate for driving a lightweight distillation framework.

In the knowledge distillation process, we adopt the same hyperparameters as used in training the teacher (table 3.2). Additionally, we set the temperature (T) parameter to 4. This temperature softens the teacher's output probabilities, allowing the student model to better capture the teacher's dark knowledge that is, the relative probabilities across classes, which often carry more nuanced information than hard labels alone.

Model	Learning Rate	Number of Epochs	Optimizer	Batch Size
Teacher (MobileNetV2)	0.0001	50	Adam	32
Distillers Models (MobileNetV2 Variant)	0.0001	15, 30, 50	Adam	32

Table 3.2: Hyperparameter Settings for Teacher and Student Models

3.5 Experimental Results

3.5.1 Justification of Low DisWOT Score as an Indicator of Better Distiller

In this section, we examine the hypothesis that the DisWOT score, a training-free metric used to evaluate untrained student models based on their semantic and relational similarity to a teacher, can reliably predict student models with superior distillation performance. The central idea is that students with lower DisWOT scores are structurally and semantically better aligned with the teacher, and therefore should exhibit better generalization after distillation.

To test this hypothesis, we selected three student models from the search space that contain 16 student's variants created from the teacher (MobileNetV2) that are generated using lightweight operations with varying DisWOT scores representing high (bad), moderate (intermediate), and low (best) score categories and evaluated their performance after undergoing the same distillation procedure. All models were trained under identical settings with a learning rate of 0.0001, batch size of 32, and trained for 15 epochs.

Model Quality	Test Accuracy (%)	Train Accuracy (%)	Parameters (#)	DisWOT Score	Observations
Best	90.92	92.99	2,257,984	1.35	Best test accuracy, lowest DisWOT score
Intermediate	86.19	93.00	410,208	138.84	Moderate test accuracy despite high training accuracy
Bad	84.52	85.11	410,208	176.12	Worst performance, highest DisWOT score

Table 3.3: Accuracy metrics and number of parameters with DisWOT Score for various models

Based on table 3.3 and figure 3.3 find the results show a clear inverse correlation between the DisWOT score and the final test performance of student models. Notably, the student with the lowest DisWOT score (1.35) achieved a test accuracy of 90.92%, significantly outperforming the other candidates. The intermediate model, despite having the highest training accuracy (93.00%), showed a noticeable drop in test accuracy (86.19%), indicating signs of overfitting where the model memorizes training patterns but fails to generalize. On the other hand, the bad model demonstrated low accuracy on both training (85.11%) and testing (84.52%) sets, suggesting weak feature learning capacity. These patterns reinforce the reliability of DisWOT as a selection metric, with lower scores aligning consistently with better generalization performance.

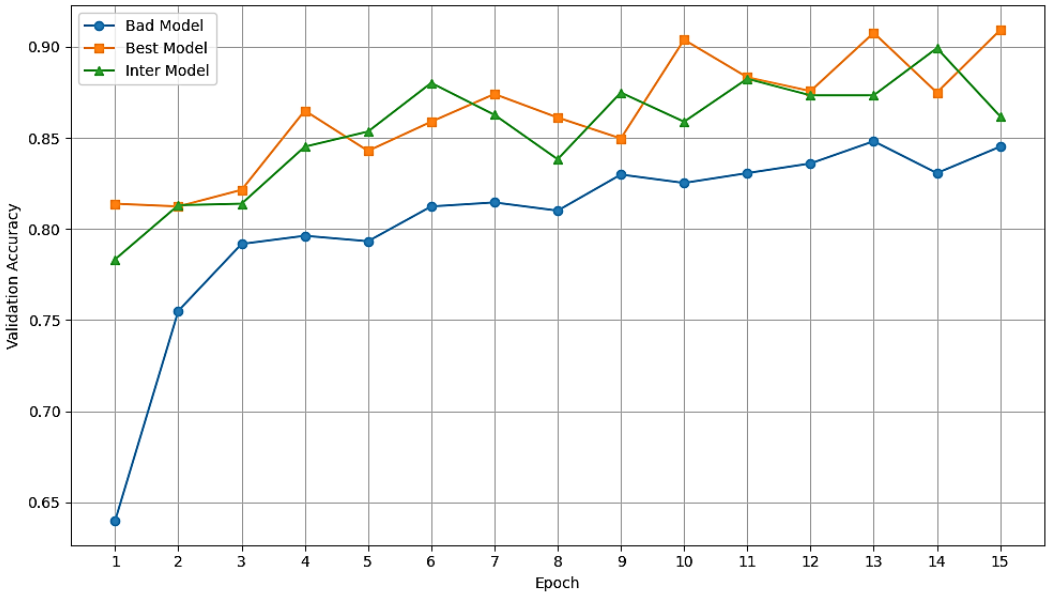


Figure 3.3: Validation Accuracy Comparison

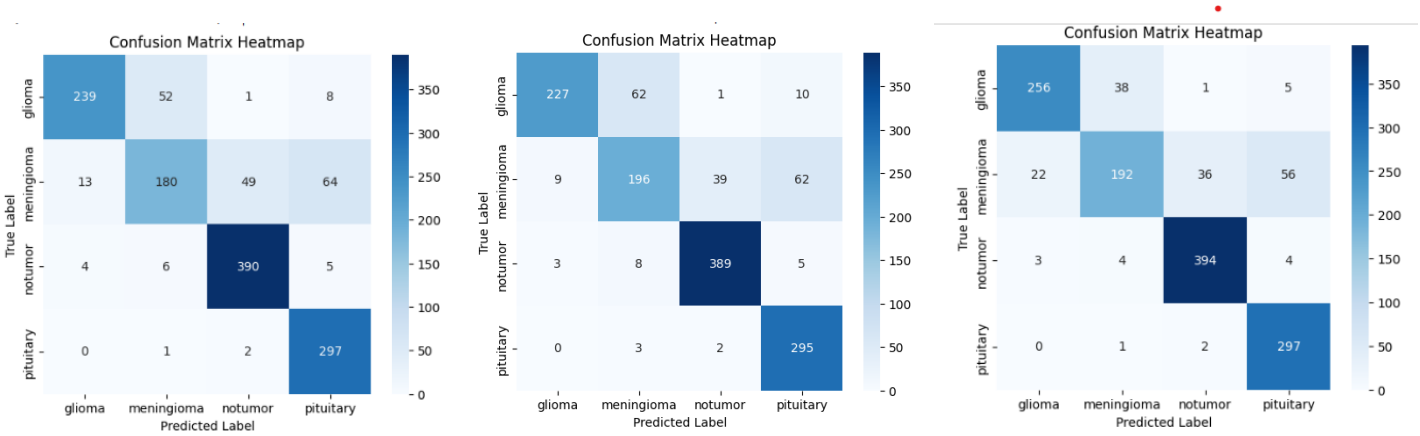
3.5.2 Comparison of Training Strategies (High Order (DisWOT+), Baseline, and Standard KD)

In this step we aim to obtain the experimental findings of training and evaluating three models: a baseline student model (trained in traditional way with data only and without distillation), a standard Knowledge Distillation (KD) setup, and a High-order distillation approach using DisWOT.

The goal is to assess how each method affects the student model's learning and analyzing the performance of each approach over 10 epochs:

Model	Train Accuracy	Validation Accuracy	Observation
Baseline	0.8749	0.8444	Signs of overfitting, the model achieves higher training accuracy but show poorer generalization on the validation set.
Standard KD	0.8875	0.8459	It shows a similar performance to the baseline model, without any noticeable improvement and a slight decrease in the accuracy.
High-Order KD (DisWOT)	0.9124	0.8688	It performs better than both baseline and standard KD, the benefits form the extra structural information provided by the teacher.

Table 3.4: The accuracies of validation and training from evaluating the three setups



(a) (b) (c)

Figure 3.4: the confusion matrices heatmap of the three setups, (a) is the confusion matrix of the standard Knowledge Distillation (KD) approach, (b) is the confusion matrix of the baseline model, (c) is the confusion matrix of high-order distillation.

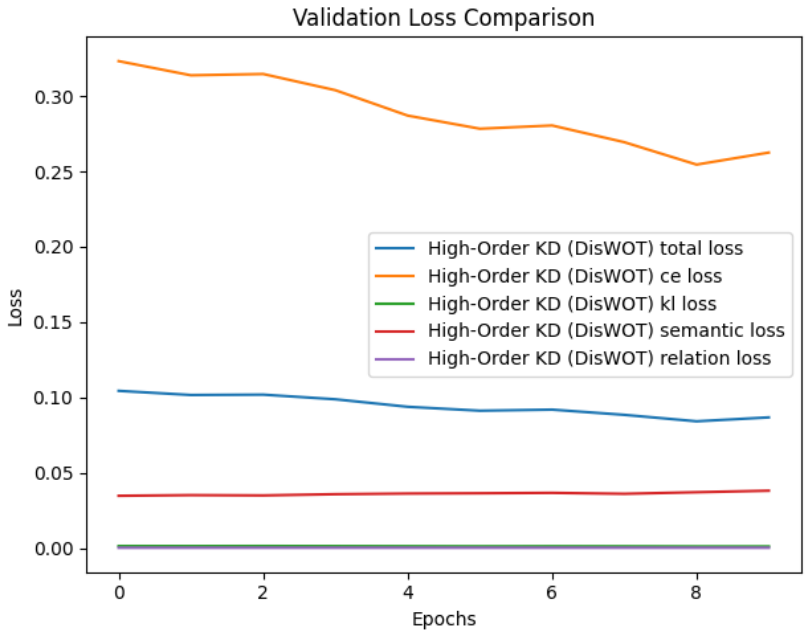


Figure 3.5: Comparison of validation loss plots for the high-order loss functions.

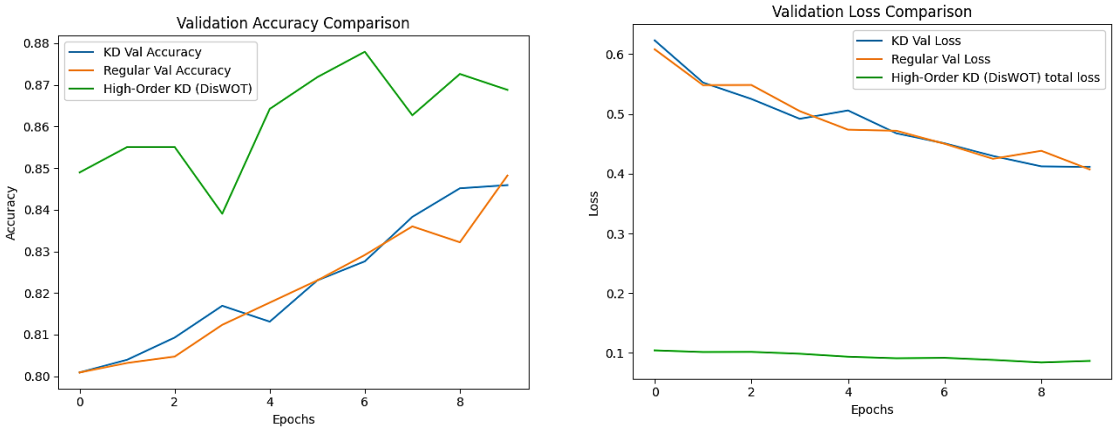


Figure 3.6: Validation Accuracy & loss Comparison plot of the three setups.

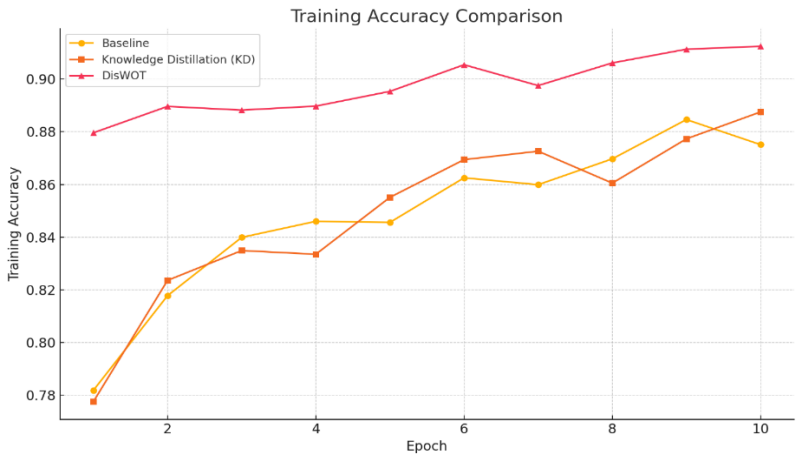


Figure 3.7: Training accuracy curve comparing the three models over 10 epochs.

The losses represented in the plot illustrates the validation loss comparison among three models (figure 3.5):

- The semantic loss minimizes the distance between intermediate feature representations of the teacher and student (e.g., via cosine similarity or MSE). These representations are encoded as normalized correlation matrices M_T and M_s, which are computed from G_T and G_{S_i} (refer to Equations 2.6 and 2.7). The loss is defined as:

$$L_{Sem} = \|M_T - M_s\|_2 \tag{3.9}$$

- The relation loss ensures that the student captures the pairwise relationships between samples encoded by the teacher. First, the feature maps are reshaped as $\tilde{A}$, then sample-wise correlation matrices are computed as:

$$R_T = \frac{\tilde{A}_T . \tilde{A}_T^T}{\left\|\tilde{A}_T . \tilde{A}_T^T\right\|_2} \tag{3.10}$$

$$R_S = \frac{A_S . A_S^T}{\|A_S . A_S^T\|_2} \tag{3.11}$$

Then the loss is defined as:

$$L_{rel} = \|R_T - R_s\|_2 \tag{3.12}$$

- The cross-entropy (CE) loss

$$L_{CE} = -\sum_i y_i \log \sigma(z_{si}) \tag{3.13}$$

where z_{si} are the student's logits, and $\sigma(.)$ denotes the softmax function.

Soft labels, which come from the teacher model's softened output. These are obtained by applying a softmax function with a temperature $T > 1$, which produces smoother probability distributions:

$$\sigma(z_i, T) = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j exp(\frac{z_j}{T})} \tag{3.14}$$

Where Z_{ti} is the teacher's logits for class i.

- the Kullback–Leibler divergence (KL loss) measures the distance between the softened output distributions of the teacher and the student:

$$L_{KL} = T^2 \sum_i \sigma(Z_{ti}, T) \log\left(\frac{\sigma(Z_{ti}, T)}{\sigma(Z_{si}, T)}\right) \tag{3.15}$$

The validation loss curves (figure 3.6) reveal key differences between these strategies. The baseline and the standard KD models have a similar validation loss trajectory, with decreasing and converging the values gradually around (0.41) in the epoch 10. The KD model shows a slightly smoother loss curve, substantially no improvement over the baseline model. Which indicates that the standard KD only transfers the softened output of the teacher's probabilities to the student model. This means no significant benefits in terms of reducing validation loss or enhancing generalization.

In the other hand, the high-order KD model using DisWOT shows a consistently lower validation loss throughout the training process, stabilizing around (0.08 and 0.1). This is supported by the detailed loss breakdown, which shows that the total loss on DisWot is combination of several losses: cross-entropy (CE), KL divergence, semantic loss, and relation loss. Among the previous losses CE loss contributes the most and declines steadily, indicating that the model aligns well with the true labels, the KL and semantic, relation losses are low and stable referring that the student effectively captures both the internal relational and outputs probabilistic of the teacher model. This combination of loss structures highlights the depth of supervision in DisWOT, where the student use more than the outputs, it also uses how knowledge is structured internally in the teacher model.

Examining the validation accuracy (figure 3.6), the baseline reaches an accuracy of (84.44%), representing a reference point of performance without any distillation. The standard KD achieves only marginal improvement, reaching (84.59%), which implies that standard KD has no significant enhance in the model's ability to generalize. In contrast, the DisWOT-based model achieves the highest accuracy of (86.88%), clearly outperforming both the baseline and standard KD. This gain confirms that the integrating of high-order information such as semantic and relationships leads to better generalizing to unseen data.

The training accuracy curves of the three model Baseline, Knowledge Distillation (KD, and DisWOT reveal distinct performance patterns over the 10 training epochs) (figure 3.7). The Baseline model begins with a moderate accuracy of 78.19% and gradually improves to 87.49% by the final epoch, showing steady learning progress. The KD model starts slightly lower at 77.76% but quickly catches up, ending with a training accuracy of 88.75%, indicating the benefits of teacher-guided learning. In the other hand, the DisWOT model demonstrates the highest and most consistent performance throughout training, beginning at 87.96% and reaching 91.24% accuracy by the tenth epoch. This suggests that the additional relational and semantic guidance in DisWOT contributes to more effective and efficient learning compared to both the Baseline and KD models.

3.5.3 Effect of LRP and Cosine Similarity

In this section, we present an enhancement to DisWOT framework by incorporating two key modifications: replacing Grad-CAM with Layer-Wise Relevance Propagation (LRP) for interpretability, and substituting the L2 loss with Cosine loss for measuring feature similarity.

To test this improvement, we conducted a set of controlled experiments using the best-performing distillation model specifically, the one that achieved the lowest DisWOT score in our prior analysis. first using Grad-Cam + L2 (This represents the original configuration used in our

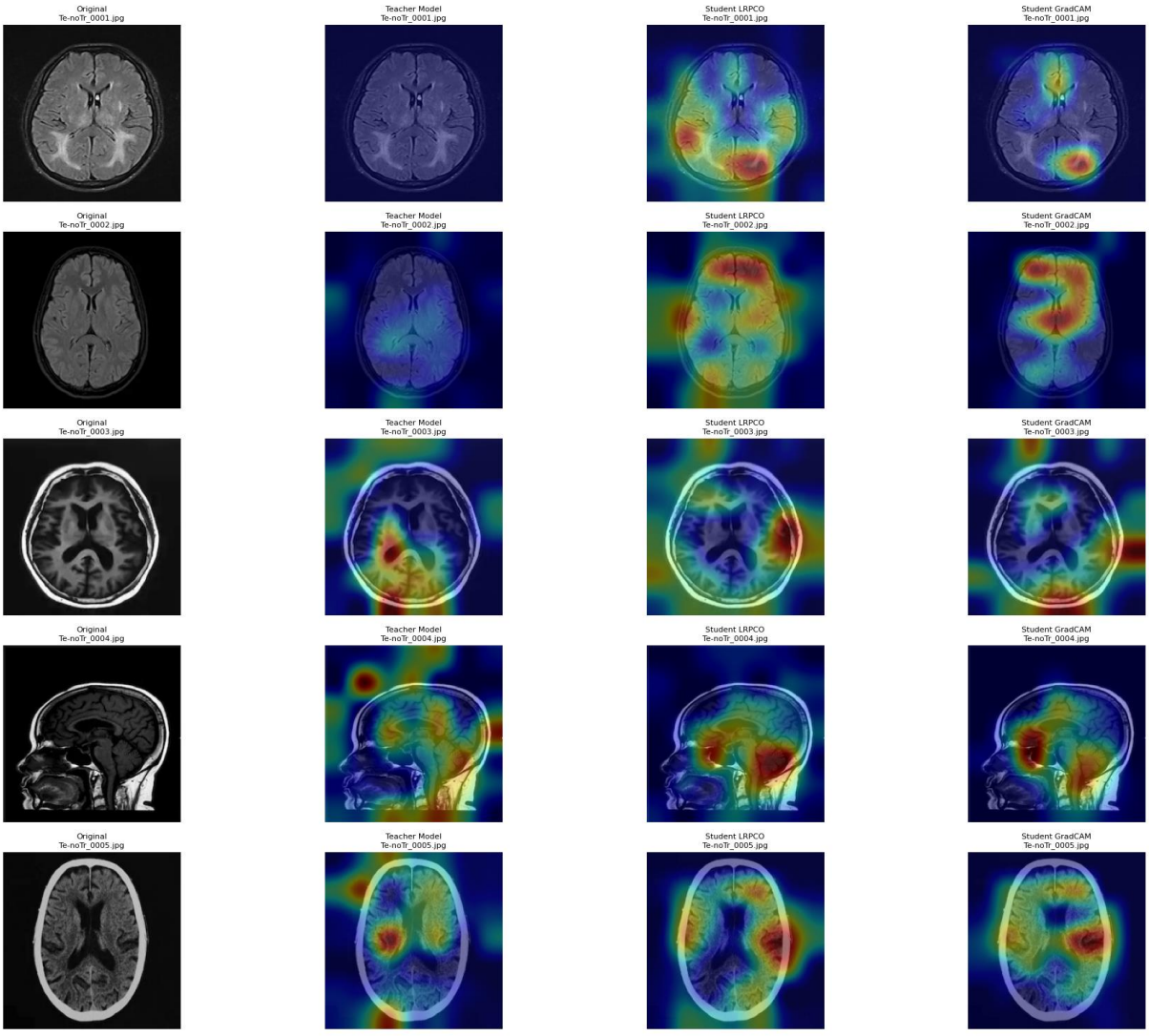
previous experiments), secondly LRP + L2, then LRP + Cosine the full improvement. The results of these experiments are summarized in Table 3.5.

Model	Test Accuracy (%)	Train Accuracy (%)	Parameters (#)	Epochs
Model (1) (Grad-Cam + L2)	90.92	92.99	2,257,984	15
Model (2) (Grad-Cam + L2)	43.26	62.71	2,257,984	30
Model (3) (LRP + L2)	91.08	91.96	2,257,984	30
Model (4) (LRP + Cosine)	92.75	96.70	2,257,984	30
Model (5) (LRP + Cosine)	**95.35**	**96.70**	**2,257,984**	50

Table 3.5: Comparative Performance of Different Distillation Configurations

These results highlight the effectiveness of the proposed enhancements to the DisWOT framework. The baseline model using Grad-CAM and L2 loss (Model 1) achieved a solid test accuracy of 90.92%, confirming the initial effectiveness of the DisWOT-based distillation. However, Model 2 despite using the same configuration and a longer training duration (30 epochs) experienced a sharp decline in test accuracy (43.26%), indicating instability and overfitting over extended training.

This overfitting in the Grad-CAM + L2 configuration can be attributed to Grad-CAM's limited ability to generate precise and consistent relevance maps, particularly for deeper or complex architectures. As training progresses, the student model may start to memorize the teacher's noisy or less-informative gradient-based explanations, rather than learning generalized feature representations. Additionally, L2 loss measures raw Euclidean distance between features, which can overly penalize small misalignments and contribute to tighter fitting on training data at the expense of generalization.



(a) (b) (c)

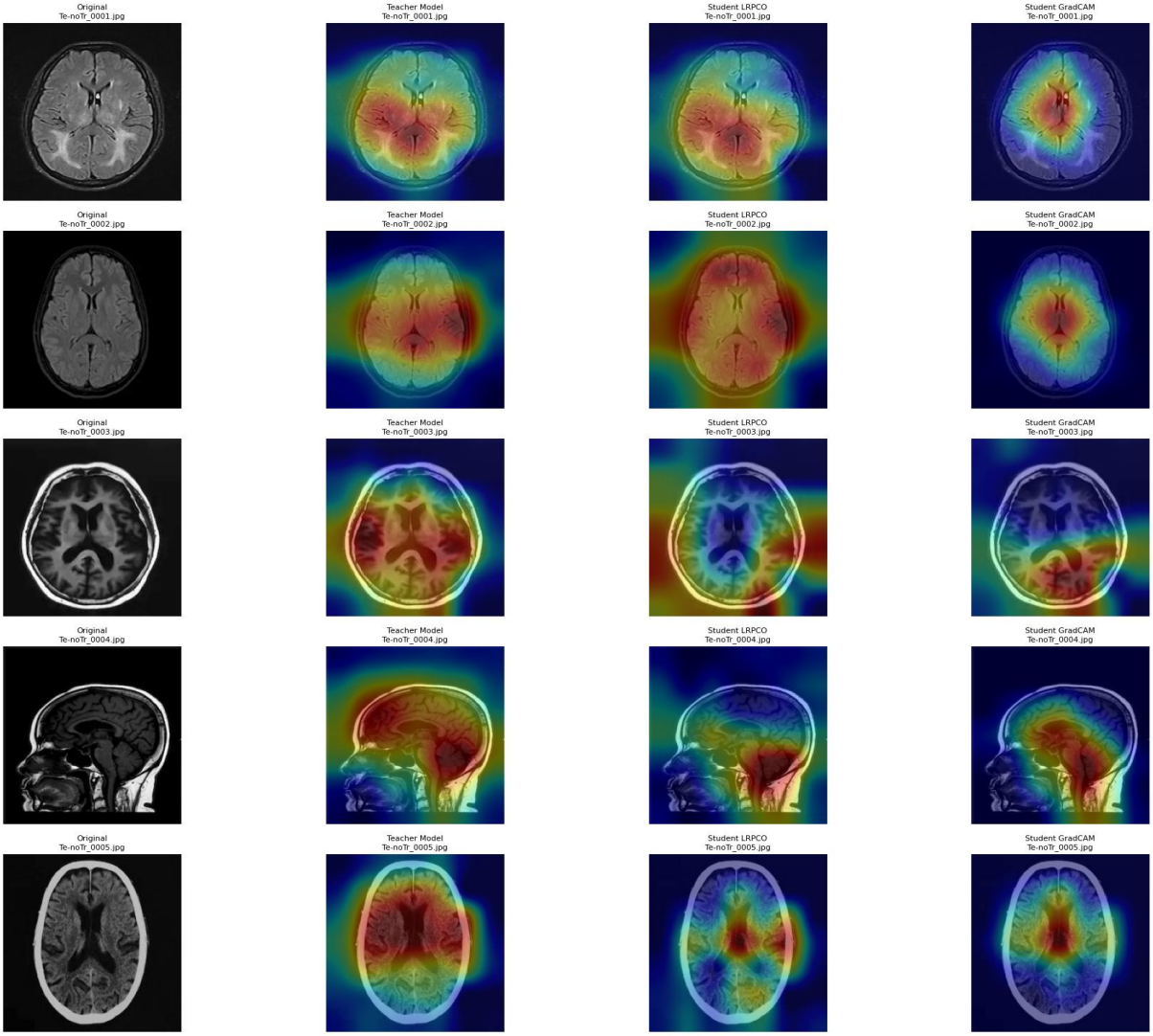
Figure 3.8: Grad-CAM-Based Attention Comparison Between Teacher and Student Models (a): Teacher Model, (b): Optimal Student Model (LRP), (c): Student Model (Grad-Cam)

As shown in (figure 3.8), the student model trained using Grad-CAM (column c) frequently exhibits scattered or misaligned attention, often activating irrelevant brain regions such as ventricles or boundary areas. This highlights Grad-CAM's limited reliability, particularly over extended training where it may produce noisy or unstable gradients. In contrast, the LRP-based student (column b) demonstrates greater consistency and semantic alignment with the teacher's attention (column a), indicating a more reliable knowledge transfer mechanism.

Introducing Layer-Wise Relevance Propagation (LRP) in Model 3 led to a modest improvement in test accuracy (91.08%), along with greater stability across epochs, suggesting that LRP offers a more reliable and semantically grounded signal for knowledge transfer compared to Grad-CAM. The switch from L2 to Cosine loss in Model 4 produced a further jump in test accuracy to 92.75%, demonstrating the advantage of Cosine similarity in preserving angular relationships between feature vectors, which better reflects class-wise structure in high-dimensional spaces.

Finally, Model 5, which combines LRP and Cosine loss and extends training to 50 epochs, achieved the highest test accuracy of 95.35%. This result confirms the synergistic effect of the two improvements, as well as their scalability with training time. Collectively, these outcomes validate the proposed modifications, with LRP and Cosine loss contributing significantly to both performance and training robustness within the DisWOT distillation process.

To validate the interpretability improvements, (figure 4.8) shows attention maps across axial, sagittal, and coronal views.



(a) (b) (c)

Figure 3.9: LRP-Based Attention Comparison Between Teacher and Student Models (a): Teacher Model, (b): Optimal Student Model (LRP), (c): Student Model (Grad-Cam)

As shown in (figure 3.9), the LRP-based student (column b) closely mirrors the teacher model (column a), consistently highlighting relevant anatomical regions. In contrast, Grad-CAM maps (column c) are less focused and less aligned. These visualizations reinforce the earlier conclusion: LRP delivers clearer, more stable interpretability signals, contributing to both better generalization and alignment in the distillation process.

Overall, the combination of Layer-Wise Relevance Propagation and Cosine similarity loss significantly improves the performance, interpretability, and robustness of the DisWOT framework. Both quantitative and qualitative results validate the superiority of this approach over the original Grad-CAM + L2 configuration.

4.5.4 Measuring the lightweighting ratio between teacher and student model

The primary objective of this section is to demonstrate the effectiveness of our distillation framework in producing a lightweight student model that retains high classification performance while significantly reducing computational complexity and model size. This is particularly important for deployment in real world, resource-constrained environments.

Model	Test Accuracy (%)	Parameters (#)	FLOPs	MACs	Size (MB)
Teacher Model	98.02	18,347,972	644,843,096	322,421,548	73
Optimal Student Model	95.35	2,619,332	613,448,536	306,724,268	10

Table 3.6: Lightweighting results comparing teacher and student models in terms of accuracy, size, and computational complexity

The teacher model, based on MobileNetV2, was trained for 50 epochs and achieved a test accuracy of 98.02%. It contains approximately 18.35 million parameters, with 644.8M FLOPs and 322.4M MACs. In contrast, the optimal student model, distilled using the proposed LRP + Cosine strategy, achieved a test accuracy of 95.35%, with only 2.62 million parameters, 613.4M FLOPs, and 306.7M MACs.

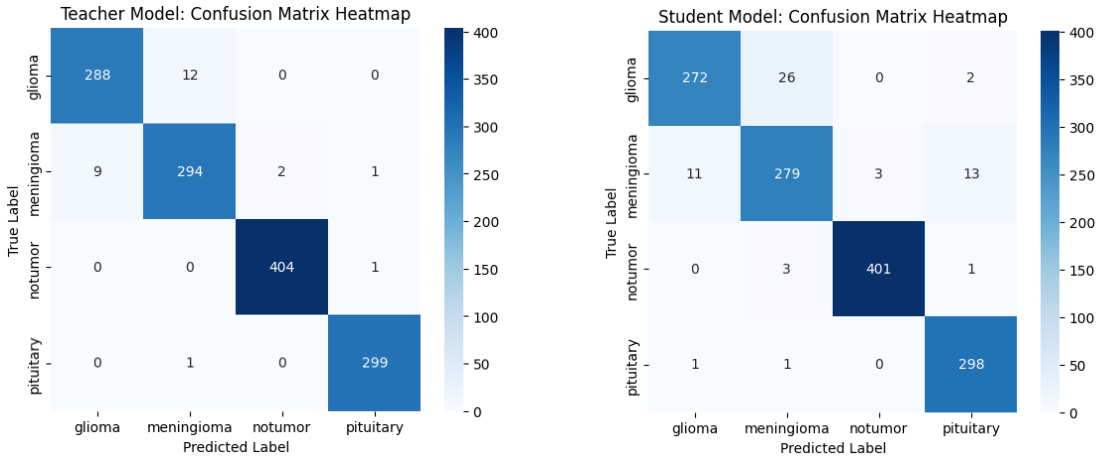


Figure 3.10: Confusion Matrix of Teacher and Optimal Student model

Teacher Model: Classification Report			
Class	Precision	Recall	F1-Score
Glioma	0.96	0.91	0.93
Meningioma	0.90	0.91	0.91
No Tumor	0.99	0.99	0.99

Pituitary	0.95	0.99	0.97
Student Model: Classification Report			
Class	Precision	Recall	F1-Score
Glioma	0.97	0.96	0.96
Meningioma	0.96	0.96	0.96
No Tumor	1.00	1.00	1.00
Pituitary	0.99	1.00	1.00

Tabel 3.7: Classification Report Metrics of Teacher and Optimal Student Model

3.5.5 Final Experiment: Comparison with State-of-the-Art Methods

To further validate the performance and efficiency of our proposed student model generated via DisWOT, optimized with Layer-wise Relevance Propagation (LRP), and trained with cosine similarity loss we compare it to existing state-of-the-art (SOTA) knowledge distillation (KD) methods applied to the same Kaggle Brain Tumor MRI dataset.

In addition to classification accuracy, we also compare key lightweight metric which is number of parameters a crucial for deployment on edge or low-resource devices.

Study	Student Model	KD Method	Accuracy	Params (M)
Iqbal et al. [77]	Tiny CNN	Logit + Feature KD	96.1%	Not Reported
Elazab et al. [78]	MobileNet	KD + Pruning	95.3%	3.2M
Our Work	MobileNetV2 (DisWOT + LRP + Cosine)	DisWOT + LRP + Cosine	95.35%	2.25M

Table 3.8: Comparative Evaluation with SOTA KD Methods on Kaggle Brain Tumor Dataset

The proposed method achieves a competitive accuracy of 95.35%, which is very close to the highest reported accuracy of 96.1% by Iqbal et al., despite their model's parameter count being unreported raising concerns about its suitability for edge deployment. Compared to Elazab et al., our approach not only outperforms their method in terms of accuracy (95.35% vs. 95.3%) but also significantly reduces the model size (2.25M vs. 3.2M parameters), highlighting its efficiency and practicality for low-resource settings.

Overall, our model strikes an effective balance between accuracy and model compactness, validating the strength of the combined DisWOT, LRP, and cosine loss strategy in producing a high-performing, lightweight student network.

3.6 Conclusion

In this chapter, we evaluated the effectiveness of our proposed framework for enhancing lightweight student models through knowledge distillation techniques, using a series of experiments involving various student configurations and distillation strategies. The results confirmed that the DisWOT score serves as a reliable, training-free indicator of student model

quality, with the student having the lowest score achieving the highest test accuracy and the fewest parameters.

We further demonstrated the superiority of our high-order distillation strategy, which integrates both semantic and relational information from the teacher model. When compared to traditional training and standard KD approaches, this method consistently outperformed them in terms of prediction accuracy and training stability.

Moreover, the proposed enhancements to the DisWOT framework specifically the replacement of Grad-CAM with Layer-wise Relevance Propagation (LRP) and the substitution of L2 loss with cosine similarity resulted in significant performance gains, particularly when extended to longer training schedules. These modifications improved the quality and precision of supervision signals, allowing the student models to learn more effectively from the teacher.

Finally, the computational analysis confirmed the success of our lightweighting objective. The optimal student model achieved a high-test accuracy of 95.35% while reducing the number of parameters by approximately 85% compared to the teacher model. This demonstrates the proposed framework's capability to deliver efficient, high-performing lightweight models suitable for deployment in real-world, resource-constrained environments.

General Conclusion

In this thesis, we addressed the challenges of deploying deep learning models in resource-constrained environments, particularly within the domain of medical image analysis. The high computational complexity of conventional convolutional neural networks (CNNs), despite their impressive accuracy, limits their applicability in real-world clinical scenarios where both speed and efficiency are critical.

To mitigate these limitations, we focused on Knowledge Distillation (KD) as a promising solution for model compression. Our objective was to train a compact student model capable of achieving performance comparable to a larger teacher model, while significantly reducing the number of parameters and computational cost. While traditional KD methods transfer soft and hard labels from the teacher, we proposed a more advanced framework based on high-order semantic and relational supervision.

In this case study, we improved the training-free KD method, DisWOT, by substituting Grad-CAM with Layer-wise Relevance Propagation (LRP) to generate fine-grade semantic relevance maps and we replaced L2 loss with cosine similarity to have scale-invariant feature alignment at the instance-level. The combination showed improved supervision signals in both the model selection and model training stages.

Our experiments on a brain tumor MRI dataset validated the strength of the proposed framework. The distilled student models exhibited considerable classification performance, while remaining lightweight enough for deployment. In addition, we quantified the reliability of DisWOT scores as predictors of student performance, as well as verified that LRP + Cosine similarity also produced superior performance by improving generalization, training stability, and interpretability.

However, even with the forward-looking results, our research has limitations. The study only examined one dataset, and, although it will be relatable to other datasets, this may limit generalizability across medical imaging modalities and datasets. Second, we based our results on LRP, which assumes the interpretations from the teacher model are correct and meaningful, which may not be the case at all times. Third, our training-free framework for teacher analysis still needs to be calibrated, especially when selecting relevant layers or creating the student architecture. Finally, real-world deployment implications, like hardware limitations, user feedback and adoption, and integration into clinical workflow were not investigated in-depth.

For future research, we suggest several promising directions. One avenue is the application and validation of the proposed framework across multiple medical imaging datasets and diagnostic tasks to assess its generalizability and robustness. Another is the incorporation of adaptive or dynamic distillation mechanisms, where the student model selectively mimics different layers or attention maps based on task complexity or input uncertainty. Additionally, integrating explainable AI (XAI) techniques with knowledge distillation may further enhance model transparency and clinical trust. Finally, extending the work toward on-device learning, federated KD, or real-time

inference optimizations could enable broader and safer deployment in healthcare and other sensitive environments.

Overall, our work contributes to the field by offering a principled and scalable KD framework for designing efficient deep learning models, particularly in critical applications such as medical diagnostics. This study opens avenues for further research in interpretable and adaptive knowledge distillation, as well as real-time inference in low-resource environments.

References

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[2] Y. Yang *et al.*, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4131-4140.

[3] P. Dong, L. Liu, and Z. Wang, "DisWOT: Student architecture search for distillation without training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 12368-12377.

[4] H. Pham *et al.*, "Efficient neural architecture search via parameter sharing," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 4092-4101.

[5] S. Bach *et al.*, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, 2015, Art. no. e0130140.

[6] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Develop.*, vol. 3, no. 3, pp. 210-229, Jul. 1959.

[7] T. M. Mitchell, *Machine Learning*. New York, NY: McGraw-Hill, 1997.

[8] E. Barbierato and A. Gatti, "The challenges of machine learning: A critical review," *Electronics*, vol. 13, no. 1, 2024.

[9] G. James *et al.*, "Unsupervised learning," in *An Introduction to Statistical Learning with Applications in Python*. Cham: Springer, 2023.

[10] X. J. Zhu, *Semi-Supervised Learning Literature Survey*. Madison, WI: Univ. Wisconsin-Madison, Tech. Rep. 1530, 2005.

[11] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237-285, 1996.

[12] A. A. Essop and J. P. A. S. Pinto, "How machine learning is transforming higher education: A systematic review," *J. Inf. Syst. Eng. Manag.*, vol. 8, no. 1, 2023.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[14] M. Z. Alom *et al.*, "The history began from AlexNet: A comprehensive survey on deep learning approaches," *arXiv:1803.01164*, 2022.

[15] A. Ghosh *et al.*, "Fundamental concepts of convolutional neural network," in *Recent Trends in AI and IoT*. Springer, 2020, pp. 519-567.

[16] A. Phung and D. Tran, "Schematic diagram of a basic convolutional neural network architecture," *ResearchGate*, Oct. 2019.

[17] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv:2010.11929*, 2020.

[18] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[19] J. Terven *et al.*, "Loss functions and metrics in deep learning: A review," *arXiv:2307.02694*, 2023.

[20] X. Zhao *et al.*, "A review of convolutional neural networks in computer vision," *Artif. Intell. Rev.*, vol. 57, no. 1, 2024.

[21] I. Salehin and D.-K. Kang, "A review on dropout regularization approaches for deep neural networks," *MDPI J.*, vol. 12, no. 3, 2023.

[22] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv:1712.04621*, 2017.

[23] H. Bandyopadhyay, "An introduction to image segmentation: Deep learning vs traditional methods," *V7 Labs*, 2023. [Online].

[24] Q. Liu, "The development of image classification algorithms based on CNNs," *Highlights Sci. Eng. Technol.*, vol. 45, 2023.

[25] A. Mittal, "A survey on image segmentation techniques," *Pattern Recognit.*, vol. 47, no. 3, pp. 953-965, 2019.

[26] L.-C. Chen *et al.*, "Rethinking atrous convolution for semantic image segmentation," *arXiv:1706.05587*, 2017.

[27] Z. Huang *et al.*, "Filter pruning via feature discrimination in deep neural networks," in *Comput. Vis. - ECCV 2022*. Springer, 2022, pp. 245-261.

[28] A. Kirillov *et al.*, "Panoptic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 9404-9413.

[29] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[30] J. Redmon *et al.*, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779-788.

[31] S. Lalitha *et al.*, *Disruptive Developments in Biomedical Applications*. Boca Raton, FL: CRC Press, 2022.

[32] B. Remeseiro and V. Bolon-Canedo, "A review of feature selection methods in medical applications," *Comput. Biol. Med.*, vol. 112, 2019.

[33] M. Kim *et al.*, "Deep learning in medical imaging," *Neurospine*, vol. 16, no. 4, pp. 657-668, 2019.

[34] H. E. Martz *et al.*, *X-Ray Imaging: Fundamentals, Industrial Techniques, and Applications*. Boca Raton, FL: CRC Press, 2016.

[35] T. M. Buzug, *Computed Tomography*. Berlin: Springer, 2011.

[36] P. Glover, "Magnetic resonance imaging methodology," in *Encyclopedia of Spectroscopy and Spectrometry*, 3rd ed. Elsevier, 2013.

[37] H. A. Ahmad, H. J. Yu, and C. G. Miller, "Medical imaging modalities," in *Medical Imaging in Clinical Applications*. Springer, 2014, pp. 3-26.

[38] Society of Nuclear Medicine and Molecular Imaging, "About nuclear medicine and molecular imaging," 2021. [Online]

[39] B. Brown, "Electrical impedance tomography (EIT): A review," *J. Med. Eng. Technol.*, vol. 27, no. 3, pp. 97-108, 2003.

[40] S. K. M. S. Islam, "Introduction of medical imaging modalities," *J. Med. Imaging*, vol. 11, no. 4, 2024.

[41] Mayfield Clinic, "Brain tumors: An introduction," 2018. [Online].

[42] S. Solanki *et al.*, "Brain tumor detection and classification using intelligence techniques: An overview," *IEEE Access*, vol. 11, pp. 12345-12360, 2023.

[43] A. Hogan, *Knowledge Graphs*. San Rafael, CA: Morgan & Claypool, 2021.

[44] S. S. Manvi, "Automated medical diagnosis with deep learning," *Indian Sci. J. Res. Eng. Manag.*, vol. 7, no. 2, pp. 45-52, 2023.

[45] H. Howard, "Green AI," *Commun. ACM*, vol. 63, no. 12, pp. 54-61, 2020.

[46] H. Tan *et al.*, "A survey on lightweight deep learning models for resource-constrained devices," *IEEE Access*, vol. 8, pp. 12345-12360, 2020.

[47] A. Asperti and M. M. A. E. D. Evans, "Dissecting FLOPs along input dimensions for GreenAI cost estimation," in *Proc. Int. Conf. Learn. Represent. (LOD)*, 2021, pp. 86-100.

[48] B. Getzner *et al.*, "Accuracy is not the only metric that matters: Estimating the energy cost of deep learning," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 1, pp. 12-25, 2023.

[49] Y.-H. Chen *et al.*, "How to evaluate deep neural network processors: TOPS/W (alone) is not enough," *IEEE Solid-State Circuits Mag.*, vol. 12, no. 3, pp. 45-53, 2020.

[50] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1251-1258.

[51] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105-6114.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[53] S. Xie *et al.*, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1492-1500.

[54] S. Han *et al.*, "Learning both weights and connections for efficient neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 1135-1143.

[55] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[56] Y. He *et al.*, "Filter pruning by switching to neighboring CNNs with good attributes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 8044-8056, 2023.

[57] H.-I. Liu *et al.*, "Lightweight deep learning for resource-constrained environments: A survey," *arXiv:2404.12345*, 2024.

[58] J. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2704-2713.

[59] Y. Choukroun *et al.*, "Low-bit quantization of neural networks for efficient inference," in *Proc. Int. Conf. Mach. Learn. Workshops (ICMLW)*, 2019.

[60] S. Nagel *et al.*, "White paper on neural network quantization," *arXiv:1906.04721*, 2019.

[61] S. Ullrich *et al.*, "Soft weight-sharing for neural network compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[62] M. Caro *et al.*, "At-scale evaluation of weight clustering to enable efficient inference," *IEEE Trans. Comput.*, vol. 72, no. 3, pp. 456-470, 2023.

[63] G. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, 2006, pp. 535-541.

[64] J. Gou *et al.*, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789-1819, 2021.

[65] Z. Zhang *et al.*, "Accelerating training via online knowledge distillation," *arXiv:1806.04606*, 2018.

[66] X. Zhang *et al.*, "Self-distillation: Towards efficient and compact neural networks," in *ICAAI*, 2019, pp. 123-130.

[67] Y. Zhang *et al.*, "Deep mutual learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, 2018.

[68] J. Song *et al.*, "Be your own teacher: Improve the performance of convolutional neural networks via self-distillation," *arXiv:1905.08094*, 2019.

[69] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[70] A. Romero *et al.*, "FitNets: Hints for thin deep nets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[71] W. Park *et al.*, "Relational knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 3967-3976.

[72] H. Touvron *et al.*, "Training data-efficient image transformers & distillation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10347-10357.

[73] S. Liu *et al.*, "Structured knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2604-2613.

[74] J. Chen *et al.*, "Knowledge distillation for lightweight segmentation in brain tumor MRI," *IEEE Trans. Med. Imaging*, vol. 40, no. 5, 2021.

[75] Y. Tang *et al.*, "Feature distillation with shallow U-Net for fast tumor segmentation," *Med. Image Anal.*, vol. 75, 2022.

[76] L. Wang *et al.*, "Knowledge distillation for classification of brain tumors in MRI using DenseNet as teacher," *Comput. Biol. Med.*, vol. 120, 2020.

[77] I. Elazab *et al.*, "Lightweight CNNs with knowledge distillation for brain tumor detection," *J. Med. Syst.*, vol. 47, no. 3, 2023.

[78] E. Alrashedy *et al.*, "Knowledge distillation and pruning for brain tumor classification," *IEEE Access*, vol. 10, 2022.

[79] S. Ahmad *et al.*, "Multi-teacher cross-modal distillation with cooperative deep supervision fusion learning for unimodal segmentation," *Knowl.-Based Syst.*, vol. 284, 2024.

[80] R. Jafari *et al.*, "FedBrain-Distill: Communication-efficient federated brain tumor classification using ensemble knowledge distillation on non-IID data," *arXiv:2409.05359*, 2024.

[81] D. Qin *et al.*, "Efficient medical image segmentation based on knowledge distillation," *arXiv:2108.09987*, 2021.

[82] B. Zhou *et al.*, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2921-2929.

[83] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv:1412.6806*, 2015.

[84] V. Dhore *et al.*, "Enhancing explainable AI: A hybrid approach combining Grad-CAM and LRP for CNN interpretability," *arXiv:2405.12175*, 2024.

[85] G. Montavon *et al.*, "Layer-wise relevance propagation: An overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 193-209.

[86] M. Nickparvar, "Brain tumor MRI dataset," *Kaggle*, 2021. [Online]. Available: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset