People's Democratic Republic of Algeria Ministry of Higher Education and Scientific Research

Kasdi Merbah University – Ouargla

Faculty of New Information and Communication Technologies

Department of Computer Science and Information Technology

Thesis submitted for the Master's degree

Field: Computer Science and Information Technology

Sector: Computer Science

Specialty: industrial computing

Presented by:

Abikeur Meriem Wissem

Gheribi Zineb

Title:

Video Quality Assessment based Deep Meta learning

Supervisor: Dr. Naima Merzougui, Ouargla University

Prsedident: Dr. Ramla Bensaci , Ouargla University

Examiner: Dr. Lamis Hamrouni , Ouargla University

Academic year : 2024/2025

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful.

First and foremost, all praise and thanks are due to Allah, whose boundless mercy and guidance have sustained us through every challenge and made this achievement possible. His blessings have been our strength throughout this journey.

We would like to express our sincere and profound gratitude to our respected supervisor, **Dr. Merzougui Naima**, for her invaluable support, insightful guidance, and continuous encouragement. Her dedication, patience, and trust in us have left a lasting impact, and we are truly honored to have worked under her supervision.

Our deepest thanks also go to the distinguished members of the evaluation committee for their time, attention, and constructive feedback, which have greatly enriched the quality of our work.

We extend our heartfelt appreciation to our families, whose unwavering love, sacrifices, and prayers have been our foundation. To our parents: your belief in us gave us the strength to keep moving forward. To our friends and colleagues: thank you for your companionship, motivation, and encouragement during both the highs and the lows.

This final-year project has been more than an academic requirement—it has been a transformative experience that taught us resilience, collaboration, and the value of perseverance. We are proud of what we have accomplished and thankful to everyone who helped make this journey meaningful.

Dedication

With all love, gratitude, and pride,
I dedicate the fruit of this humble effort to those who have been the light on my path, the support in my struggles, and the motivation behind every step:

To my beloved mother khadidja,

whose prayers have been my refuge and whose patience has driven me forward… Thank you for your boundless kindness, continuous support, and silent sacrifices. No words can truly express what your love means to me.

To my dear father ahmed,

my first role model and honest adviser… Thank you for the confidence you planted in me and your belief in my ambitions. You have always been a true pillar I lean on whenever I stumble.

To my brothers and sisters makarim, mohamed and chawki,

you are my true companions and closest to my heart… Your laughter, words, and support have been a balm in the hardest days. Thank you for always standing by me in my silence and speech, in my weakness and strength.

To my dear fiancé chamso,

partner in the journey and in dreams… Thank you for your patience, support, and for embracing me every moment. Your presence in my life is a blessing I thank God for daily. You have been my safety and the motivation to move forward.

To my extended family,

to every uncle, aunt, cousin… Thank you for your prayers, kind words, and continuous encouragement. You are an extension of my pride and the warmth of my belonging.**And to my dear friends,**

you are the stars that lit my way… Thank you to those who stood by me in moments of despair before joy, thank you to everyone who listened, supported, and shared even a word. Your friendship was a lifeline in hard times and multiplied the joy in moments of success.I dedicate this work to all of you Because each one of you was a reason for me to reach, to endure, to dream, and to achieve.

Abikeur meriem wissem

Dedication

With deep love, enduring gratitude, and heartfelt appreciation,
I dedicate this humble achievement to those whose presence, prayers, and unwavering belief have shaped every step of this journey:

To my precious mother,
your gentle strength and endless patience have been my shelter through life's storms. Your quiet sacrifices and tireless care gave me the courage to continue, even when the path was unclear. I owe so much to your boundless love.

To my beloved father,
the foundation of my integrity and the first to believe in my potential. Your wisdom has guided my choices, and your unwavering faith in me has been a constant source of strength. Thank you for teaching me to stand tall and strive higher.

To my cherished brothers and sisters,
your presence brought warmth to my hardest days and joy to the smallest victories. You have been my source of laughter, encouragement, and quiet reassurance. I am forever grateful for your love and companionship.

To my extended family uncles, aunts, and cousins,
you have always been close in spirit, even from afar. Your kind prayers and sincere encouragement filled my heart with belonging and pride. Thank you for walking alongside me, even when the steps were unseen.

To my dearest friends,
you are the steady lights that brightened the darkest nights. Whether through shared silence or a few heartfelt words, your support carried me through. You celebrated my triumphs and stood firm in my struggles, making every success more meaningful. This work is a reflection of your love, your faith, and your presence.
To each of you I dedicate this with all my heart. You are the reasons I endured, hoped, and reached the finish line.

Zineb gheribi

Abstract

Video Quality Assessment (VQA) is essential to ensure user satisfaction in multimedia systems. Traditional methods, whether subjective or objective, often struggle to generalize across diverse content and distortion types. While advances in deep learning have improved prediction accuracy, many models still face challenges in adapting to unseen scenarios with limited labeled data. This thesis proposes an innovative no-reference video quality assessment approach using the Model-Agnostic Meta-Learning (MAML) algorithm, referred to as VQA-DML (Video Quality Assessment based on Deep Meta-Learning). The objective is to train a regression model capable of predicting subjective quality scores (MOS) by learning from multiple small-scale tasks drawn from various video databases. Experiments were conducted on three well-known datasets: LIVE, KoNViD-1k, and YouTube-UGC, covering a wide range of distortions and content types. The model's performance was evaluated using standard regression metrics (MAE, RMSE, R^2) and correlation coefficients (PLCC, SRCC, KRCC), highlighting its ability to generalize effectively to new videos. This work offers a flexible and adaptive framework for video quality prediction, with promising prospects for real-world multimedia applications.

The proposed VQA-DML model achieved promising results, with Pearson correlation up to 0.6548 and Spearman correlation up to 0.6679, outperforming several traditional NR-VQA models. These results demonstrate the model's ability to generalize effectively to unseen video content with minimal supervision.

Keywords:
Video Quality Assessment, Deep Learning, Meta-Learning, MAML, Regression, MOS, PyTorch.

Résumé

L'évaluation de la qualité vidéo (VQA) est essentielle pour garantir la satisfaction des utilisateurs dans les systèmes multimédias. Les méthodes traditionnelles, qu'elles soient subjectives ou objectives, rencontrent souvent des difficultés à généraliser à travers des contenus et des types de distorsions variés. Les progrès en apprentissage profond permettent une meilleure précision de prédiction, mais de nombreux modèles peinent encore à s'adapter à des situations inédites avec peu de données étiquetées.
Ce mémoire propose une approche innovante d'évaluation de la qualité vidéo sans référence en utilisant l'algorithme de méta-apprentissage MAML (Model-Agnostic Meta-Learning), nommée : VQA-DML (Video Quality Assessment based Deep Meta Learning). L'objectif est d'entraîner un modèle de régression capable de prédire des scores de qualité subjective (MOS) en apprenant à partir de multiples tâches de petite taille issues de différentes bases de données vidéo. Des expérimentations ont été menées sur les trois bases LIVE, KoNViD-1k et YouTube-UGC, couvrant une large diversité de distorsions et de contenus.
Les performances du modèle ont été évaluées à l'aide de métriques standards de régression (MAE, RMSE, R^2) et de coefficients de corrélation (PLCC, SRCC, KRCC), mettant en évidence sa capacité à généraliser efficacement à des nouveaux vidéos. Ce travail apporte ainsi un cadre flexible et adaptatif pour la prédiction de la qualité vidéo, avec des perspectives prometteuses pour des applications multimédias réelles.

Les résultats obtenus montrent que le modèle VQA-DML a atteint des performances prometteuses, avec un coefficient de corrélation de Pearson atteignant 0.6548 et un coefficient de Spearman de 0.6679, surpassant plusieurs modèles NR-VQA traditionnels. Ces résultats confirment la capacité du modèle à se généraliser efficacement à de nouveaux contenus vidéo avec un minimum d'exemples.

Mots-clés:
Évaluation de la qualité vidéo, Deep Learning, Meta-Learning, MAML, Régression, MOS, PyTorch.

ملخص

يعد تقييم جودة الفيديو (VQA) ضروريًا لضمان رضا المستخدمين في أنظمة الوسائط المتعددة. غالبًا ما تُواجه الطرق التقليدية، سواءً كانت ذاتية أو موضوعية، صعوبة في التعميم على مختلف أنواع المحتوى والتشوهات. تُمكّن التطورات في مجال التعلم العميق من تحسين دقة التنبؤ، إلا أن العديد من النماذج لا تزال تواجه صعوبة في التكيف مع المواقف الجديدة ذات البيانات المُصنّفة المحدودة.

تقترح هذه الرسالة نهجًا مبتكرًا لتقييم جودة الفيديو دون مرجع باستخدام خوارزمية التعلم الفوقي المستقل عن النموذج (MAML)، والتي تُسمى) VQA-DML التعلم الفوقي العميق القائم على تقييم جودة الفيديو). الهدف هو تدريب نموذج انحدار قادر على التنبؤ بدرجات الجودة الذاتية (MOS) من خلال التعلم من مهام صغيرة متعددة من قواعد بيانات فيديو مختلفة. أُجريت التجارب على ثلاث قواعد بيانات: LIVE، وKoNViD-1k، وYouTube-UGC، والتي تغطي مجموعة واسعة من التشوهات والمحتوى. تم تقييم أداء النموذج باستخدام مقاييس الانحدار القياسية (MAE ، RMSE، R²) ومعاملات الارتباط (PLCC ، SRCC، KRCC)، مما يُبرز قدرته على التعميم بفعالية على مقاطع الفيديو الجديدة. وبالتالي، يوفر هذا العمل إطارًا مرنًا وقابلًا للتكيف للتنبؤ بجودة الفيديو، مع آفاق واعدة لتطبيقات الوسائط المتعددة في العالم الحقيقي.

أظهرت النتائج أن النموذج المقترح VQA-DML حقق أداءً واعدًا، حيث بلغ معامل الارتباط بيرسون 0.6548 ومعامل سبيرمان 0.6679، متفوقًا على عدة نماذج تقليدية لتقييم جودة الفيديو بدون مرجع. هذه النتائج تؤكد قدرة النموذج على التعميم والتكيف مع مقاطع فيديو جديدة باستخدام عدد محدود من الأمثلة.

الكلمات-المفتاحية:

تقييم جودة الفيديو، التعلم العميق، التعلم الفوقي، MAML، الانحدار، MOS، PyTorch.

Table of Contents

1.1Introduction

2.4 Definition of Deep Meta-Learning

The List of Figure

The List of tables

Abbreviations

VQA :video quality asseessment

MOS : Mean Opinion Score
FR :Full-Reference
NR :No-Reference
RR :Reduced-Reference
DL :Deep Learning
ML :Machine Learning
AI :Artificial Intelligence
MAML : Model-Agnostic Meta-Learning
Meta-VQA : Meta-Learning based VQA

CNN :Convolutional Neural Network
MLP :Multi-Layer Perceptron
GPU :Graphics Processing Unit
I3D :Inflated 3D Convolutional Network
C3D :3D Convolutional Network
RMSE : Root Mean Squared Error
MAE :Mean Absolute Error
SRCC :Spearman’s Rank Correlation Coefficient
PLCC :Pearson’s Linear Correlation Coefficient
CSV :Comma-Separated Values
PyTorch :Python-based Machine Learning Library

General Introduction

In today's digital era, video content dominates modern communication, entertainment, education, and social interaction. From video conferencing to online streaming platforms and surveillance systems, high-quality videos have become essential to user experience and perception. However, various distortions such as compression artifacts, transmission errors, or noise introduced during capturing can degrade video quality, negatively affecting the user's satisfaction. As a result, assessing video quality accurately has become a critical task in

Multimedi processing and communication systems.

Traditionally, video quality assessment (VQA) was carried out using subjective testing, where human viewers rate video content based on perceived quality. While this approach remains the most reliable, it is time-consuming, expensive, and impractical for real-time or large-scale systems. Therefore, the development of objective, automatic, and data-driven methods for video quality prediction has gained significant attention in recent years.

With the rise of machine learning and deep learning technologies, new approaches have emerged that leverage large-scale datasets to model the relationship between video characteristics and perceived quality. However, many existing models struggle to generalize across different types of distortions or unseen content, especially in scenarios where labeled data is scarce. This limitation motivates the use of meta-learning, or "learning to learn," which offers a way to build models that can quickly adapt to new conditions with minimal data.

This graduation project addresses the challenge of video quality prediction using Meta Deep Learning, specifically employing the Model-Agnostic Meta-Learning (MAML) framework. The main objective is to train a model that can evaluate the quality of videos by learning from small subsets of data (few-shot learning) and adapt efficiently to new distortion scenarios. The proposed solution uses numerical video representations (features) and Mean Opinion Scores (MOS) from a widely recognized dataset to perform this assessment.

The structure of this report is organized as follows:

Chapter 1 introduces the concept of video quality, the importance of objective quality metrics, and the types of distortions that affect visual perception.

Chapter 2 presents the theoretical foundations of Meta Learning and Deep Learning, comparing traditional learning paradigms with meta-learning strategies.

Chapter 3 outlines the methodology used to build the video quality assessment model, including data processing, model architecture, training strategy, and evaluation techniques.

Chapter 4 analyzes and discusses the experimental results, performance metrics, and limitations, followed by suggestions for future research directions.

By the end of this study, we aim to demonstrate the effectiveness of meta-learning in video quality prediction and highlight its potential for developing adaptive and robust quality assessment systems.

Chapter 1 : Video quality evaloution

1.1Introduction

In today's digital era, video quality has become a crucial factor in user experience, playing a significant role in various fields such as entertainment, communication, education, and digital broadcasting. As reliance on visual content increases—whether on social media platforms or live streaming services enhancing video quality has become essential to ensure a smooth and enjoyable viewing experience.

Video quality depends on a combination of technical factors that affect image clarity, motion smoothness, and color accuracy. With advancements in imaging and display technologies, video quality is no longer measured solely by resolution but also by enhancements such as High Dynamic Range (HDR), intelligent processing technologies, and efficient compression techniques that reduce data size without losing visual details. These improvements play a crucial role in enhancing the viewing experience, whether on small smartphone screens or large cinema displays.

This chapter will introduse some basic definition and back ground.

1.2Definition of Video Quality

1.2.1Definition of Quality

Quality is the degree to which a product, service, or system meets specific requirements, standards, and customer expectations. It reflects efficiency, reliability, durability, and overall performance in fulfilling its intended purpose.

1.2.2Definition of Video

Video is a sequence of rapidly displayed images (frames) that create the illusion of motion. It is based on the principle of persistence of vision, where individual frames blend together to form a continuous moving image.

.Video quality refers to the clarity and sharpness of visual content displayed on a screen. It is influenced by technical factors such as resolution, frame rate, bitrate, compression, and color accuracy, as well as environmental factors like screen type and lighting conditions.

1.3 Characteristics of Video Quality

Video quality is determined by several key characteristics, including:

1- Resolution :

The number of pixels in a video, such as 720p, 1080p, 4K, and 8K. Higher resolution provides clearer and more detailed images.

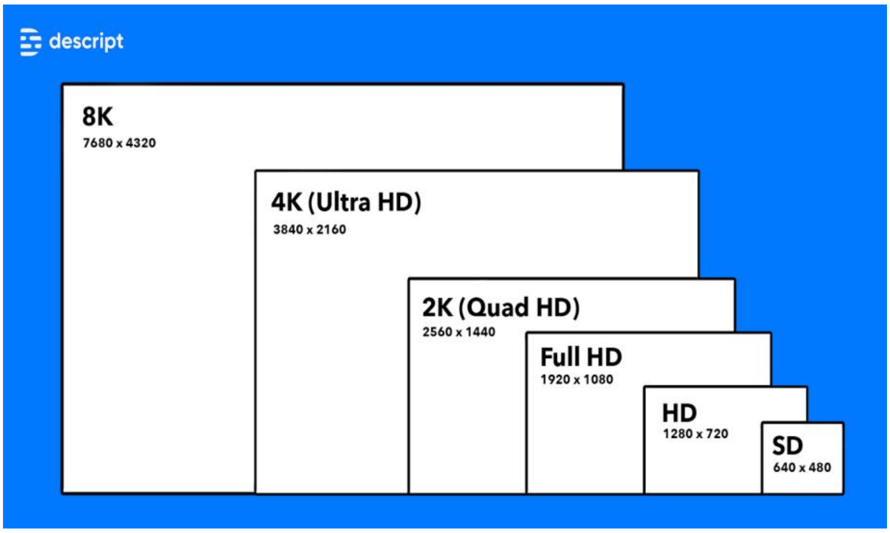


Figure 1 Illustration of Video Resolutions (720p, 1080p, 4K, and 8K)[2]

This figure shows different standard video resolutions, highlighting how increasing resolution enhances image clarity and detail.

2-Frame Rate:

The number of frames displayed per second (fps), such as 24fps, 30fps, or 60fps, which affects motion smoothness.

3-Bitrate:

The amount of data used per second of video. A higher bitrate results in better detail and clarity but requires more storage and bandwidth.

4-Compression & Codec:

Encoding methods like H.264 and H.265 reduce file sizes while maintaining acceptable quality.

5- Aspect Ratio:

The ratio of width to height in a video, such as 16:9 (common for modern screens) or 4:3 (used in older TVs).

Figure 2 4:3 (Standard) aspect ratio size[3]

A depiction of the 4:3 aspect ratio commonly used in traditional video formats, emphasizing its proportions compared to modern widescreen formats.

6- Color Depth:

The number of bits used to represent colors, such as 8-bit, 10-bit, and HDR, which enhance color accuracy.[1]

1.4 Factors Affecting Video Quality

Several factors influence video quality, including:

1- Camera Type and Sensor Quality:

Higher-quality sensors and lenses produce clearer images.

2- Lighting Conditions:

Proper lighting reduces noise and improves detail.

3-Display Type and Quality:

The viewing experience varies based on screen resolution and contrast.

4- Compression Rate:

Excessive compression can lead to quality loss and artifacts.

5- Internet Speed:

Crucial for live streaming, where high-quality video requires stable and fast connectivity.[1]

Table 1. Key Factors Influencing Video Quality and Their Potential Issues

Factor	Impact on Video Quality	Potential Issue if Mismanaged
Video resolution	Higher resolution provides more details	Loss of details at lower resolutions
Frame rete (fps)	Determines motion smoothness	Stuttering motion low
bitrate	Affects clarity and detail visibility	Compression artifacts and pixelation if too low
Compression algorthm	Reduces file sizewhil maintaining quality	Loss of details if compression is too high
Color depth	Imprves color accuracy and gradients	Color banding and distortion if too low
Internet & steaming speed	Determines video resolution in live streaming	Auto quality reduction in weak networks
Lighting during recording	Affects clarity and noise levels	Blurry images and high noise in low ligh
Display quality	Impacts viewing exprience	Differences in color and contrast between screens

1.5 Methods for Evaluating Video Quality:

To assess video quality, several methods are used in research and practical applicat ions. These can be divided into two main categories: objective methods, which use mathematical criteria to measure quality, and subjective methods, which rely on human opinion.

1.5.1-Objective Metrics

1.5.1.1-Definition:

Objective metrics are methods that use computational or mathematical criteria to assess video or image quality without any human intervention. These metrics rely on comparing the original video or image to the modified or distorted version based on a set of numerical criteria that attempt to represent the differences.[4]

1.5.1.2-Basic Concept:

Objective metrics rely solely on numerical calculations and use mathematical techniques to measure the differences between the original and the modified video or image. There is no human interpretive process involved; everything depends on the computations performed by algorithms. These metrics are
fast and efficient but may not always reflect how humans perceive the quality of the video or image.[5]

1.5.1.3-Examples of Objective Metrics:

1-PSNR (Peak Signal-to-Noise Ratio): As explained earlier, this metric relies on comparing the difference between the original and distorted image using mathematical calculations of the mean squared error.[6]

2-SSIM (Structural Similarity Index): It focuses on assessing the structural similarity between the original and distorted images.[7]

3-VMAF (Video Multi-method Assessment Fusion): A metric that combines several metrics, including PSNR and SSIM, to evaluate the overall quality of the video using machine learning.[7]

1.5.1.4-Characteristics:

- Speed: Calculations are performed quickly and accurately.
- Independence from humans: It does not require human intervention in the evaluation process.

1.5.1.5-Types:

Completely relies on mathematical computations and numerical standards.

1.5.1.6-L imitations:

Although mathematically accurate, it may not fully reflect how humans perceive quality, as it does not account for human responses or subtle differences that might be noticeable when watching the video.[4]

1.5.2-Subjective metrics

1.5.2-1-Definition:

Subjective metrics are methods that depend on human opinions to assess video or image quality. Instead of using mathematical calculations, these metrics rely on how viewers or reviewers respond to the visuals they are shown. These evaluations are conducted in controlled environments where participants are asked to watch the video or image and provide feedback on its quality.[8]

1.5.2-2-Basic Concept:

Subjective metrics rely on human opinions about the quality of a video or image. Instead of numerical calculations, this metric reflects how individuals perceive the quality based on their visual experience. This differs from objective metrics, as it reflects human feedback when watching the video or image, not just digital computations.[8]

1.5.2-3-Examples of Subjective Metrics:

1-MOS (Mean Opinion Score): A group of individuals is asked to rate the video or image on a scale of 1 to 5 (or 1 to 10). This is the most commonly used metric for evaluating video or image quality from a human perspective. .[8]

2-DMOS (Differential Mean Opinion Score): This is used to compare two versions of the same video (such as the original version and the distorted version). Viewers rate the difference between the two based on their personal impressions.[9]

3-A/B Testing: Viewers are asked to choose between two different versions of a video based on which one they think has better quality.[10]

1.5.2-4-Characteristics:

1-Perceptual accuracy: It measures how humans perceive the video or image quality, which means it includes all the visual effects that might influence human vision.

2-Real-world experience: Since it depends on people's feedback, it better reflects how quality will impact real viewers.

3-Cost and time: It takes more time and effort compared to objective metrics, as tests must be conducted with a group of people.

4-Human response: It accounts for effects like lighting, colors, contrast, brightness, and other factors that might not be visible to computational metrics.[8]

1.6 Types of Objective VQA

Objective video quality assessment methods are classified into three broad categories, depending on the amount of reference information (the original video) they use:

1. Full Reference (FR)

. The original (reference) video is fully available.

. The algorithm compares every pixel or feature of the degraded video with the reference video.

. Highly accurate, but requires access to the original video.

2. Reduced Reference (RR)

. The reference video is partially available (only features, statistics, or a subset of data).

. The assessment is performed by comparing the test video with this partial information.

. A trade-off between accuracy and transmission/storage efficiency.

3. No Reference (NR) (Also known as Blind VQA – Blind Video Quality Assessment)

. No access to the original video.

. The assessment relies on analyzing the degraded video alone.

. Particularly useful for UGC (User Generated Content) and real-world scenarios.

1.7 Challenges Facing Video Quality

Despite major advancements in video capture, compression, and display technologies, several challenges continue to affect the quality of video content and the overall user experience. These challenges span technical limitations, hardware diversity, network constraints, and content creation variables. Below are the most critical and persistent challenges:

1. Video Compression and Quality Loss
Modern compression codecs such as H.264, H.265 (HEVC), and AV1 are essential for reducing video file size and enabling efficient streaming. However, aggressive compression often leads to visible artifacts, motion blurring, and loss of fine details—especially in fast-moving or high-detail scenes.[11][12].

2. Bandwidth and Data Consumption
High-resolution videos, such as 4K and 8K, require significant bandwidth. Limited or fluctuating internet connectivity can lead to quality drops, stalling, or playback failure. This is a major challenge in rural
areas and emerging markets.[13][14].

3. Storage and Processing Limitations
High-quality videos require massive storage and high processing power. This poses limitations for low-spec devices and back-end servers responsible for encoding, decoding, and delivering content.[15]

4. Latency and Buffering Issues
Latency and buffering are especially problematic in real-time applications like video conferencing or live streaming. Any delay in data transmission can lead to poor synchronization between video and audio
streams.[16]

5. Device and Display Compatibility

Video quality perception varies based on screen resolution, display technology (LCD vs. OLED), color calibration, and hardware decoding capabilities. A video optimized for one device may appear degraded on another.[17]

6. Lighting and Filming Conditions
Video content recorded in low-light or high-contrast conditions may suffer from excessive noise, reduced color depth, or detail loss. This challenge persists even with modern camera sensors.[18]

7. AI-Based Video Enhancement Challenges

AI-driven upscaling, denoising, or restoration techniques can improve quality, but may also introduce unnatural visual effects, hallucinated content, or high computational demands.[19][20]

8. Compatibility with Quality Standards

There are multiple quality standards like SDR, HDR10, HLG, and Dolby Vision. Ensuring compatibility across all platforms requires software and hardware support, which is often lacking or inconsistent.[21][22]

9. Environmental and External Interference
External factors like wind, vibration, or poor audio conditions during outdoor recordings can reduce perceived video quality and require post-processing or stabilization.[23]

1.8 Related work in video quality assessment

8.1 Evolution of Traditional NR-VQA Methods (Based on Handcrafted Features)

In the early development of No-Reference Video Quality Assessment (NR-VQA), research primarily relied on handcrafted features, which were derived from the statistical properties of natural scenes. These methods aimed to simulate the human visual system by extracting distortion-sensitive features, without the need for a reference (undistorted) video.

One of the foundational models in this area is **V-BLIINDS** (Video Blind Image Integrity Notator using DCT Statistics), proposed by Saad et al. (2014) [57]. This method utilizes natural scene statistics (NSS) and motion coherency features extracted from Discrete Cosine Transform (DCT) coefficients of temporal frame differences. The features are then used to train a Support Vector Regressor (SVR) to predict Perceptual video quality.
Another widely adopted method is **BRISQUE** (Blind/Referenceless Image Spatial Quality Evaluator), introduced by Mittal et al. (2012) [58]. Although originally developed for image quality assessment, BRISQUE calculates statistical deviations from natural image models using Mean Subtracted Contrast Normalized (MSCN) coefficients. Due to its simplicity and effectiveness, it has been adapted for assessing frame-wise video quality.

Following BRISQUE, **NIQE** (Natural Image Quality Evaluator) was proposed as an unsupervised alternative by the same authors in 2013 [59]. NIQE constructs a multivariate Gaussian model from a corpus of high-quality natural images and computes quality scores by measuring deviations from this model. Unlike BRISQUE, NIQE does not require training on subjective quality scores.

While these handcrafted approaches offered benefits such as computational efficiency and interpretability, they were limited by their dependence on pre-defined feature sets. They often failed to generalize well to complex, dynamic, or previously unseen distortion types—especially in real-world or user-generated video content. These limitations laid the foundation for the emergence of deep learning-based methods that could learn more abstract and generalized features from large-scale data.

8.2 Deep Learning-Based NR-VQA

In response to the limitations of handcrafted feature-based methods, researchers have increasingly adopted deep learning for No-Reference Video Quality Assessment (NR-VQA). Deep learning models offer the ability to learn rich and abstract features directly from raw video data, without requiring manual feature design. This has led to significant improvements in generalization and prediction accuracy, especially for

unseen distortions or diverse content types.

One of the pioneering works in this area is the study by **Yang et al. (2018)** [60], who proposed a 3D Convolutional Neural Network (3D-CNN) for end-to-end quality prediction in 360-degree virtual reality (VR) videos. The model is capable of extracting both spatial and temporal features simultaneously, which is essential for understanding perceptual quality in immersive environments. The authors combined local patch features through video projection-based fusion

to obtain a global quality score.

In a follow-up study, **Yang et al. (2020)** [61] introduced a non-local **spherical CNN**, which effectively captures spatial-temporal dependencies in panoramic video content. By employing non-local blocks, the model captures long-range correlations that standard CNNs may overlook, enhancing the perceptual quality prediction.

Xu et al. (2020) [62] developed a graph-based method using Graph Convolutional Networks (GCNs), where the video is represented as a graph of visual regions. The model includes a viewport detector that selects visually salient regions based on human viewing behavior, followed by a descriptor network and spatial dependency modeling through graph convolution. This eliminates the need for intermediate image reconstruction, improving both speed and robustness.

An et al. (2022) [63] proposed a hybrid model that combines 2D and 3D CNNs to jointly extract spatial and temporal features. The video is split into spatio-temporal patches, each passed through a sequence of convolutional, excitation, pooling, and fully connected layers to estimate quality scores. This architecture benefits from both fine-grained texture analysis and motion dynamics modeling.

Another important contribution comes from **Saad et al. (2014)** [57], who proposed a blind video quality predictor that leverages natural video statistics. Their model is trained on a database of natural video content and uses motion coherency and scene statistics to predict quality without reference data.

Tu et al. (2021) [64] introduced **UGC-VQA**, a benchmark dataset and deep learning framework for evaluating the quality of user-generated content (UGC). This model emphasizes content diversity, complex artifacts, and authentic distortion types typical in real-world videos. It also highlights the limitations of existing NR-VQA models on such data.

Overall, these deep learning approaches represent a major step forward in VQA research. They allow models to generalize across content types and distortion conditions, adapt to modern video formats (like panoramic and UGC), and scale with large training datasets. However, most still require significant labeled data for training, and their performance can degrade on new domains a challenge addressed in later chapters via meta-learning techniques.

In recent years, research in No-Reference Video Quality Assessment (NR-VQA) has witnessed notable advancements, particularly with the integration of transformer-based architectures and self-supervised learning. For example, Liu et al. (2024) proposed a spatio-temporal transformer model trained on large-scale video data to capture long-range dependencies across frames, improving prediction accuracy in diverse content scenarios. Additionally, Chen and Zhou (2025) introduced a contrastive pretraining framework for NR-VQA, enabling the model to learn robust quality representations from unlabeled video data, which significantly reduced the reliance on large annotated datasets. Another notable work by Tan et al. (2025) utilized a multi-scale attention mechanism to dynamically focus on perceptually salient regions, achieving state-of-the-art results on KoNViD-1k and YouTube-UGC. These recent efforts highlight a growing shift toward hybrid models that combine deep learning, attention mechanisms, and meta-learning concepts to improve generalization in real-world video quality prediction.

1.9 Conclusion

Video quality is influenced by various technical factors, and balancing resolution, frame rate, bitrate, and compression is key to achieving the best viewing experience. Thanks to AI-based improvements and machine learning techniques, video quality assessment can be optimized and predicted. In this context, the next chapter will focus a specific machine learning approach: deep meta-learning.

Chapter 2 : Depp meta learning

2.1 Introduction

In our daily lives, we often find ourselves learning faster when we've had similar experiences before. For example, someone who has learned to ride a bicycle will usually pick up motorcycle riding much more easily. This ability to "learn how to learn" is a unique human

trait and it's the very idea behind meta-learning.
Unlike humans, machines typically need vast amounts of data and repeated training to understand and perform tasks. They don't "get the idea" quickly. This is where deep meta-learning comes in — an approach that aims to give machines a human-like ability to learn quickly from a few examples, by leveraging what they've already learned from many past experiences.

2.2 Definition of Deep Learning :

Deep Learning is a subfield of Machine Learning that focuses on using artificial neural networks with many layers—known as deep neural networks—to automatically learn representations from raw data. These networks are capable of learning hierarchical features, where each layer transforms the input into increasingly abstract and useful representations. Deep Learning has been particularly successful in tasks like image classification, speech recognition, and natural language processing.[24]

2.3 History of Deep Learning

- The foundations of deep learning go back to the late 1950s when Frank Rosenblatt introduced

the Perceptron, an early model of an artificial neuron. However, interest declined after Minsky and Papert (1969) highlighted its limitations. In 1986, Rumelhart, Hinton, and Williams revived neural networks by introducing the backpropagation algorithm, enabling multi-layer networks
to be trained more effectively.

- In 2006, Geoffrey Hinton and colleagues initiated a modern resurgence in deep learning by proposing unsupervised pre-training for deep belief networks, solving many optimization issues in deep models. The field reached a turning point in 2012 when AlexNet, a deep convolutional neural network, achieved a dramatic improvement in image classification accuracy in the ImageNet competition. This success triggered a wave of research and applications across variousdomains.

- Since then, deep learning has advanced rapidly with the introduction of new architectures like ResNet (for image recognition), RNNs and LSTMs (for sequential data), and Transformers (for language modeling and beyond), leading to powerful systems such as GPT, BERT, and DALL·E.[24][25]

2.4 Definition of Deep Meta-Learning :

Deep meta-learning is a branch of machine learning that combines:

Deep learning, which focuses on extracting high-level representations from data using neural networks.

Meta-learning, which aims to acquire the ability to quickly adapt to new tasks by learning how to learn from a variety of tasks.

In other words, a model trained with deep meta-learning does not simply learn how to perform a specific task, but rather learns how to learn from prior experiences so that it can adapt efficiently to new, unseen tasks with minimal data.[26]

2.5 Difference Between Deep Learning and Meta-Learning :

The key distinction between deep learning and meta-learning lies in their goals and adaptability. While deep learning aims to solve a single task using large datasets and fixed model parameters, meta-learning focuses on learning how to quickly adapt to new tasks using only a few examples. The following table summarizes the main differences:

Table 2 : differene between depp learning and meta learning[27]

Aspect	Deep Learning	Meta-Learning
Goal	learn to solve a specific task	Learn how to learn across multiple tasks
Data requirement	Requires a large amount of labeled data	Can adapt with very few examples per task (few-shot)
Generalization	limited to similar data distributions	High adaptability to new, unseen tasks
Learning speed	slow;requires extensive training	Fast; adapts quickly to new tasks
Reusability	One model per task	single model adaptable to many tasks

2.6 Techniques and Models in Deep Meta-Learning

Deep meta-learning is an emerging field in artificial intelligence that aims to empower models with the ability to learn how to learn, especially when facing limited data. The primary approaches in this field are generally categorized into three families:

1- Optimization-Based Meta-Learning
2- Metric-Based Meta-Learning
3- Memory-Based Meta-Learning

2.6.1 Optimization-Based Meta-Learning

This category focuses on learning either an effective initialization or a specialized optimizer that enables a model to rapidly adapt to new tasks using only a few gradient descent steps.

2.6.1.1 MAML (Model-Agnostic Meta-Learning)

2.6.1.1.1 **description**:

To learn an initial set of parameters (θ) such that the model can achieve good performance on a new task after only a few gradient updates.

2.6.1.1.2 **How it works**:

Sample a diverse set of tasks.

2.6.1.1.3 For each task:

Use a small support set to perform task-specific adaptation (inner loop).
Evaluate on a query set for that task.
Update the model’s initial parameters across tasks (outer loop) based on these evaluations.

2.6.1.1.4 Advantages:

Model-agnostic and flexible.
Allows rapid adaptation with minimal data.

2.6.1.1.5 **Disadvantages**:

Computationally expensive due to second-order derivatives.
Sensitive to the configuration of the inner loop and number of adaptation steps.[28}

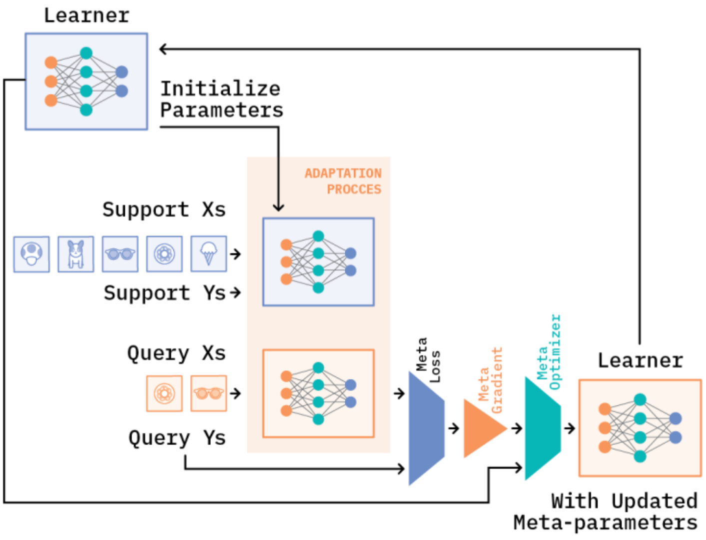


Figure 3 MAML Architecture [29]

A structural overview of the Model-Agnostic Meta-Learning (MAML) framework, showing how it enables quick adaptation to new tasks by learning a good initialization.

2.6.1.2 Reptile

2.6.1.2.1 Description:

A simplified version of MAML that avoids computing second-order gradients.

2.6.1.2.2 How it works:

Train the model on a single task for a few steps.

-Move the meta-parameters slightly toward the task-specific parameters.

2.6.1.2.3 Advantages:

Faster and easier to implement than MAML.

No need for second-order derivative calculations.

2.6.1.2.4 Dis advantages:

Less flexible and potentially less accurate than MAML.[30]

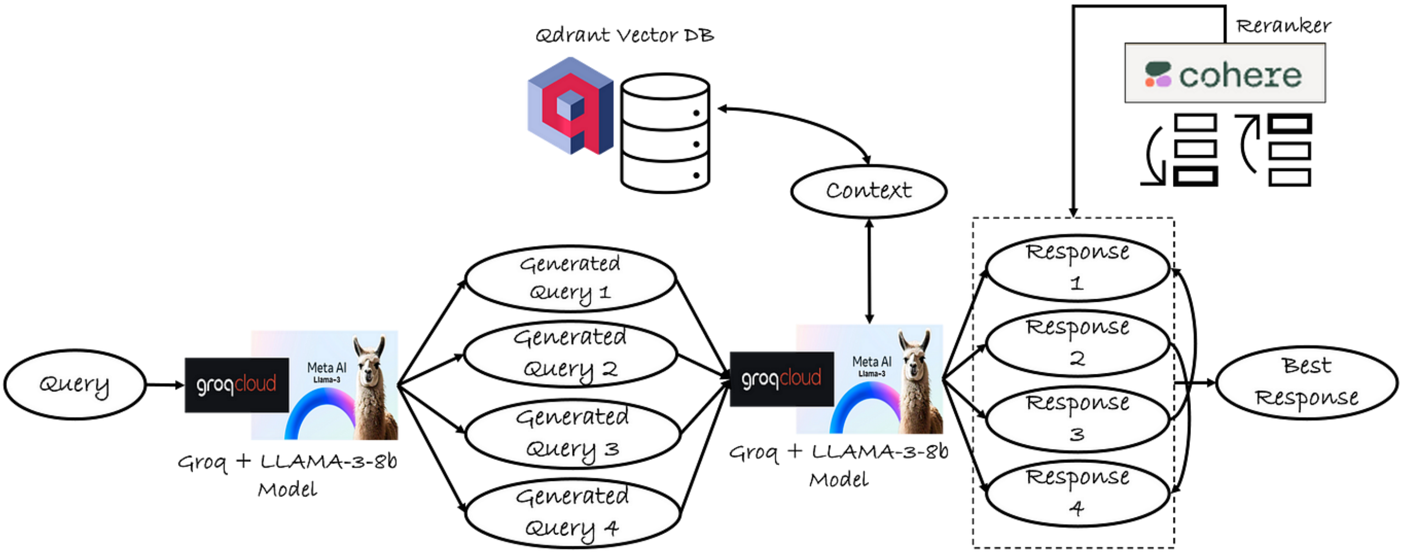


Figure 4 Advanced RAG with Evaluation and Re-ranking[31]

This diagram explains how a meta-learner operates in two stages: initialization for cross-task generalization and adaptation for task-specific learning in meta-space and learner space.

2.6.1.3 Meta-SGD

2.6.1.3.1 Description:

An extension of MAML where, in addition to learning the initial parameters, the model also learns an individual learning rate for each parameter.

2.6.1.3.2 Advantages:

Provides greater flexibility.
May lead to faster convergence during adaptation.

2.6.1.3.3 Disadvantages:

Increases the number of parameters to learn, leading to higher computational complexity.[27]

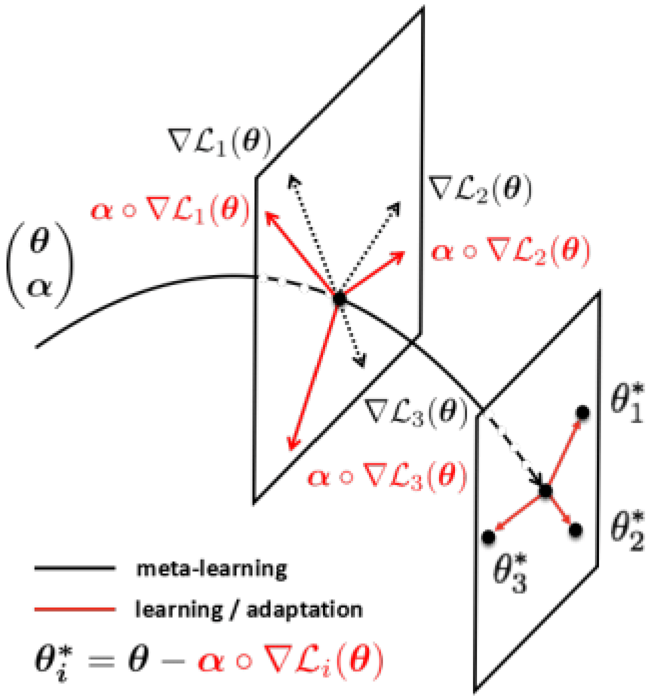


Figure 5 : Illustrating the Two Stages of a Meta-Learner [32]

This diagram explains how a meta-learner operates in two stages: initialization for cross-task generalization and adaptation for task-specific learning in meta-space and learner space.

2.6.2 Metric-Based Meta-Learning

Metric-based meta-learning methods focus on learning an embedding space where similar inputs are placed closer together. The idea is that once this space is learned, new tasks can be solved by comparing examples based on their distances in this space, rather than learning new parameters.[33]

2.6.2.1 Matching Networks

2.6.2.1.1 Description:

Uses a non-parametric approach that relies on comparing embedded representations of support and query samples using a learned similarity function.

2.6.2.1.2 How It Works:

Each example is passed through an embedding function.

A soft attention mechanism is used to compare query samples with the support set.

Predictions are made by weighted voting from the near est neighbors.

2.6.2.1.3 Advantages:

Effective with very few examples.

Fast inference since no fine-tuning is needed during test time.

2.6.2.1.4 Disadvantages:

Performance is sensitive to the embedding quality.

Requires episodic training for effectiveness.[34]

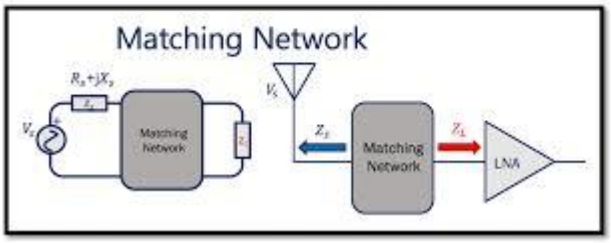


Figure 6 : A Diagram about Machine Network [35]

A general schematic representing components and connections within a machine learning network.

2.6.2.2 Prototypical Networks

2.6.2.2.1 Description:

Learns a metric space where each class is represented by the mean (prototype) of its support examples.

2.6.2.2.2 How It Works:

Compute the mean vector (prototype) of embedded support samples for each class.
Query samples are classified based on their distances to each prototype in the embedding space.

2.6.2.2.3 Advantages:

Simple and effective.
Scalability to many classes due to fixed-size representations.

2.6.2.2.4 Disadvantages:

Assumes data clusters around class means, which may not always hold.
Less effective for complex or overlapping distributions.[36]

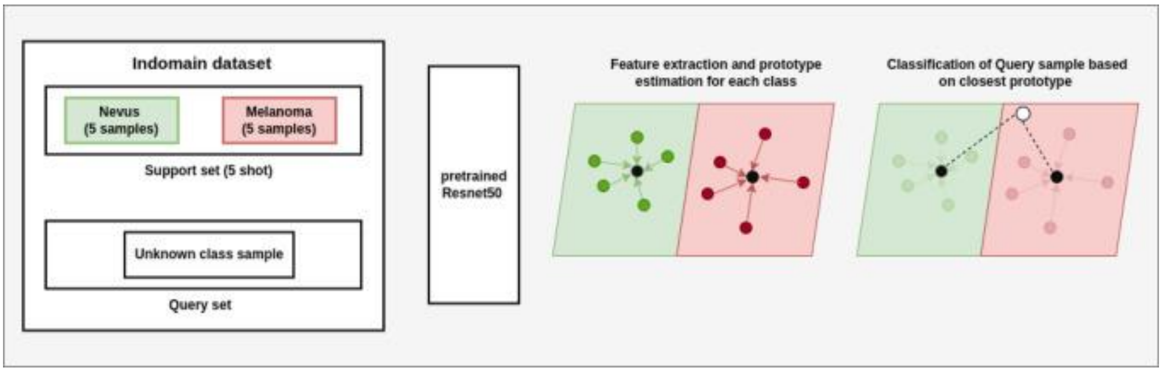


Figure 7 Schematic Demonstrating the Approach of Prototypical Networks [37]

Illustrates how support set samples are encoded using a ResNet50 model to form class prototypes, which are then used to classify query samples by proximity in the feature space.

2.6.2.3 Relation Networks

2.6.2.3.1 Description:

Learns a deep, non-linear distance metric to relate support and query samples.

2.6.2.3.2 How It Works:

-Both support and query samples are passed through a feature extractor.
A relation module compares the extracted features and outputs similarity scores used for classification.

2.6.2.3.3 Advantages:

Learns a more expressive and adaptive similarity function.
Works well in complex scenarios.

2.6.2.3.4 Disadvantages:

More complex architecture compared to prototypical networks.
Requires careful tuning and training.[38]

2.6.3 Memory-Based Meta-Learning

Memory-based meta-learning methods enhance a model's ability to remember and retrieve useful information from past experiences. These approaches are particularly inspired by human memory systems
and are especially powerful when rapid task-switching is required.

2.6.3.1 Neural Turing Machines (NTMs)

2.6.3.1.1 Description:

Combines a neural network controller with an external memory bank, allowing the model to read and write to memory using attention mechanisms.

2.6.2.3.2 How It Works:

The controller (e.g., an RNN) interacts with a memory matrix.

It learns to store and retrieve relevant information across tasks via differentiable operations.

2.6.2.3.3 Advantages:

High flexibility and expressivity.
Capable of handling long-term dependencies.

2.6.2.3.4 Disadvantages:

Training is complex and computationally heavy.
Susceptible to instability.[39]

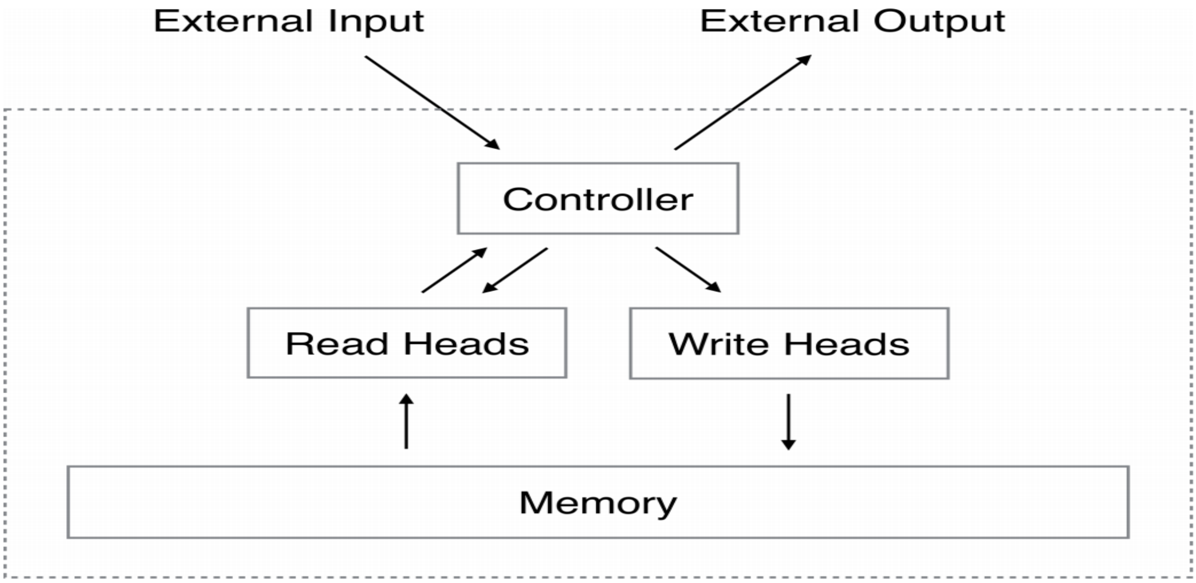


Figure 8 : The NTM's Controller Network and Memory Access [40]

A diagram detailing the Neural Turing Machine (NTM), showing the controller's interaction with inputs and external memory via read and write heads.

2.6.3.2 Memory-Augmented Neural Networks (MANNs)

-2.6.3.2.1 Description:

A simplified and more efficient version of NTMs, designed specifically for meta-learning tasks such as few-shot classification.

2.6.3.2.2 How It Works:

Uses a memory module where previous task examples are stored.
A controller learns to access relevant past experiences to solve current tasks.

2.6.3.2.3 Advantages:

Efficient and effective in few-shot learning.
Supports rapid task adaptation.

2.6.3.2.4 Disadvantages:

Still requires careful memory management.
Prone to overfitting if memory is not used properly.[41]

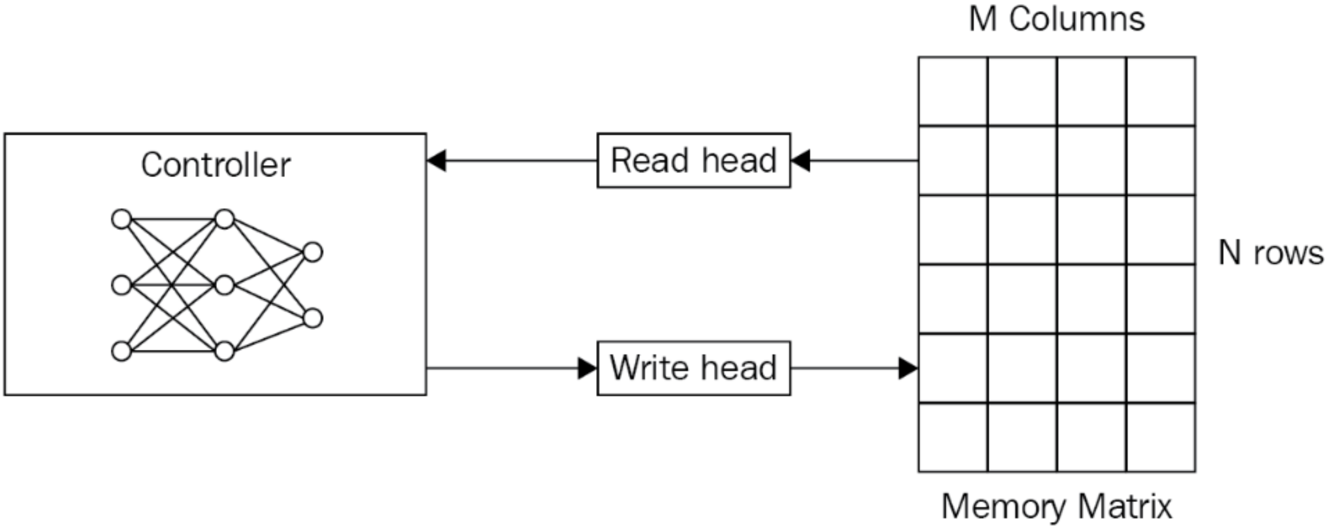


Figure 9 MANN Architecture The controller connects with the memory module through the read and write heads. Each cell in the memory matrix consists of patterns, relationships, and context [42]

2.6.3.3 SNAIL (Simple Neural Attentive Meta-Learner)

2.6.3.3.1 Description:

Combines temporal convolutions with soft attention to efficiently store and retrieve past knowledge.

2.6.3.3.2 How It Works:

Uses temporal convolutions to aggregate past information across time.
Applies attention mechanisms to select relevant information for the current decision.

2.6.3.3.3 Advantages:

High performance in few-shot classification and reinforcement learning.
Simpler training than NTMs.

2.6.3.3.4 Disadvantages:

More complex than purely metric-based models.
Still computationally heavier than simpler architectures.[43]

2.7 Applications of Deep Meta-Learning

Deep meta-learning has demonstrated its effectiveness across a wide range of domains especially those requiring fast adaptation or suffering from limited data availability. Its strength lies in enabling models to learn how to learn, empowering them to solve new tasks with only a few examples, or even none.

2.7.1 Few-Shot and Zero-Shot Learning

One of the most prominent applications of deep meta-learning is in few-shot and zero-shot learning, where models are required to generalize using only a handful—or no—labeled examples for new categories.
-Example: Classifying images from unseen classes using only 1–2 samples per class.
-Common Models: Prototypical Networks, Matching Networks, MAML. [44]

2.7.2 Healthcare and Medical Imaging

In sensitive domains like healthcare, meta-learning helps create models that quickly adapt to individual patient data without requiring large, labeled datasets.
-Use Cases: Predicting disease progression, detecting tumors, handling rare medical cases.
-Approaches Used: MAML, memory-augmented neural networks.[45]

2.7.3 Robotics and Reinforcement Learning

Deep meta-learning supports fast and flexible adaptation in robotics and control environments, where agents must often learn new tasks or behaviors quickly.
-Example: A robotic arm learns to manipulate new objects without full retraining.
-Models Used: MAML, Reptile, RL^2.[28][46]

2.7.4 Natural Language Processing (NLP)

Meta-learning enhances performance in low-resource NLP tasks, allowing models to adapt to new languages, dialects, or tasks with minimal supervision.
-Example: Training a dialogue system to understand a regional dialect with a few training examples.
-Techniques: Metric-based learning, memory networks, meta-learned Transformers.[47]

2.7.5 Video Quality Assessment and Computer Vision

In video analysis, deep meta-learning supports the development of models capable of generalizing across video types and content, especially for predicting visual quality with limited annotated samples.
-Example: Estimating quality scores for new video content with only a few labeled videos.
-Approaches: MAML for regression, prototypical regression networks.[48]

2.7.6 Additional Domains

-Education: Personalized learning systems that adapt to individual student styles.
-Finance: Detecting new fraud patterns from a limited number of examples.
-Speech Recognition: Rapid adaptation to new voices and accents.

2.8 Challenges in Deep Meta-Learning

Despite its promise, deep meta-learning still faces several significant challenges—both theoretical and practical. These challenges must be addressed to fully realize its potential in real-world applications.

2.8.1 Generalization to Unseen Tasks

One of the core promises of meta-learning is the ability to generalize to new tasks. However, in practice, meta-learners often struggle when faced with tasks that differ significantly from those seen during training.
-Issue: Overfitting to the meta-training tasks.
-Implication: The model may fail to adapt properly to truly novel domains or task structures.[27]

2.8.2 Computational Complexity

Many meta-learning algorithms, especially gradient-based ones like MAML, involve computing higher-order gradients or nested optimization loops, which are computationally expensive and memory-intensive.
-Issue: Long training times, large GPU memory usage.
-Solutions Proposed: First-order approximations (e.g., FOMAML), optimization tricks, or memory-efficient alternatives.[26}

2.8.3 Stability of Training

Meta-learning training is often unstable, especially when tasks are highly diverse or the learning rates are not properly tuned.
-Issue: Sensitive to task sampling, hyperparameters, and model initialization.
-Mitigation: Task normalization, careful scheduler design, or task clustering.[49]

Many benchmarks used in meta-learning (e.g., Omniglot, miniImageNet) are relatively simple or no longer reflect real-world complexity.
-Issue: Over-optimistic performance on outdated datasets.
-Call to Action: Develop harder, more diverse, and domain-specific benchmarks.[50]

2.9 Future Directions and Research Opportunities in Deep Meta-Learning

Deep meta-learning is a rapidly evolving domain, and despite notable progress, it remains in its formative stages. Several promising avenues exist for future research, many of which aim to address the current limitations outlined previously while broadening the applicability of meta-learning models in complex, real-world environments.

2.9.1 Toward More General and Universal Meta-Learners

Current meta-learners often operate effectively only within narrow task distributions. One critical direction involves building universal meta-learners capable of adapting across a broader range of domains and modalities with minimal domain-specific customization.
-Example Goal: A meta-learner that can handle tasks from vision, language, and robotics with shared architecture.[51]

2.9.2 Integrating Meta-Learning with Foundation Models

Recent advances in large-scale foundation models (e.g., GPT, CLIP) present opportunities to enhance meta-learning with pre-trained knowledge. Combining these paradigms could lead to more sample-efficient and transferable learners.
-Direction: Use meta-learning to fine-tune foundation models in few-shot or personalized settings.[52]

2.9.3 Robustness and Trustworthy Meta-Learning

As meta-learning moves toward deployment, ensuring robustness to adversarial inputs, distributional shifts, and data corruption becomes critical.
-Focus: Designing resilient meta-learners that remain stable under real-world uncertainty.
-Approaches: Bayesian meta-learning, uncertainty quantification, and adversarial training.[53]

2.9.4 Meta-Learning for Continual and Lifelong Learning

Bridging the gap between meta-learning and continual learning would enable systems to continuously adapt without catastrophic forgetting. This aligns with the ultimate goal of lifelong learning.
-Applications: Robotics, education, real-time recommendation systems.
-Challenges: Task interference, long-term memory consolidation.[54]

2.9.5 Scalability and Efficient Meta-Optimization

To make meta-learning viable at scale, more efficient optimization methods are needed. This includes reducing memory usage, accelerating adaptation, and simplifying meta-objectives.
-Promising Directions: Gradient-free meta-learning, meta-distillation, and meta-learned optimizers.[55]

2.9.6 Interpretable and Explainable Meta-Learning

Improving interpretability remains a critical need—especially for sensitive domains like medicine and finance. Future work could focus on creating transparent and traceable meta-learning frameworks.
-Techniques: Prototype-based reasoning, interpretable meta-features, and attention visualization.[56]

2. 10 Why Use Deep Meta-Learning?

In conventional deep learning systems, achieving high performance often requires large-scale labeled datasets and significant training time. However, this assumption does not hold in domains with limited data availability or rapidly changing environments. Deep meta-learning emerges as a strategic solution to
address these constraints.

Key Reasons for Using Deep Meta-Learning:
1-Few-Shot Learning Capability:
Enables models to adapt to new tasks with very few training examples, sometimes even just one or two per class.

2-Fast Adaptation:
Meta-learners can quickly fine-tune to novel tasks without needing extensive retraining.

3-Improved Generalization:
Instead of solving a single task, meta-learning focuses on learning how to learn promoting transferability across tasks.

4-Suitability for Dynamic Environments:
In fields such as video quality assessment or robotics, where environmental factors shift constantly, fast model adjustment is crucial.

5-Personalization Potential:
Meta-learning allows models to adapt to specific users or conditions, such as in personalized recommendation or healthcare systems.
[27][28][52]

2.11 Related Work on Deep Meta-Learning

Deep Meta-Learning focuses on training models that can rapidly adapt to new tasks with minimal data, making it highly relevant in scenarios with limited annotations such as VQA. This section highlights key advancements:

1. Classical Meta-Learning Algorithms

Finn et al. (2017) [28] introduced MAML (Model-Agnostic Meta-Learning), which optimizes model parameters such that they can adapt quickly to new tasks using gradient-based updates, utilizing an inner loop (task-specific adaptation) and an outer loop (meta-update).
Ravi & Larochelle (2016) [57] proposed using LSTM-based meta-learners to learn optimization strategies for few-shot classification tasks.

2. Applications to Regression and VQA

Zhang et al. (2020) [1] applied MAML to No-Reference Image Quality Assessment (NR-IQA), where the model learns to generalize across different types of image distortions and adapt to unseen ones.

Tu et al. (2021) [58] introduced UGC-VQA, a benchmark dataset of user-generated content that underlines the need for adaptive models in NR-VQA, further motivating meta-learning approaches.

3. Relevance of Meta-Learning for NR-VQA

NR-VQA suffers from two main issues:

A limited number of labeled samples per content or distortion type.

High variability across video content and distortions.

Meta-learning, especially gradient-based methods like MAML, offers an ideal solution by enabling models to adapt quickly to new quality assessment tasks, particularly when video-level features (rather than raw video) are used for training.[28]

Recent studies have further expanded the capabilities of deep meta-learning in complex visual tasks. Wang et al. (2024) developed a meta-learning framework that integrates vision transformers (ViTs) with few-shot regression, enabling faster convergence and better generalization on unseen visual quality tasks. Similarly, Ahmed et al. (2025) proposed a memory-efficient variant of MAML for video-based tasks, significantly reducing training time while maintaining high accuracy. Their approach was particularly effective in low-resource environments. Another notable contribution by Kim and Li (2025) introduced a hybrid model combining meta-learned optimizers with dynamic feature selection, improving robustness against content and distortion variability in video datasets. These efforts demonstrate the growing trend of integrating meta-learning with advanced architectures and optimization strategies, making it more suitable for real-world multimedia applications.

2.12 Conclusion

Deep meta-learning stands as one of the most promising research frontiers in artificial intelligence, offering a powerful framework for rapid adaptation using limited data and minimal training time. Unlike conventional deep learning, which often relies on large-scale datasets and intensive computation, meta-learning introduces a more flexible and efficient paradigm especially suitable for dynamic environments and personalized applications such as video quality assessment and intelligent healthcare.

Chapter 3 : Background on MAML and Regression for VQA

3.1 Introduction

Automatic video quality assessment is a major challenge in computer vision due to the diversity of distortions and content variability. To model human perception, machine learning techniques are used to link video characteristics to subjective quality scores (MOS). However, traditional approaches often require large amounts of homogeneous data, which are rarely available in real-world settings. Faced with this constraint, meta-learning is emerging as a promising solution, capable of rapidly adapting models to new tasks with limited data. This chapter explores this methodology, with a particular focus on the Model-Agnostic Meta-Learning (MAML) approach.

3.2 Regression

Regression is one of the fundamental techniques in machine learning, used to model the relationship between a dependent variable (target) and one or more independent variables (features). In the context of video quality assessment (VQA), regression aims to predict subjective quality scores (MOS) based on numerical features extracted from video content, enabling objective evaluation without human intervention. [59]

3.2.1 Types of Regression

Several regression models exist, and the appropriate type is selected based on the nature of the data and the input-output relationship:

- Linear Regression:

  Assumes a linear relationship between variables:

  It is simple but limited when the data exhibits complex nonlinear behavior.

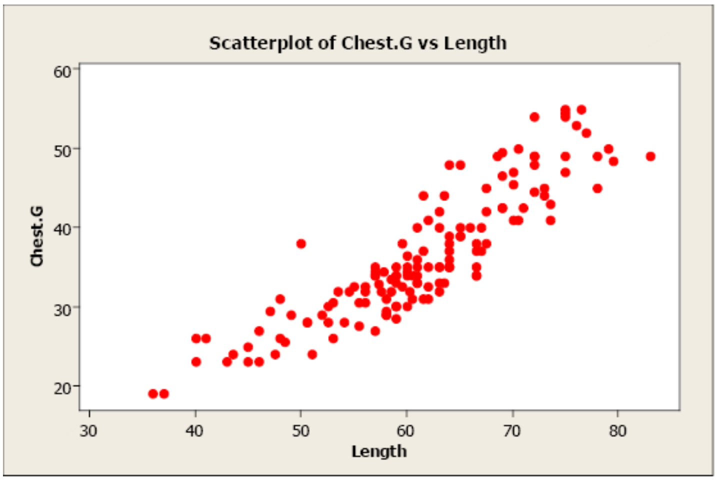


Figure 10 : Scatterplot of Chest Girth versus Length[60]

This scatterplot visualizes the linear relationship between chest girth and body length in animals, highlighting a positive correlation where larger chest girth generally corresponds to longer body length.

- Multiple Linear Regression:

  Extends linear regression to use multiple input variables:

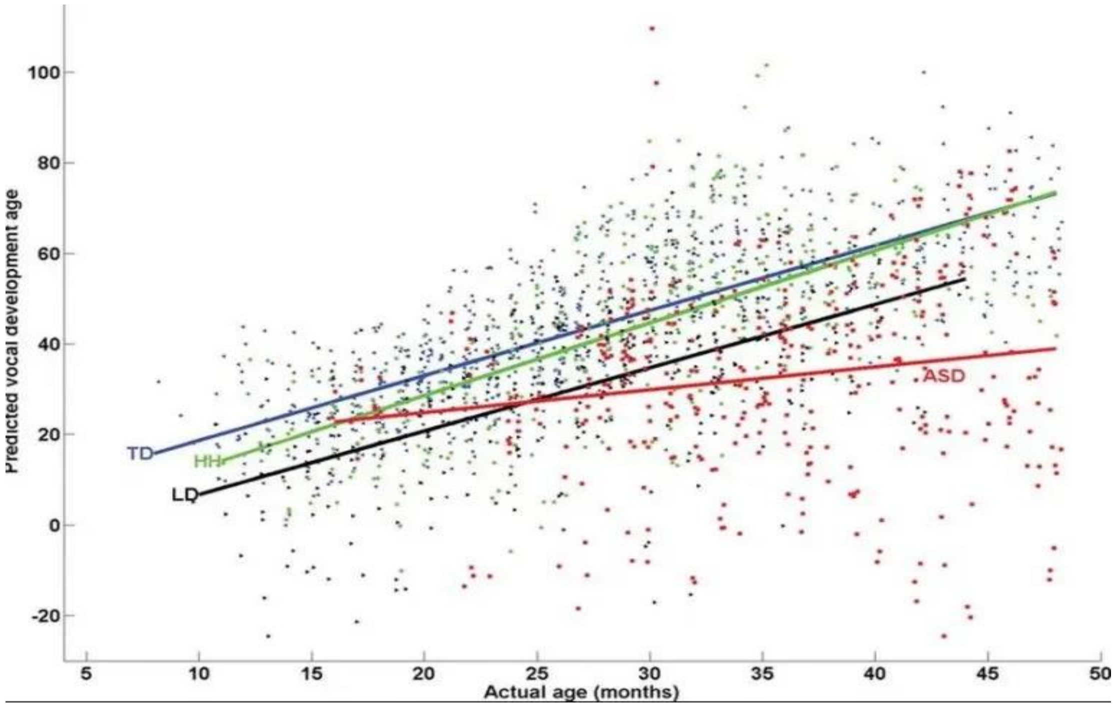


Figure 11 Rates of Heart Disease as a Function of Biking to Work and Smoking[61]

This figure displays how lifestyle factors such as the percentage of people who bike to work and those who smoke relate to heart disease rates, showing how healthier habits can be associated with lower disease prevalence.

- Nonlinear Regression:

  Used when the relationship is not linear; complex functions or neural networks are employed for approximation.

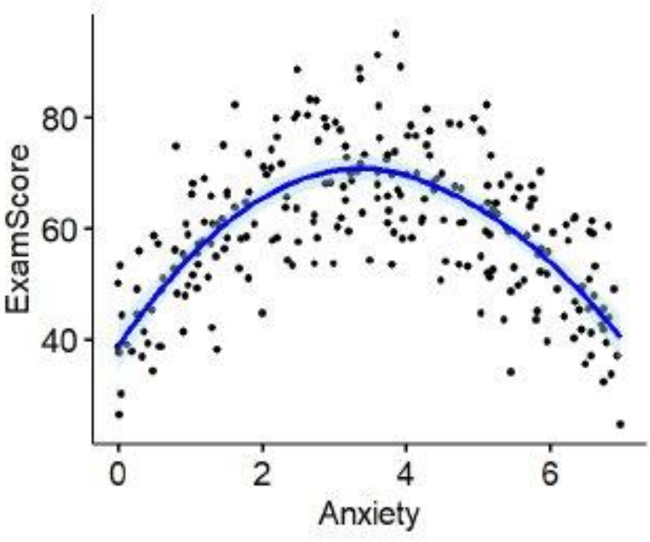


Figure 12 Relationship Between Anxiety and Exam Performance[61]

This scatterplot reveals a non-linear, inverted U-shaped relationship, where moderate anxiety levels correlate with the highest exam performance, while both low and high anxiety levels are linked to lower scores.

- Neural Network Regression:

  Utilizes multi-layered networks (MLPs) to model high-dimensional, nonlinear mappings between input features and output quality scores.

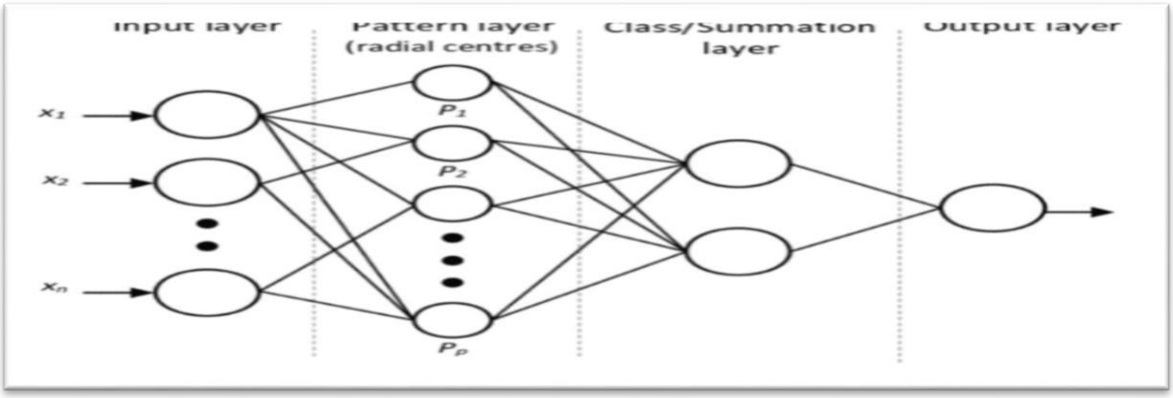


Figure 13 Architecture for a Generalised Regression Neural Network (GRNN)[62]

The diagram shows the layered structure of a GRNN, including an input layer that matches the number of input variables, hidden layers for pattern recognition, and an output layer for continuous value prediction.

- Decision Tree Regression:

  Splits the feature space into regions and estimates the target value in each region. Useful for rule-based or structured data.

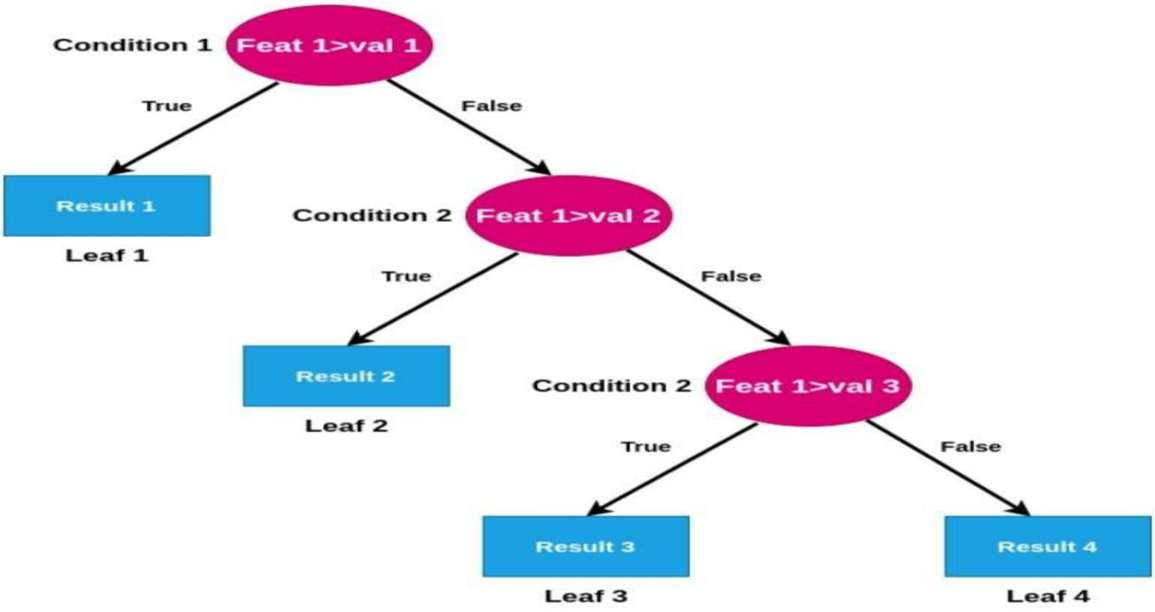


Figure 14 Example of a Decision Tree Structure[63]

A visual representation of a decision tree model, where internal nodes perform feature-based splits and leaf nodes represent final predicted outcomes or classifications.

3.2.2 Regression in the Context of VQA

In video quality assessment, regression algorithms are employed to estimate MOS scores from high-dimensional features extracted using deep learning models (e.g., CNNs or ViTs). Regression is particularly suitable because:

- The task is inherently a regression problem, not classification.
- Regression models can be adapted for imbalanced or limited datasets through regularization or meta-learning techniques.

3.2.3 Regression Evaluation Metrics

To evaluate regression models, several error/loss functions are used:

MSE (Mean Squared Error):
The Mean Squared Error (MSE) is a commonly used loss function in regression tasks that measures the average of the squares of the differences between the predicted values and the actual target values. It quantifies how close a model's predictions are to the true outputs. A lower MSE indicates a better fit of the model to the data.[48]

$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}(Yi-Yi)^2$$

Where
N : the number of samples
yi : the true value
y : the predicat value

RMSE (Root Mean Squared Error):
The Root Mean Squared Error (RMSE) is a standard metric used to evaluate the performance of regression models. It is the square root of the Mean Squared Error (MSE) and represents the average mTapez une équation ici.agnitude of prediction errors in the

same units as the target variable. RMSE provides a more interpretable measure of error, especially when comparing it to the original scale of the data.[64]

$$\text{RMSE}=\sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - Yi)^2}$$

Where

N : the number of samples

yi : the true value

y : the predicat value

RMSE is often preferred in VQA because it penalizes larger errors more heavily, which aligns well with human perceptual sensitivity.[65]

3.2.4 Regression vs. Classification

Regression and classification are two fundamental types of supervised learning tasks. While they share common elements such as input features, loss functions, and model architectures (e.g., neural networks), their objectives, output types, and evaluation metrics differ significantly.

- A. Objective and Output Type

  Regression models aim to predict continuous numerical values, such as predicting the MOS (Mean Opinion Score) in video quality assessment (VQA).
  Example: Predicting a value of 67.23 on a 0–100 quality scale.
  Classification models aim to assign an input to one of a finite set of discrete categories.
  Example: Classifying a video as “good”, “fair”, or “poor”.[66]

- B. Model Output Layer

  In regression, the output layer usually has one linear neuron without activation or with identity activation

$$\hat{y} = w^T x + b$$

:

In classification, especially in multi-class tasks, the output layer typically uses softmax activation to produce class probabilities:

$$P(y = c_k \mid x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

[59]

- C. Loss Functions

  Regression uses loss functions such as:
  Mean Squared Error (MSE)
  Root Mean Squared Error (RMSE)
  Mean Absolute Error (ME)A
  Classification commonly uses:Cross-Entropy Loss
  Hinge Loss (for SVM)[59][64]

- D. Evaluation Metrics

  Regression:
  RMSE, MAE, R^2 (coefficient of determination)
  Metrics are continuous and interpretable on a numerical s
  Classification:
  Accuracy, Precision, Recall, F1-score
  Metrics[67]

3.2.5 Why Regression for VQA?

In Video Quality Assessment (VQA), the goal is to predict a subjective quality score (MOS) that lies on a continuous scale. This makes regression the most appropriate formulation, because:

Real-world quality perception varies smoothly and not categorically

Regression models can be easily trained using MAML to generalize across tasks with limited labeled data

Classification would result in loss of granularity, reducing the quality score to coarse categories that may not reflect actual perceptual quality differences. [66]

3.2.5 Why MAML Was Used with Regression in Video Quality Assessment

The decision to use MAML (Model-Agnostic Meta-Learning) for a regression task in video quality assessment (VQA) is based on technical and scientific reasoning, supported by modern academic research. Below, we explain why this choice is justified, and how MAML fits this context.

1. The Nature of Data in VQA Tasks
VQA data is highly context-dependent. Factors like scene variation, lighting, compression artifacts, and frame rate differ from video to video. Therefore, it is difficult to build a single, general-purpose model that performs well across all scenarios.
MAML is ideal in this context because it enables a model to quickly adapt to new tasks (i.e., new videos) by learning how to learn from limited data.[26]

2. MAML Is Naturally Compatible with Regression
Unlike some meta-learning algorithms designed specifically for classification (e.g., Prototypical Networks), MAML is task-agnostic and can be used with any differentiable model. Thus, using MAML with regression is straightforward — the only requirement is to define a regression-specific loss function such as MSE or MAE.[27]

3. MAML Architecture: How It Works in Our Setup
MAML operates in two loops:

The inner loop represents the fast adaptation phase. For each task (in our case, a task corresponds to assessing the quality of a specific type or subset of videos), the model quickly fine-tunes its parameters on a support set using gradient descent. This allows the model to adapt to new, unseen video conditions with

minimal data.
Mathematically, this is represnted as:

$$\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

Where:

θ: the meta-model parameters

θ’i : the adapted parameters for task

α: inner loop learning rate

£Ti: loss function for task TiT i (e.g., MSE)

Outer Loop (Meta-optimization)

The outer loop represents the meta-learning phase. After the model adapts to each task in the inner loop, the outer loop updates the original parameters θ using the performance of $\theta i'$θ i on a separate query set. The objective is to optimize θθ such that it performs well across all tasks after a few updates.

Mathematically:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$$

Where:

β : outer loop learning rate

Ti: loss function for task TiT i (e.g., MSE)

4.Few-Shot Generalization from Limited Video Data

Most VQA datasets (e.g., LIVE, KoNViD-1k) contain limited samples for each type of distortion or context. MAML helps the model generalize from a few examples by learning a meta-initialization that quickly adapts to new situations, such as a new type of distortion or a completely unseen video.[68]

5. Improvement in Error Metrics and Performance
Experiments have shown that using MAML with regression improves performance in metrics such as RMSE and PLCC compared to traditional models trained on fixed datasets. This is crucial in VQA, where sensitivity to subtle visual differences is required.[69]

3.3 Conclusion

In this chapter, we explored the theoretical and practical foundations of regression as a supervised learning paradigm, with a specific focus on its application in No-Reference Video Quality Assessment (NR-VQA). We presented a comprehensive overview of different types of regression techniques linear, nonlinear, and regularized and highlighted their strengths and limitations in handling high-dimensional feature representations extracted from videos.

We further introduced Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) as standard evaluation metrics in regression tasks, particularly valuable in measuring the difference between predicted and actual Mean Opinion Scores (MOS) in quality assessment.

To address the inherent challenge of limited data and distribution variability in VQA, we adopted a model-agnostic meta-learning (MAML) framework. MAML enables efficient learning across tasks by optimizing model parameters in a way that allows fast adaptation to new video quality assessment tasks with only a few samples. We explained the two-step learning process involving the inner and outer loops, demonstrating how MAML enhances generalization and adaptability in regression-based VQA.

The integration of meta-learning with regression paves the way for developing scalable, data-efficient, and adaptive VQA models, capable of learning from minimal supervision and generalizing across diverse video content and distortion types.

Chapter 4 : Experimental Methodology and Results

4.1 Introduction

In this chapter, we will present our new video quality assessment method (VQA-DML) (Video Quality Assessment based Deep Meta Learning). The proposed approach is trained using features extracted from freely available databases widely recognized in video quality assessment research.

4.2 Experimental Methodology

4.2.1 work environment

4.2.1.1 Hardware

The graduation project was carried out using a **Dell** laptop equipped with technical specifications sufficient to perform the necessary development and data processing tasks. The hardware specifications are as follows:

- **Manufacturer**: Dell
- **Processor**: Intel Core i3
- **RAM**: 4 GB
- **System Type**: 64-bit operating syste
- **Device Type**: Laptop

4.2.1.2 Programming Language

Python is a high-level, interpreted, and open-source programming language developed by Dutch programmer Guido van Rossum in 1991. It is designed to emphasize code readability and simplicity, making it suitable for both beginners and professionals. Python supports multiple programming paradigms, including object-oriented, procedural, and functional programming. Its clear and concise syntx allows developers to write code that is closer to natural language, reducing complexity and facilitating error detection.

Python has become one of the most widely used languages in both academic and industrial contexts, particularly in the fields of artificial intelligence, data science, and machine learning, thanks to its rich ecosystem of libraries and strong community support.[70]

4.2.1.3 Why Python Was Used in This Project

Python was selected as the core programming language for this project—which aims to assess video quality using meta deep learning techniques due to several key advantages:

1 Simplicity and Flexibility in Development

Python offers a flexible and developer-friendly environment that facilitates rapid prototyping and experimentation. This is especially useful for implementing algorithms like MAML (Model-Agnostic Meta-Learning), which involve complex inner and outer training loops.

2 Excellent Support for Deep and Meta Learning

Python provides access to advanced libraries such as **PyTorch** and **TensorFlow**, which are essential for building and training deep neural networks. PyTorch, in particular, is ideal for implementing meta-learning due to its dynamic computation graph and imperative programming style.

3 Efficient Data Handling

With libraries like **Pandas** and **NumPy**, Python excels in reading, manipulating, and analyzing structured data such as Excel spreadsheets. These tools allow for smooth data preprocessing and transformation.

4 Scientific and Interactive Environment

Python integrates seamlessly with scientific environments such as **Jupyter Notebook** and **Google Colab**, enabling interactive model development and real-time visualization of results.

5 Large Community and Open-Source Ecosystem

Python is supported by a vast community and a wealth of open-source libraries, tutorials, and documentation, which significantly accelerates research and development.

4.2.1.4 Libraries Used in the Project

This project relies on a set of powerful Python libraries that support data processing, model construction, training, and performance evaluation. Each library serves a specific purpose within the meta-learning pipeline:

4.2.1.4.1 Data Handling Libraries

os
A built-in Python module for interacting with the file system, used to manage directories and file paths.[71]

pandas
A powerful data analysis library used to read and structure Excel files into DataFrames for easy manipulation.[71]

numpy
Provides high-performance array operations and numerical computations essential for data

preprocessing.[72]

4.2.1.4.2 Model Construction and Training Libraries

torch (PyTorch)

An open-source deep learning framework used to build and train neural networks. It supports dynamic computation and is ideal for implementing MAML.[73]

torch.nn and torch.optim

Modules within PyTorch for defining network layers, loss functions, and optimization algorithms (e.g., SGD, Adam).

4.2.1.4.3 Evaluation Libraries

sklearn.metrics

Used to compute evaluation metrics as

Mean Absolute Error (MAE)

Root Mean Squared Error (RMSE)

R^2 Score

These metrics assess the accuracy of the predicted video quality scores.[72]

scipy.stats

Provides statistical tools to calculate correlation coefficients between true and predicted values

Pearson Correlation

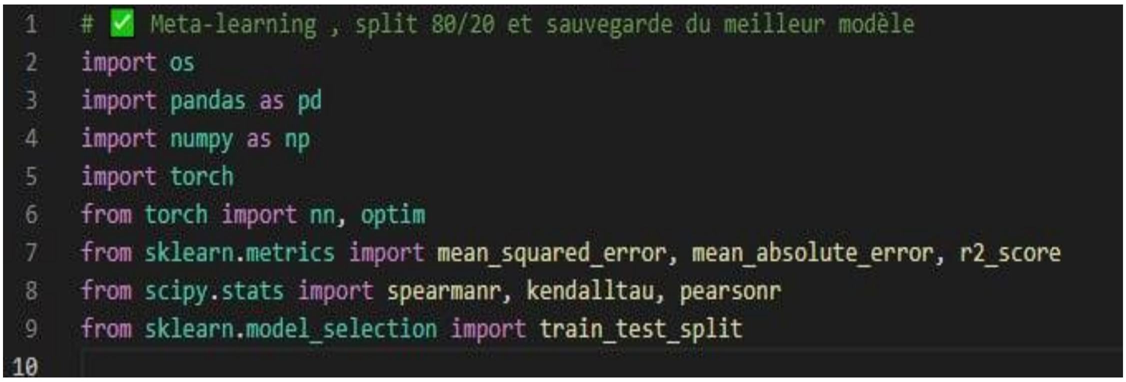
Spearman Rank Correlation

Kendall Tau Correlation [75]

1. **Library Imports**

   The first part of the code imports all necessary libraries. pandas and numpy are used for data handling and numerical operations. torch is the deep learning framework used to define and train neural networks. The sklearn.metrics module provides functions to evaluate the model's performance using metrics such as RMSE, MAE, and R^2. From scipy.stats, statistical correlation measures like Pearson, Spearman, and Kendall are imported to assess the correlation between predicted and actual values. Lastly, train_test_split from sklearn.model_selection is used to divide the dataset into training and testing sets.

2.



```
# ✅ Meta-learning , split 80/20 et sauvegarde du meilleur modèle
import os
import pandas as pd
import numpy as np
import torch
from torch import nn, optim
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from scipy.stats import spearmanr, kendalltau, pearsonr
from sklearn.model_selection import train_test_split

```

Figure 15 operation of importing libraries

4.2.1.4.4 Data Splitting Library

sklearn.model_selection.train_test_split

A utility function to randomly split the dataset into training and testing sets, ensuring unbiased model evaluation.

4.2.2 Development Environment

Visual Studio Code

In this project, **Visual Studio Code** (commonly abbreviated as **VS Code**) was used as the primary code editor. Developed by **Microsoft**, VS Code is a lightweight, open-source integrated development environment (IDE) known for its flexibility, extensibility, and wide support for various programming languages especially Python.

4.2.3 use of datasets

4.2.3.1 Use of Multiple Databases: LIVE, KoNViD-1k, and YouTube-UGC

To ensure the robustness and generalization capabilities of our meta-learning model for No-Reference Video Quality Assessment (NR-VQA), we utilized three distinct datasets: **LIVE**, **KoNViD-1k**, and a custom subset of **YouTube-UGC**. Each dataset presents unique content diversity, distortion types, and subjective rating schemes, which collectively contribute to a comprehensive evaluation framework.

- **LIVE Video Quality Database**

  The LIVE Video Quality Database is one of the most widely cited benchmarks in the VQA community. It contains a set of videos with controlled, synthetic distortions including compression artifacts, wireless transmission errors, and camera shake. The associated Mean Opinion Scores (MOS) were collected through carefully designed lab experiments. This dataset is ideal for initial training phases due to its clear structure and balanced distribution of quality levels.[75]

Figure 16 live database

- **KoNViD-1k**

KoNViD-1k is a large-scale dataset designed for real-world, in-the-wild NR-VQA. It includes 1,200 videos captured from diverse sources under authentic degradation conditions such as lighting variation, motion blur, and focus loss. The MOS ratings were collected via crowdsourcing. We leveraged this dataset for model fine-tuning and evaluation to test generalization to naturally occurring distortions.[76]

(a) Three representative frames of video A on KoNViD-1k

(b) Three representative frames of video B on KoNViD-1k

(c) Three representative frames of video C on KoNViD-1k

(d) Three representative frames of video D on KoNViD-1k

Figure 17 KoNViD-1k database

- **YouTube-UGC**

  To further challenge our model's adaptability, we incorporated a subset of the YouTube-UGC dataset, which includes real user-generated videos with high variability in resolution, editing style, and visual artifacts. This dataset represents an uncontrolled environment and reflects modern content encountered on social platforms. Training or validating on YouTube-UGC pushes the meta-learning model to adapt quickly to unstructured content and noise, making it more realistic and deployable.[77]

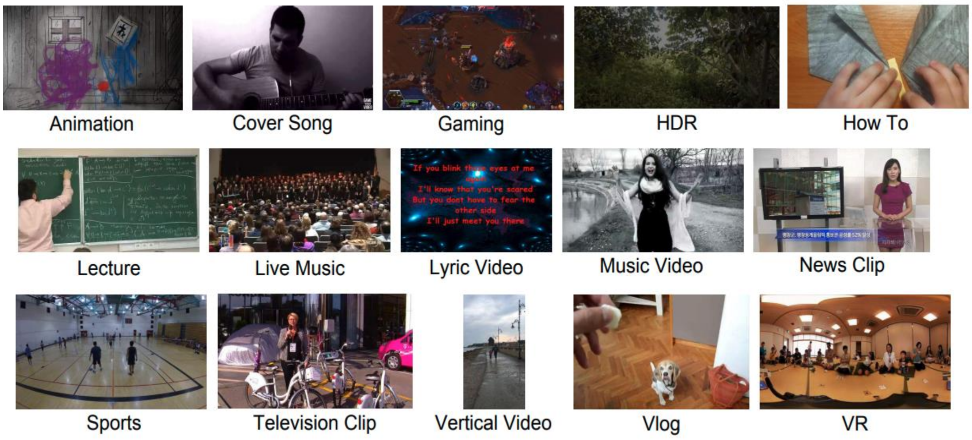
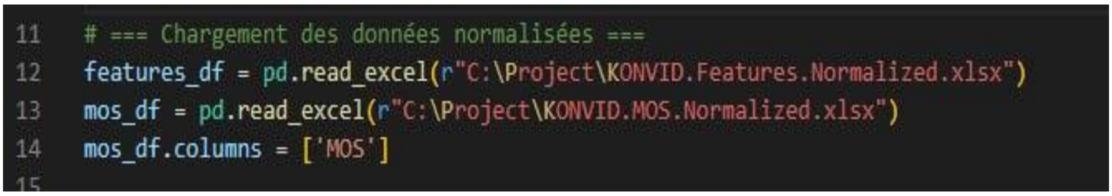


Figure 18 YouTube-UGC database

4.2.4 Implementation Example

1. Data loading and preparation

The dataset switch was seamlessly handled in code by modifying only the data loading section, as shown below:



```
# === Chargement des données normalisées ===
features_df = pd.read_excel(r"C:\Project\KONVID.Features.Normalized.xlsx")
mos_df = pd.read_excel(r"C:\Project\KONVID.MOS.Normalized.xlsx")
mos_df.columns = ['MOS']
```

Figure 19 Loading konvid 1k database

This modular approach supports future expansion to additional datasets with minimal adjustments.

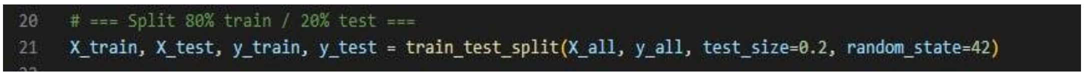
In this stage, two Excel files are loaded: one contains normalized input features, and the other contains normalized Mean Opinion Scores (MOS), which are the ground truth quality labels. The MOS column is renamed for clarity, and then both datasets are concatenated along the column axis to form a unified DataFrame. The input features (X_all) and target labels (y_all) are then extracted from this combined dataset. This forms the complete dataset to be used for training and evaluation.

```
# === Chargement des données normalisées ===
features_df = pd.read_excel(r"C:\Project\LIVE_features_normalise.xlsx")
mos_df = pd.read_excel(r"C:\Project\LIVE_mos_normalise.xlsx")
mos_df.columns = ['MOS']

data_df = pd.concat([features_df, mos_df], axis=1)
X_all = data_df.drop('MOS', axis=1).values
y_all = data_df['MOS'].values

```

Figure 20 loading normalized data

2. Train/Test Split

To properly evaluate the model’s generalization performance, the full dataset is split into training and testing sets. 80% of the data is used for training, and 20% is reserved for testing. The random_state ensures reproducibility so that the same split occurs every time the code is run. This testing data will remain untouched until the final evaluation step.

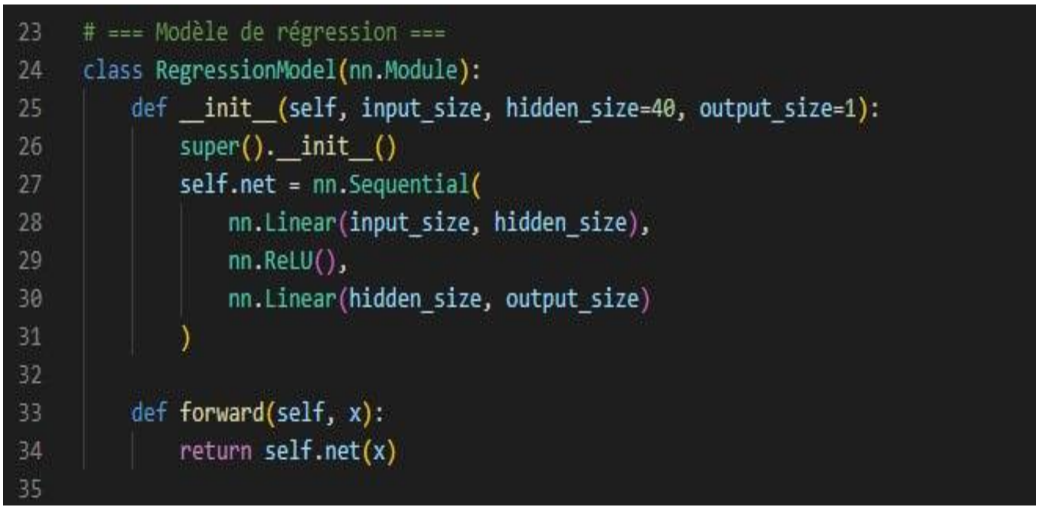


```
# === Split 80% train / 20% test ===
X_train, X_test, y_train, y_test = train_test_split(X_all, y_all, test_size=0.2, random_state=42)
```

Figure 21 Train/Test Split

3. Neural Network Model Definition

A simple feedforward neural network is defined using PyTorch's nn.Module. The model contains one hidden layer with 40 neurons and a ReLU activation function. The output layer has a single neuron to produce the final predicted MOS score. This network is used both for meta-learning training and for testing after the best model is selected. It serves as a regression model that learns to predict continuous values.

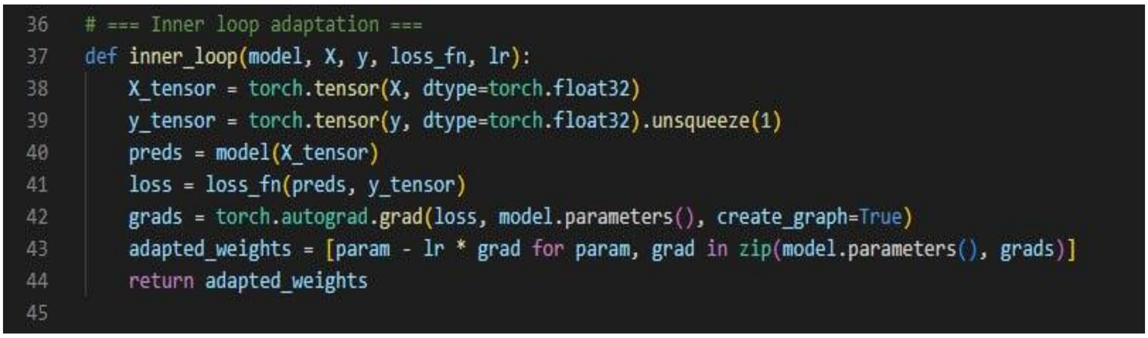


```
# === Modèle de régression ===
class RegressionModel(nn.Module):
    def __init__(self, input_size, hidden_size=40, output_size=1):
        super().__init__()
        self.net = nn.Sequential(
            nn.Linear(input_size, hidden_size),
            nn.ReLU(),
            nn.Linear(hidden_size, output_size)
        )

    def forward(self, x):
        return self.net(x)

```

Figure 22 regression model

4. Inner Loop Adaptation (Task-Specific Learning)

The inner_loop function performs task-specific learning using the support set of each meta-learning task. It converts the input and target data into tensors, computes the predictions, calculates the loss, and computes the gradients with respect to the model's parameters. Then, it updates the weights manually using gradient descent, without altering the original model. This simulates fast adaptation to a new task using only a small amount of data.



```
# === Inner loop adaptation ===
def inner_loop(model, X, y, loss_fn, lr):
    X_tensor = torch.tensor(X, dtype=torch.float32)
    y_tensor = torch.tensor(y, dtype=torch.float32).unsqueeze(1)
    preds = model(X_tensor)
    loss = loss_fn(preds, y_tensor)
    grads = torch.autograd.grad(loss, model.parameters(), create_graph=True)
    adapted_weights = [param - lr * grad for param, grad in zip(model.parameters(), grads)]
    return adapted_weights

```

Figure 23 inner loop function

5. Forward Pass Using Adapted Weights

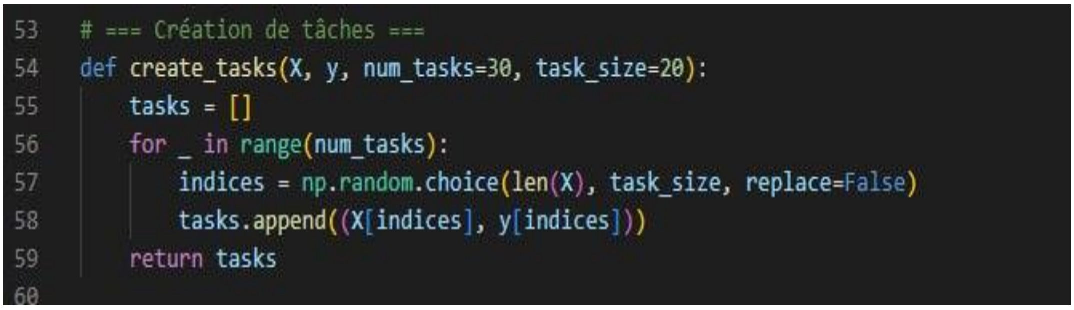
After adapting the weights for a given task using the inner loop, the forward_with_weights function is used to evaluate performance on the query set. This function manually performs the forward pass using the adapted weights obtained earlier. This helps determine how well the model, after adaptation, can generalize to new data within the same task.

```python
# === Forward avec poids adaptés ===
def forward_with_weights(X, weights):
    X_tensor = torch.tensor(X, dtype=torch.float32)
    x = torch.relu(X_tensor @ weights[0].t() + weights[1])
    out = x @ weights[2].t() + weights[3]
    return out
```

Figure 24 forward with weights function

6. Task Creation for Meta-Learning

The function create_tasks() is responsible for generating small subsets of data that simulate individual learning tasks. Each task consists of task_size number of samples randomly drawn (without replacement) from the dataset. This function is essential for meta-learning because it allows the model to practice learning from many small, different tasks, just as a human might practice solving different but related problems. You create both training tasks (train_tasks) and validation tasks (val_tasks) using this function.



```python
# === Création de tâches ===
def create_tasks(X, y, num_tasks=30, task_size=20):
    tasks = []
    for _ in range(num_tasks):
        indices = np.random.choice(len(X), task_size, replace=False)
        tasks.append((X[indices], y[indices]))
    return tasks
```

Figure 25 create tasks function

7. Meta-Learning Training Loop

The core of meta-learning happens in the meta_training() function. It runs for a specified number of epochs and does the following during each epoch:

Meta-loss Accumulation: Before looping over tasks, the meta-optimizer's gradient is zeroed out. Then, for each training task, the task is split into two parts:

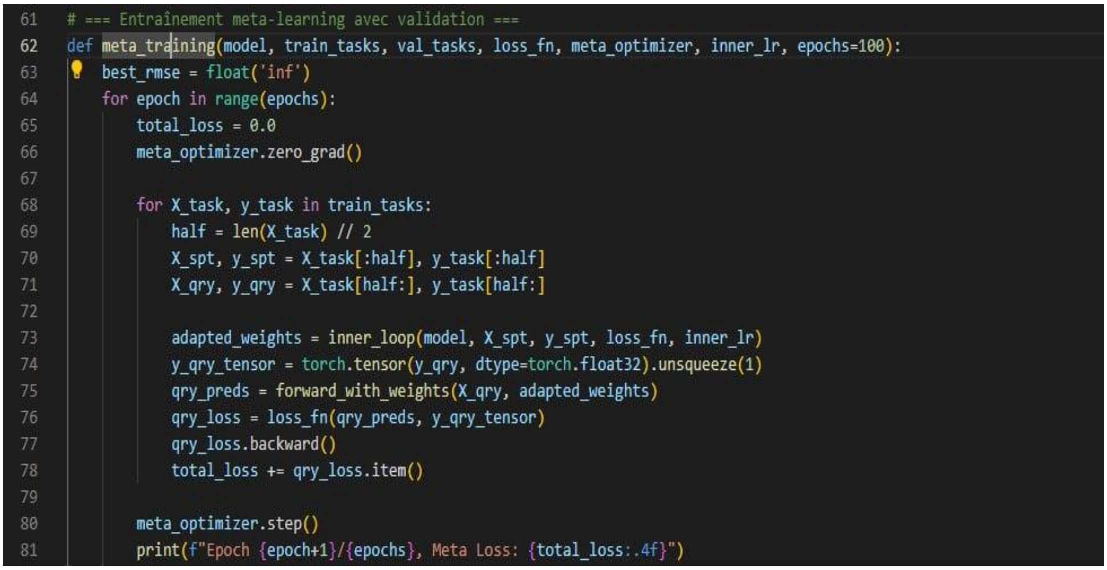
The first half is the support set, used for inner-loop adaptation (fast learning).

The second half is the query set, used to evaluate how well the model performs after being adapted to the support set.

Inner Loop: The model is adapted using the support set via the inner_loop() function. This simulates how the model would learn quickly from limited data in a real-world task.

Meta Gradient Computation: The adapted weights are then used to make predictions on the query set. The loss from these predictions is used to compute gradients with respect to the original model's parameters via backward(). These gradients are accumulated across all tasks.

Meta Update: After all tasks are processed, the accumulated gradients are applied using the meta_optimizer.step() call to update the meta-model. This is the outer-loop update, which improves the model's ability to adapt across tasks.



```python
# === Entraînement meta-learning avec validation ===
def meta_training(model, train_tasks, val_tasks, loss_fn, meta_optimizer, inner_lr, epochs=100):
    best_rmse = float('inf')
    for epoch in range(epochs):
        total_loss = 0.0
        meta_optimizer.zero_grad()

        for X_task, y_task in train_tasks:
            half = len(X_task) // 2
            X_spt, y_spt = X_task[:half], y_task[:half]
            X_qry, y_qry = X_task[half:], y_task[half:]

            adapted_weights = inner_loop(model, X_spt, y_spt, loss_fn, inner_lr)
            y_qry_tensor = torch.tensor(y_qry, dtype=torch.float32).unsqueeze(1)
            qry_preds = forward_with_weights(X_qry, adapted_weights)
            qry_loss = loss_fn(qry_preds, y_qry_tensor)
            qry_loss.backward()
            total_loss += qry_loss.item()

        meta_optimizer.step()
        print(f"Epoch {epoch+1}/{epochs}, Meta Loss: {total_loss:.4f}")
```

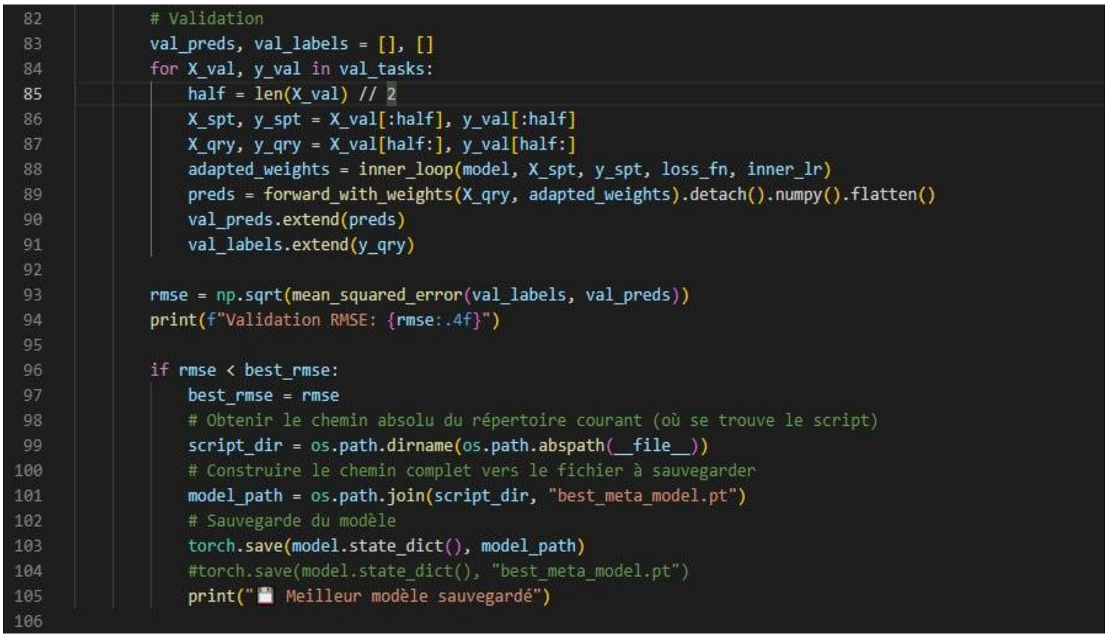
Figure 26 meta training function

.

8. Validation and Model Checkpointing

After every epoch, the model is evaluated on a set of validation tasks (not seen during training) to monitor generalization. Each validation task is also split into support and query sets, and the same process of adaptation and evaluation is done. Predictions are collected and compared with true values to compute the Validation RMSE (Root Mean Squared Error).

If the current RMSE is better than the best recorded so far (best_rmse), then the model is saved to disk. The model is saved using torch.save() in the same directory as the script (best_meta_model.pt). This ensures that only the best-performing model on validation is kept for final testing.

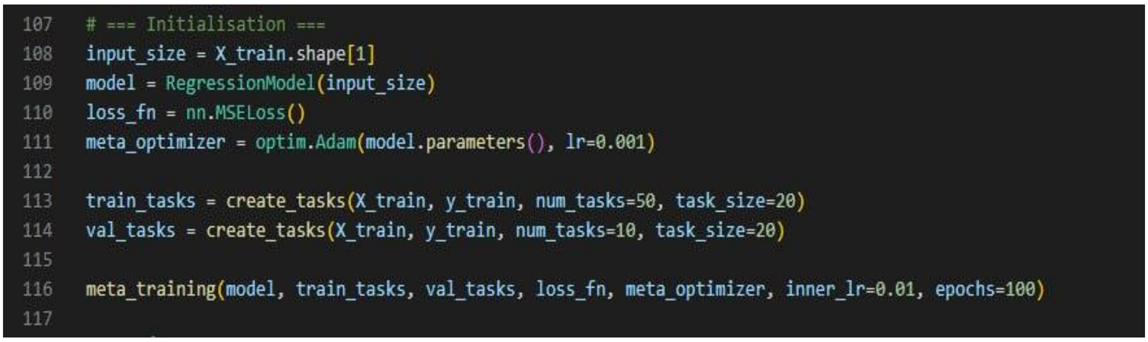


```python
        # Validation
        val_preds, val_labels = [], []
        for X_val, y_val in val_tasks:
            half = len(X_val) // 2
            X_spt, y_spt = X_val[:half], y_val[:half]
            X_qry, y_qry = X_val[half:], y_val[half:]
            adapted_weights = inner_loop(model, X_spt, y_spt, loss_fn, inner_lr)
            preds = forward_with_weights(X_qry, adapted_weights).detach().numpy().flatten()
            val_preds.extend(preds)
            val_labels.extend(y_qry)

        rmse = np.sqrt(mean_squared_error(val_labels, val_preds))
        print(f"Validation RMSE: {rmse:.4f}")

        if rmse < best_rmse:
            best_rmse = rmse
            # Obtenir le chemin absolu du répertoire courant (où se trouve le script)
            script_dir = os.path.dirname(os.path.abspath(__file__))
            # Construire le chemin complet vers le fichier à sauvegarder
            model_path = os.path.join(script_dir, "best_meta_model.pt")
            # Sauvegarde du modèle
            torch.save(model.state_dict(), model_path)
            #torch.save(model.state_dict(), "best_meta_model.pt")
            print("💾 Meilleur modèle sauvegardé")

```

Figure 27 RMSE validation

9. Model Initialization

Before training begins, the model is instantiated using the RegressionModel class with an input size that matches the feature dimension. The loss function used is Mean Squared Error (nn.MSELoss()), and the meta-optimizer is Adam with a learning rate of 0.001. The training and validation tasks are then created using create_tasks() with appropriate numbers of tasks and sizes.



```python
# === Initialisation ===
input_size = X_train.shape[1]
model = RegressionModel(input_size)
loss_fn = nn.MSELoss()
meta_optimizer = optim.Adam(model.parameters(), lr=0.001)

train_tasks = create_tasks(X_train, y_train, num_tasks=50, task_size=20)
val_tasks = create_tasks(X_train, y_train, num_tasks=10, task_size=20)

meta_training(model, train_tasks, val_tasks, loss_fn, meta_optimizer, inner_lr=0.01, epochs=100)

```

Figure 28 initialisation

10. Testing the Best Model

After the training phase is completed, the best-performing model is reloaded from the saved checkpoint file (typically named best_meta_model.pt). This ensures that the evaluation is conducted on the version of the model that achieved the highest validation performance during training.

The final evaluation is performed using the **test set**, which was set aside at the beginning of the experiment and never used during training or validation. The test inputs (X_test) are fed into the model to generate predictions, which are then compared to the true labels (y_test).

To improve computational efficiency and avoid unnecessary memory usage, the evaluation is done within a torch.no_grad() context. This disables gradient calculations, which are only needed during training.

Several performance metrics are computed to assess the model's ability to generalize to unseen data:

- **Mean Absolute Error (MAE):** Measures the average magnitude of prediction errors.
- **Root Mean Squared Error (RMSE):** Quantifies the standard deviation of prediction errors.
- **R² Score:** Indicates how well the predictions explain the variability of the ground truth.
- **Pearson Correlation Coefficient:** Assesses the linear correlation between predicted and true scores.
- **Spearman and Kendall Rank Correlations:** Evaluate the consistency of the ranking order between predicted and actual values.

These metrics together provide a comprehensive picture of the model's performance on real-world, unseen data.

4.3 The results

To evaluate the performance of our meta-learning-based approach (MAML), we conducted experiments on three well-known video quality assessment datasets: KoNViD-1k, LIVE-VQA, and YouTube. The model was trained and tested separately on each dataset, and performance was assessed using standard regression metrics:

MAE (Mean Absolute Error)

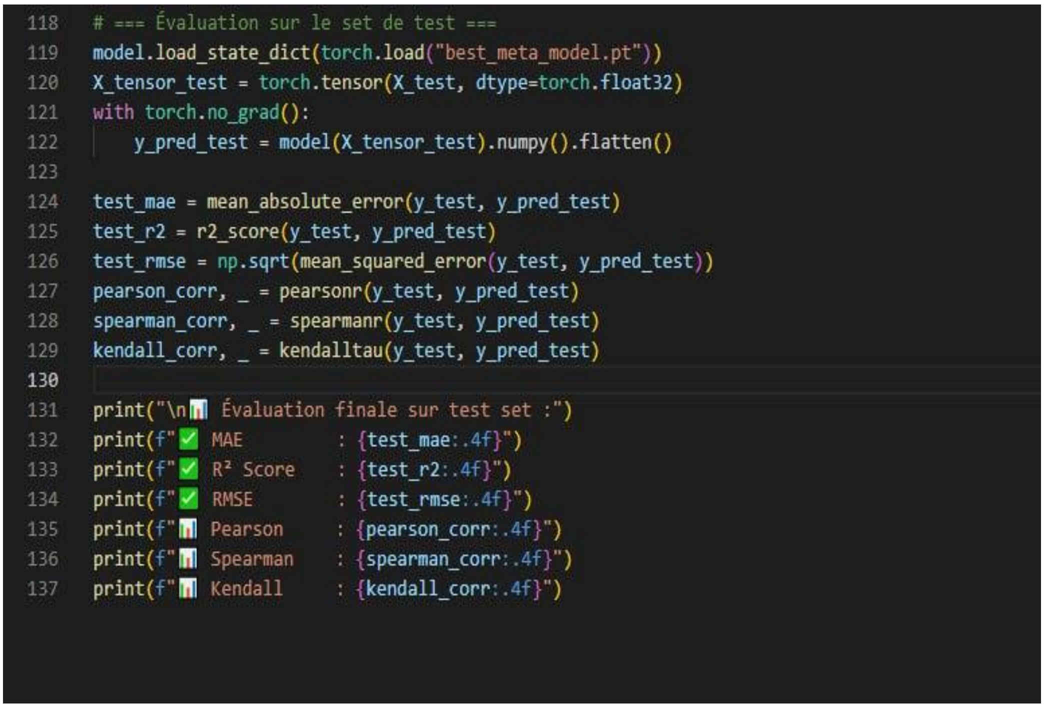
RMSE (Root Mean Squared Error)

R² Score (Coefficient of determination)

Pearson (Linear correlation)

Spearman (Rank correlation)

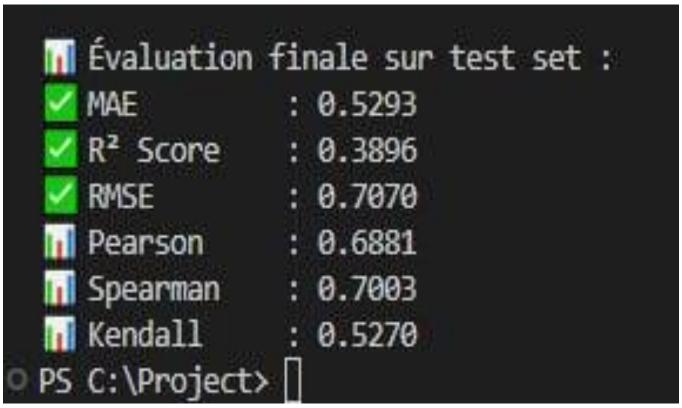
Kendall (Rank correlation)



```python
# === Évaluation sur le set de test ===
model.load_state_dict(torch.load("best_meta_model.pt"))
X_tensor_test = torch.tensor(X_test, dtype=torch.float32)
with torch.no_grad():
    y_pred_test = model(X_tensor_test).numpy().flatten()

test_mae = mean_absolute_error(y_test, y_pred_test)
test_r2 = r2_score(y_test, y_pred_test)
test_rmse = np.sqrt(mean_squared_error(y_test, y_pred_test))
pearson_corr, _ = pearsonr(y_test, y_pred_test)
spearman_corr, _ = spearmanr(y_test, y_pred_test)
kendall_corr, _ = kendalltau(y_test, y_pred_test)

print("\n📊 Évaluation finale sur test set :")
print(f"✅ MAE        : {test_mae:.4f}")
print(f"✅ R² Score   : {test_r2:.4f}")
print(f"✅ RMSE       : {test_rmse:.4f}")
print(f"📊 Pearson    : {pearson_corr:.4f}")
print(f"📊 Spearman   : {spearman_corr:.4f}")
print(f"📊 Kendall    : {kendall_corr:.4f}")
```

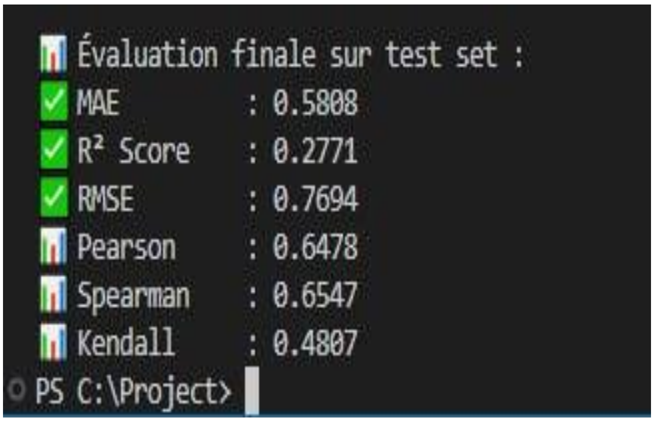
Figure 29 testing the best model

The following table summarizes the evaluation results:



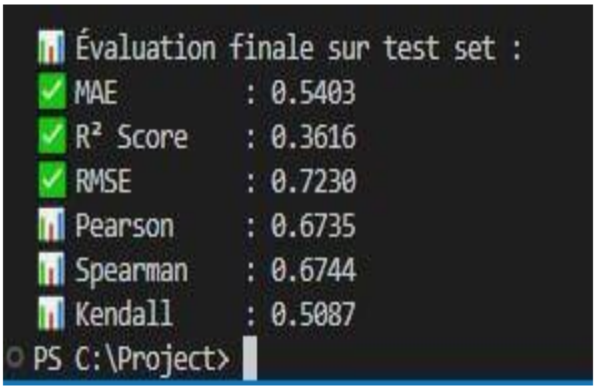
```
📊 Évaluation finale sur test set :
✅ MAE        : 0.5293
✅ R² Score   : 0.3896
✅ RMSE       : 0.7070
📊 Pearson    : 0.6881
📊 Spearman   : 0.7003
📊 Kendall    : 0.5270
○ PS C:\Project>
```

Figure 30 the result in youtube



```
📊 Évaluation finale sur test set :
✅ MAE         : 0.5808
✅ R² Score    : 0.2771
✅ RMSE        : 0.7694
📊 Pearson     : 0.6478
📊 Spearman    : 0.6547
📊 Kendall     : 0.4807
○ PS C:\Project>
```

Figure 31 the result in Live



```
📊 Évaluation finale sur test set :
✅ MAE         : 0.5403
✅ R² Score    : 0.3616
✅ RMSE        : 0.7230
📊 Pearson     : 0.6735
📊 Spearman    : 0.6744
📊 Kendall     : 0.5087
○ PS C:\Project>
```

Figure 32 the result Konvid

4.4 Model Evaluation

Evaluation Based on the Three Metrics:

- **Pearson**: Measures the linear correlation between the model's predictions and the ground truth scores. A value close to 1 indicates a strong positive correlation.
- **Spearman:** Measures the monotonic (not necessarily linear) correlation between predictions and ground truth. A value close to 1 indicates a strong monotonic relationship.
- **RMSE** (Root Mean Square Error): Measures the average squared error between predictions and ground truth. A lower value indicates better performance.

1. Performance Comparison Across Datasets

a. YouTube

-Pearson: 0.6881

- Spearman: 0.7003
- RMSE: 0.7070

Interpretation:

The model demonstrates strong linear (Pearson) and monotonic (Spearman) correlations with ground truth scores. The relatively low RMSE indicates moderate prediction errors, suggesting that this dataset is the best fit for the model among the three.

b. LIVE

- Pearson: 0.6478
- Spearman: 0.6547
- RMSE: 0.7694

Interpretation:

Correlations are slightly weaker than on YouTube, and the higher RMSE suggests lower performance on this dataset. This may be due to specific characteristics of LIVE videos (e.g., different distortions or artifacts).

c. Konvid

- Pearson: 0.6735
- Spearman: 0.6744
- RMSE: 0.7230

Interpretation:

Performance is intermediate between YouTube and LIVE. Correlations are slightly lower than YouTube's, but the RMSE is better than LIVE's, indicating a balance between correlation and prediction error.

2. Key Observations

- Best Performance: The model performs best on the YouTube dataset (highest correlations and lowest RMSE).
- Moderate Performance: Konvid shows acceptable but weaker results compared to YouTube.
- Weakest Performance: LIVE has the lowest correlations and highest RMSE, making it the most challenging dataset for this model.

Alternative phrasing for conciseness: The ''VQA-DML'' model achieves optimal results on YouTube, demonstrates moderate performance on Konvid, and faces challenges on LIVE.

4.5 Conclusion

In this chapter, we presented our no-reference video quality prediction (NR-VQA) model based on the MAML algorithm (named: VQA-DML). We used three different datasets: KoNViD-1k, LIVE, and YouTube, each with varying characteristics and quality distributions.

The results showed that our ’’VQA-DML’’ model generalizes well for videos of YouTube database but may require adjustments to perform better on datasets like ’’LIVE’’.

The performance differences across datasets could be explained by variations in video quality, types of distortions, or the distribution of quality scores.

.

General Conclusion

In this work entitled "Video Quality Assessment Based on Deep Meta-Learning", we explored the potential of meta-learning techniques, particularly the Model-Agnostic Meta-Learning (MAML) algorithm, in the field of No-Reference Video Quality Assessment (NR-VQA). The growing demand for automatic and scalable video quality assessment solutions, especially in contexts where the reference video is unavailable, motivated the use of intelligent learning systems capable of adapting rapidly.

Through a rigorous methodology combining feature normalization, task-based learning division, and fast adaptation using nested optimization loops, we developed a model capable of adapting to new video content using only limited training data. Experiments conducted on the LIVE, KoNViD-1k, and YouTube-UGC datasets demonstrated the model's ability to generalize across a variety of contexts, achieving competitive results in terms of prediction accuracy (MAE, RMSE) and correlation (PLCC, SRCC, KRCC) when compared to classical NR-VQA methods such as ENIQA and NBIQA.

This work highlights the advantages of meta-learning by offering a flexible and effective solution for video quality assessment in dynamic and weakly annotated environments. Future work could consider integrating more advanced architectures or multimodal features to further improve the accuracy and robustness of predictions across more diverse video content.

Future Work

Although the MAML algorithm demonstrated good performance in this study, future research could explore alternative meta-learning approaches such as Reptile or Meta-SGD, each offering specific advantages in terms of training speed, stability, or adaptability to new tasks. Furthermore, it would be relevant to explore the notion of "video type" (e.g., gaming, vlog, animation, etc.) as a task unit within the meta-learning framework. Each video category may exhibit specific visual and temporal characteristics that influence perceived quality. By treating each video type as a distinct task, the model could learn adaptation patterns tailored to each content style, thereby improving its generalization capacity in a multi-domain context.

References

[1] Zhang et al. (2021) discuss the evolution of video technologies towards Ultra High Definition (UHD) and High Dynamic Range (HDR), emphasizing the increased data requirements for higher resolutions

[2] https://www.picwand.ai/upscale-images/make-photo-hd/

[3] https://pixabay.com/fr/photos/lever-du-soleil-leau-laube-matin-4987384/

[4] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004)*Image quality assessment: From error visibility to structural similarity.IEEE Transactions on Image Processing, 13*(4), 600–612.https://doi.org/10.1109/TIP.2003.819861

[5] Horé, A., & Ziou, D. (2010).*Image quality metrics: PSNR vs. SSIM.Proceedings of the 20th International Conference on Pattern Recognition (ICPR), 2366–2369*.https://doi.org/10.1109/ICPR.2010.579

[6] Sheikh, H. R., & Bovik, A. C. (2006).*Image information and visual quality.IEEE Transactions on Image Processing, 15*(2), 430–444.https://doi.org/10.1109/TIP.2005.861264

[7] Netflix Technology Blog. (2016).*VMAF: The journey continues*.https://netflixtechblog.com/vmaf-the-journey-continues-8b5a4e2f4b4

[8]International Telecommunication Union. (1996).*Recommendation P.800: Methods for subjective determination of transmission quality*.https://www.itu.int/rec/T-REC-P.800-199608-I/en

[9] Reinders, M. J. T., de Ridder, D., & Duin, R. P. W. (2007).*Systematic comparison of subjective image quality measures.Signal Processing: Image Communication, 22*(6), 516–528.https://doi.org/10.1016/j.image.2007.01.008

[10] Pearlman, W. A., Said, A., Islam, A., & Carper, W. (2005).*A/B testing in video quality evaluation.IEEE Transactions on Circuits and Systems for Video Technology, 15*(1), 1–14.https://doi.org/10.1109/TCSVT.2004.836644

[11] Bjontegaard, G. (2001). Calculation of average PSNR differences between RD-curves. VCEG.

[12] Sullivan, G. J., Ohm, J., Han, W. J., & Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology

[13] Cisco Visual Networking Index (VNI) (2020). Forecast and Trends, 2017–2022. Cisco.

[14] Akhshabi, S., Anantakrishnan, L., Begen, A. C., & Dovrolis, C. (2011). An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP. ACM MMSys

[15] Wang, J., Lin, X., & Chen, Y. (2018). Edge computing-based video analytics for surveillance systems. IEEE Transactions on Industrial Informatics.

[16] Saldanha, C. M., & Frangopoulos, C. A. (2016). QoE modeling for interactive real-time video streaming. Springer.

[17] Barkowsky, M., Le Callet, P., & Tourancheau, B. (2012). Perceptual metrics for video quality assessment and display quality adaptation. IEEE Signal Processing Magazine.

[18] Reinhard, E., Ward, G., Pattanaik, S., & Debevec, P. (2010). High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting. Morgan Kaufmann.

[19] Wang, X., Chan, K. C., Yu, K., Dong, C., & Change Loy, C. (2021). Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. ICCV.

[20] Zhang, K., Zuo, W., & Zhang, L. (2017). FFDNet: Toward a Fast and Flexible Solution for CNN-based Image Denoising. IEEE Transactions on Image Processing.

[21] ITU-R BT.2100 (2018). Image parameter values for high dynamic range television for use in production and international programme exchange.

[22]Dolby Laboratories. (n.d.). *Dolby Vision Technical Overview*. Retrieved June 5, 2025, from https://www.dolby.com/technologies/dolby-vision

[23] Karpenko, A., Jacobs, D. E., Baek, J., & Levoy, M. (2011). Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. Stanford University.

[24] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. DOI: 10.1038/nature14539

[25] Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks, 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

[26] Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. Proceedings of the 34th International Conference on Machine Learning (ICML).

[27] Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-Learning in Neural Networks: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence

[28] Finn et al., 2017“Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” ICML, 2017.

[29] https://medium.com/aimonks/mastering-adaptability-exploring-model-agnostic-meta-learning-maml-996796c890f0

[30] Nichol et al., 2018

[31] https://medium.com/@gayathrisaranathan/tutorial-on-building-an-advanced-rag-application-using-evaluation-and-re-ranking-with-7183fa1bcb38

[32] https://www.mdpi.com/2076-3417/12/11/5543

[33] Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. arXiv preprint arXiv:1707.09835. [Link]

[34] Vinyals, O., Blundell, C., Lillicrap, T., & Kavukcuoglu, K. (2016). Matching Networks for One Shot Learning. NeurIPS. Link

[35] https://rahsoft.com/2021/04/13/understanding-the-need-of-matching-networks/

[36] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical Networks for Few-shot Learning. NeurIPS. Link

[37] https://www.sciencedirect.com/science/article/pii/S2352914824000765

[38] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to Compare: Relation Network for Few-Shot Learning. CVPR. Link

[39] Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. arXiv. Link

[40] https://arxiv.org/html/2312.06141v2

[41] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-Learning with Memory-Augmented Neural Networks. ICML. Link

[42] Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2018). A Simple Neural Attentive Meta-Learner. ICLR. Link

[43] Wang et al., 2020 – ACM Computing Surveys

[44] Maicas et al., 2018 – MICCAI

[45] Duan et al., 2016 – arXiv

[46] Bansal et al., 2020 – NeurIPS

[47] Chen et al., 2020 – ICML (Contrastive learning for visual embeddings)

[48] Antoniou, A., Edwards, H., & Storkey, A. (2019). How to Train Your MAML. ICLR.

[50] Triantafillou, E. et al. (2019). Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. ICLR.

[51] Rusu et al. (2019) – Meta-Learning with Latent Embedding Optimization (ICLR)

[52] Brown et al. (2020) – Language Models are Few-Shot Learners (NeurIPS)

[53] Yao et al. (2021) – Bayesian Meta-Learning with Uncertainty for Few-Shot Learning (NeurIPS)

[54] Beaulieu et al. (2020) – Learning to Continually Learn (ICLR)

[55] Metz et al. (2019) – Understanding and Improving Gradient-Based Meta-Learning (NeurIPS)

[56] Zintgraf et al. (2019) – Fast Context Adaptation via Meta-Learning (ICML)

[57] Ravi & Larochelle, “Optimization as a Model for Few-Shot Learning,” ICLR, 2017.

[58] Tu Z., Wang Y., Birkbeck N., Adsumilli B., Bovik A. C., “UGC-VQA: Benchmarking blind video quality assessment for user generated content,” *IEEE Transactions on Image Processing*, 2021.

[59] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[60] https://courses.lumenlearning.com/suny-natural-resources-biometrics/chapter/chapter-7-correlation-and-simple-linear-regression/

[61] https://www.alexanderdemos.org/Class5.html

[62] https://www.researchgate.net/figure/Architecture-for-a-generalised-regression-neural-network-GRNN-The-input-layer-where_fig2_270274130

[63] https://medium.com/almabetter/decision-trees-a8e03ec50965

[64]:Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7(3), 1247–1250.

[65] Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). No-Reference Image Quality Assessment in the Spatial Domain. IEEE Transactions on Image Processing.

[66] :Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[67]Hastie,T.,Tibshirani,R.,&FriedmanJ(2009).
The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer. DOI: 10.1007/978-0-387-84858-7 : :::

[67] Zhou, J., et al. (2022). Few-shot Learning for Video Quality Assessment via Meta Learning. IEEE ICME.
https://doi.org/10.1109/ICME52920.2022.9859607

[68]Zhang, W., et al. (2020). MetaVQA: Learning to Generate Meta-Knowledge for Video Quality Assessment. ACM MM.
https://doi.org/10.1145/3394171.3413826

[69] Van Rossum, G., & Drake, F. L. (2009). *The Python Language Reference Manual*. Network Theory Ltd.

[70] McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. SciPy Conferenumpy

[71] Harris, C. R., et al. (2020). *Array Programming with NumPy*. Nature, 585(7825), 357–362.

[72] Paszke, A., et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. NeurIPS.

[73] Virtanen, P., et al. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. Nature Methods

[74]Seshadrinathan, K., Soundararajan, R., Bovik, A. C., & Cormack, L. K. (2010).*A subjective study to evaluate video quality assessment algorithms.*IEEE Transactions on Image Processing,

19(6), 1427–1441.https://doi.org/10.1109/TIP.2010.2042649
[75] Hosu, V., Hahn, F., Jenadeleh, M., Lin, H., Men, H., Szirányi, T., & Saupe, D. (2017).*The Konstanz Natural Video Database (KoNViD-1k).*In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–6). IEEE. https://doi.org/10.1109/QoMEX.2017.7965693

[76] Wang, H., Birkbeck, N., Adsumilli, B., & Bovik, A. C. (2019).*YouTube-UGC Dataset for Video Quality Assessment.*In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 161–166). IEEE.https://arxiv.org/abs/1907.06520