Thèse de doctorat

Pour obtenir le grade de Docteur de

l'UNIVERSITE POLYTECHNIQUE HAUTS-DE-FRANCE

et de l'INSA HAUTS-DE-FRANCE

et de l'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Discipline: Électronique, microélectronique, nanoélectronique et micro-ondes

Dépôt final, présenté et soutenu par Yujing ZHANG

Le 23/09/2024, à Valenciennes

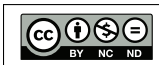
École doctorale :École Doctorale Polytechnique Hauts-de-France (ED PHF n° 635)

Unité de recherche : Institut d'Electronique de Microélectronique et de Nanotechnologie - Site de Valenciennes (IEMN - UMR CNRS 8520)

Décodage par liste de vidéos assisté par apprentissage profond dans des systèmes de transmission vidéo sujets aux erreurs

JURY

Président du jury :	M. Marco PEDERSOLI, Professeur, ÉTS Montréal
Rapporteurs :	M. Hassan RABAH, Professeur, Institut Jean-Lamour, Univ. Lorraine, Nancy Mme. Anissa MOKRAOUI, Professeure, L2TI, Université Paris 13 Nord
Examinatrice :	Mme. Farida CHERIET, Professeure, École Polytechnique de Montréal
Membres invités :	M. Carlos VAZQUEZ, Professeur, ÉTS Montréal M. Patrick CORLAY, Professeur, UPHF Valenciennes
Co-directeurs :	M. Stéphane COULOMBE, Professeur, ÉTS Montréal M. François-Xavier COUDOUX, Professeur, UPHF Valenciennes



This Creative Commons license allows readers to download this work and share it with others as long as the author is credited. The content of this work cannot be modified in any way or used commercially.

PhD Thesis

Submitted for the degree of Doctor of Philosophy from

UNIVERSITE POLYTECHNIQUE HAUTS-DE-FRANCE

And INSA HAUTS-DE-FRANCE

And ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Discipline: Electronics, microelectronics, nanoelectronics and microwaves

The final submission, presented and defended by Yujing ZHANG

On 23/09/2024, Valenciennes

Doctoral school : Doctoral School Polytechnique Hauts-de-France (ED PHF n° 635)

Research unit : Institute of Electronics Microelectronics and Nanotechnology - Valenciennes site (IEMN - UMR CNRS 8520)

Deep learning-assisted video list decoding in error-prone video transmission systems

JURY

President of jury :	M. Marco PEDERSOLI, Professor, ÉTS Montréal
Reviewers :	M. Hassan RABAH, Professor, Institut Jean-Lamour, Univ. Lorraine, Nancy Mme. Anissa MOKRAOUI, Professor, L2TI, Université Paris 13 Nord
Examiner :	Mme. Farida CHERIET, Professor, École Polytechnique de Montréal
Invited members :	M. Carlos VAZQUEZ, Professor, ÉTS Montréal M. Patrick CORLAY, Professor, UPHF Valenciennes
Co-directors :	M. Stéphane COULOMBE, Professor, ÉTS Montréal M. François-Xavier COUDOUX, Professor, UPHF Valenciennes

REMERCIEMENTS

Je profite de ce manuscrit pour exprimer ma gratitude envers mes directeurs, les Professeurs Stéphane Coulombe, François-Xavier Coudoux et Patrick Corlay. Leurs expertises et conseils m'ont permis de développer mes connaissances et parfaire la maitrise de certains sujets liés à l'évaluation de la qualité d'image et à l'intelligence artificielle. Ils ont toujours su apporter une vision pertinente sur mes problématiques tout en me laissant une grande autonomie.

Je remercie également les Professeurs Marco Pedersoli, Hassan Rabah, Anissa Mokraoui, Farida Cheriet et Carlos Vazquez d'avoir accepté de faire partie de mon jury de thèse.

Je tiens également à remercier Vivien Boussard, pour le temps qu'il m'a consacré lors de mes débuts sur ce projet, et Alexis Guichemerre, pour le temps qu'il a consacré à discuter de certains aspects du projet ainsi que son aide pour les simulations. En effet, sa compréhension du problème et des méthodes utilisées m'ont été très précieuses pour la réussite de ce projet.

J'aimerais aussi souligner la contribution du Professeur Anthony Trioux pour son temps et le partage de son expertise qu'il m'a accordé afin de permettre l'avancement de ce projet.

Je souhaiterais également remercier ma famille pour le soutien dont ils ont toujours fait preuve dans le cadre de mes études.

Pour finir, je souhaite remercier le Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG) ainsi que l'Université Polytechnique Hauts-de-France (UPHF) et la Région Hauts-de-France pour leur soutien financier.

Décodage par liste de vidéos assisté par apprentissage profond dans des systèmes de transmission vidéo sujets aux erreurs

Yujing ZHANG

RÉSUMÉ

Au cours des dernières années, les applications vidéo ont connu un développement rapide. Par ailleurs, l'expérience en matière de qualité vidéo s'est considérablement améliorée grâce à l'avènement de la vidéo HD et à l'émergence des contenus 4K. En conséquence, les flux vidéo ont tendance à représenter une plus grande quantité de données. Pour réduire la taille de ces flux vidéo, de nouvelles solutions de compression vidéo telles que HEVC ont été développées.

Cependant, les erreurs de transmission susceptibles de survenir sur les réseaux peuvent provoquer des artefacts visuels indésirables qui dégradent considérablement l'expérience utilisateur. Diverses approches ont été proposées dans la littérature pour trouver des solutions efficaces et peu complexes afin de réparer les paquets vidéo contenant des erreurs binaires, en évitant ainsi une retransmission coûteuse et incompatible avec les contraintes de faible latence de nombreuses applications émergentes (vidéo immersive, télé-opération). La correction d'erreurs basée sur le contrôle de redondance cyclique (CRC) est une approche prometteuse qui utilise des informations facilement disponibles sans surcoût de débit. Cependant, elle ne peut corriger en pratique qu'un nombre limité d'erreurs. Selon le polynôme générateur utilisé, la taille des paquets et le nombre maximum d'erreurs considéré, cette méthode peut conduire non pas à un paquet corrigé unique, mais plutôt à une liste de paquets possiblement corrigés. Dans ce cas, le décodage de liste devient pertinent en combinaison avec la correction d'erreurs basée CRC ainsi qu'avec les méthodes exploitant l'information sur la fiabilité des bits reçus. Celui-ci présente toutefois des inconvénients en termes de sélection de vidéos candidates. Suite à la génération des candidats classés lors du processus de décodage de liste dans l'état de l'art, la sélection finale considéra souvent le premier candidat valide dans la liste finale comme vidéo reconstruite. Cependant, cette sélection simple est arbitraire et non optimale, la séquence vidéo candidate en tête de liste n'étant pas nécessairement celle qui présente la meilleure qualité visuelle. Il est donc nécessaire de développer une nouvelle méthode permettant de sélectionner automatiquement la vidéo ayant la plus haute qualité dans la liste des candidats.

Nous proposons de sélectionner le meilleur candidat en fonction de la qualité visuelle déterminée par un système d'apprentissage profond (DL). Considérant que la distorsion sera gérée sur chaque image, nous considérons l'évaluation de la qualité de l'image plutôt que l'évaluation de la qualité vidéo. Plus précisément, chaque candidat subit un traitement par une méthode d'évaluation de la qualité d'image (image quality assessment, IQA) sans référence basée sur l'apprentissage profond pour obtenir un score. Par la suite, le système sélectionne le candidat ayant le score IQA le plus élevé. Pour cela, notre système évalue la qualité des vidéos soumises à des erreurs de transmission sans éliminer les paquets perdus ni dissimuler les régions perdues. Les distorsions causées par les erreurs de transmission diffèrent de celles prises en compte par les mesures de qualité visuelle traditionnelles, qui traitent généralement des distorsions globales

et uniformes de l'image. Ainsi, ces métriques ne parviennent pas à distinguer la version corrigée des différentes versions vidéo corrompues lorsque les erreurs sont locales et non-uniformes. Notre approche revisite et optimise la technique de décodage de liste classique en lui associant une architecture CNN d'abord, puis Transformer pour évaluer la qualité visuelle et identifier le meilleur candidat. Elle est sans précédent et offre d'excellentes performances. En particulier, nous montrons que lorsque les erreurs de transmission se produisent dans une trame intra, nos architectures basées sur CNN et Transformer atteignent une précision de décision de 100%. Pour les erreurs dans une image inter, la précision est de 93% et 96%, respectivement.

Mots-clés: Transmission vidéo, décodage par liste, correction d'erreur vidéo basée sur contrôle de redondance cyclique, distorsions non uniformes, évaluation de la qualité de l'image, réseau neuronal convolutif, transformeur de vision

Deep learning-assisted video list decoding in error-prone video transmission systems

Yujing ZHANG

ABSTRACT

In recent years, video applications have developed rapidly. At the same time, the video quality experience has improved considerably with the advent of HD video and the emergence of 4K content. As a result, video streams tend to represent a larger amount of data. To reduce the size of these video streams, new video compression solutions such as HEVC have been developed.

However, transmission errors that may occur over networks can cause unwanted visual artifacts that significantly degrade the user experience. Various approaches have been proposed in the literature to find efficient and low-complexity solutions to repair video packets containing binary errors, thus avoiding costly retransmission that is incompatible with the low latency constraints of many emerging applications (immersive video, tele-operation). Error correction based on cyclic redundancy check (CRC) is a promising approach that uses readily available information without throughput overhead. However, in practice it can only correct a limited number of errors. Depending on the generating polynomial used, the size of the packets and the maximum number of errors considered, this method can lead not to a single corrected packet but rather to a list of possibly corrected packets. In this case, list decoding becomes relevant in combination with CRC-based error correction as well as methods exploiting information on the reliability of the received bits. However, this has disadvantages in terms of selection of candidate videos. Following the generation of ranked candidates during the state-of-the-art list decoding process, the final selection often considers the first valid candidate in the final list as the reconstructed video. However, this simple selection is arbitrary and not optimal, the candidate video sequence at the top of the list is not necessarily the one which presents the best visual quality. It is therefore necessary to develop a new method to automatically select the video with the highest quality from the list of candidates.

We propose to select the best candidate based on the visual quality determined by a deep learning (DL) system. Considering that distortions will be assessed on each frame, we consider image quality assessment rather than video quality assessment. More specifically, each candidate undergoes processing by a reference-free image quality assessment (IQA) method based on deep learning to obtain a score. Subsequently, the system selects the candidate with the highest IQA score. To do this, our system evaluates the quality of videos subject to transmission errors without eliminating lost packets or concealing lost regions. Distortions caused by transmission errors differ from those accounted for by traditional visual quality measures, which typically deal with global, uniform image distortions. Thus, these metrics fail to distinguish the repaired version from different corrupted video versions when local, non-uniform errors occur. Our approach revisits and optimizes the classic list decoding technique by associating it with a CNN architecture first, then with a Transformer to evaluate the visual quality and identify the best candidate. It is unprecedented and offers excellent performance. In particular, we show that when transmission errors occur within an intra frame, our CNN and Transformer-based

architectures achieve 100% decision accuracy. For errors in an inter frame, the accuracy is 93% and 96%, respectively.

Keywords: Video transmission, List decoding, Cyclic Redundancy Check based video error correction, Non-uniform distortions, Image quality assessment, Convolutional neural network, Vision Transformer

Décodage par liste de vidéos assisté par apprentissage profond dans des systèmes de transmission vidéo sujets aux erreurs

Yujing ZHANG

RÉSUMÉ

Introduction

Au cours des dernières années, on a assisté à un développement extrêmement rapide des appareils, des systèmes et des applications vidéo. Cette croissance de la vidéo ne fera que s'intensifier dans les années à venir, car le contenu vidéo représente déjà près de 80% du trafic internet actuel (Laghari *et al.*, 2023; Systems, 2016). La vidéo en temps réel est de plus en plus populaire et la transmission de contenu vidéo constitue la principale catégorie de données transmises dans le monde aujourd'hui. Dans le contexte de l'émergence de l'Internet of Things (IoT), les applications visées sont de plus en plus nombreuses pour lesquelles l'information visuelle peut considérablement enrichir la connaissance de l'environnement : systèmes vidéo pour la télésurveillance et le contrôle des machines, dispositifs d'e-santé, réalité virtuelle et augmentée. Les systèmes de transport intelligents (Intelligent Transport Systems (ITS)) sont également directement concernés ; la vidéo permet de communiquer des informations sur l'environnement de conduite ou l'état du réseau de transport entre les véhicules et l'infrastructure.

En outre, la qualité de l'expérience vidéo s'est considérablement améliorée ces dernières années, grâce à l'avènement de la vidéo haute définition (high definition (HD)) et à l'émergence de contenus 4K. Par conséquent, les flux vidéo ont tendance à représenter une plus grande quantité de données. Pour réduire considérablement la taille de ces flux vidéo, de nouvelles solutions de compression vidéo ont été développées (Sullivan & Wiegand, 2005; Wiegand *et al.*, 2003; Sullivan *et al.*, 2012).

Toutefois, des erreurs de transmission se produisent sur des réseaux peu fiables et sujets aux erreurs, tels que les réseaux de capteurs et les objets Internet vidéo (WiFi (IEEE, 2016), BLE (Collotta *et al.*, 2018), etc.). Ces erreurs peuvent dégrader considérablement l'expérience de l'utilisateur en provoquant des distorsions indésirables telles que des flous, des motifs géométriques ou des effets d'écran vert. Une retransmission des paquets erronés peut être demandée mais cela réduit également l'efficacité du réseau. Outre le fait qu'elles prennent beaucoup de temps et de ressources, ces retransmissions peuvent être incompatibles avec certains domaines d'application, tels que les applications de transport et de vidéo immersive, où les informations vidéo doivent arriver de manière fiable et en temps réel avec un temps de latence très faible. Dans ces domaines d'application, il est préférable de conserver les paquets erronés, même s'ils peuvent entraîner des artefacts visuels, plutôt que de les rejeter et d'en demander de nouveaux.

Diverses approches ont été proposées dans la littérature afin de trouver des solutions efficaces et peu complexes pour réparer les paquets vidéo contenant des erreurs de bits. Les méthodes

de dissimulation et de correction des erreurs sont les deux principales catégories d'approches utilisées pour traiter les paquets endommagés. La dissimulation d'erreurs (Shirani *et al.*, 1999; Wang & Zhu, 1998) est une technologie appliquée du côté du décodeur pour régénérer les informations perdues dans le flux vidéo décodé (c'est-à-dire tenter de reconstruire les parties qui ont été endommagées pendant le transport et mises au rebut). D'autre part, la correction des erreurs implique l'identification et la correction des bits erronés dans un paquet à l'aide de diverses stratégies (Jokela & Lehtonen, 2007; Balatsoukas-Stimming *et al.*, 2015; Boussard *et al.*, 2020a). Étant donné que le contrôle de redondance cyclique (Cyclic Redundancy Check (CRC)) est déjà largement utilisé dans les communications IP (par exemple dans les paquets UDP et TCP) et accessible à la couche application, la correction d'erreurs basée sur CRC est une approche prometteuse. Elle utilise des informations facilement disponibles et n'ajoute pas de surcharge. Toutefois, dans la pratique, la correction d'erreurs basée sur le CRC ne peut corriger qu'un nombre limité d'erreurs. En effet, en fonction du polynôme générateur, de la taille du paquet et du nombre maximal d'erreurs considérées, la méthode peut ne pas aboutir à un seul candidat permettant de corriger le paquet, mais plutôt à une liste de candidats parmi lesquels il n'est pas possible, sans information additionnelle, d'identifier le paquet corrigé. C'est là que le décodage par liste devient pertinent en combinaison avec la correction d'erreurs basée sur le CRC ainsi que les méthodes qui tirent parti de la fiabilité des bits reçus.

Dans le décodage par liste, un nombre limité de candidats potentiels est présenté séquentiellement au décodeur jusqu'à ce qu'une vidéo valide soit générée (c'est-à-dire sans générer d'erreurs de syntaxe ou faire planter le décodeur). Chaque candidat est une variante du paquet reçu, dont un ou plusieurs bits ont été modifiés pour tenter de le corriger. Les candidats sont généralement classés et présentés au décodeur du plus probable au moins probable (Balatsoukas-Stimming *et al.*, 2015; Caron & Coulombe, 2015; Golaghazadeh *et al.*, 2018). Dans le cas de la correction d'erreurs basée sur le CRC, de nombreux candidats peuvent être décodés sans erreur et le premier n'est pas nécessairement celui associé au paquet corrigé. Par conséquent, il est hautement souhaitable de développer un système capable d'évaluer la qualité visuelle des candidats décodables afin de sélectionner le meilleur d'entre eux.

Par conséquent, l'idéal pour sélectionner la version n'est pas de prendre le premier candidat décodable, mais d'évaluer la qualité visuelle de chaque candidat. Comme la séquence vidéo originale n'est pas disponible, nous devons explorer diverses approches d'évaluation de la qualité visuelle sans référence (No-Reference (NR)). Étant donné que la gestion des distortions s'effectue sur chaque image, nous considérerons l'évaluation de la qualité d'image (Image Quality Assessment (IQA)) plutôt que l'évaluation de la qualité vidéo (Video Quality Assessment (VQA)). Nous proposons d'utiliser une méthode d'évaluation sans référence de la qualité d'image basée sur l'apprentissage profond pour sélectionner la meilleure vidéo candidate avec le score de qualité le plus élevé.

Ce système évalue la qualité des vidéos soumises à des erreurs de transmission sans rejeter les paquets perdus ni dissimuler les régions perdues. Les distorsions causées par les erreurs de transmission diffèrent considérablement de celles prises en compte par les mesures traditionnelles de qualité d'image, qui traitent généralement des distorsions globales et uniformes de l'image.

Par conséquent, ces mesures ne parviennent pas à distinguer la version corrigée des diverses versions corrompues de l'image. Notre approche globale, qui combine des techniques de décodage de liste traditionnelles, mais revisitées, avec une architecture d'apprentissage profond pour évaluer la qualité visuelle et identifier le meilleur candidat, est sans précédent et offre d'excellentes performances.

L'objectif principal de ce projet de recherche est de développer, dans le contexte du décodage de liste vidéo pour les transmissions vidéo sujettes aux erreurs, une nouvelle méthode pour sélectionner automatiquement la vidéo de meilleure qualité à partir d'une liste de candidats générée par une méthode de correction telle que la correction d'erreurs basée sur le CRC. Nous visons à sélectionner le meilleur candidat sur la base de la qualité visuelle déterminée par un système d'apprentissage profond (Deep Learning (DL)). En supposant que l'erreur sera évaluée sur chaque image individuellement, nous considérons l'évaluation de la qualité de l'image plutôt que l'évaluation de la qualité de la vidéo. Plus précisément, chaque candidat sera traité par un système d'apprentissage profond NR IQA pour obtenir un score. Ensuite, le système sélectionne le candidat ayant le score IQA le plus élevé. Les principales contributions de notre recherche sont donc les suivantes :

- Nous proposons tout d'abord un cadre de décodage de liste vidéo assisté par réseaux de neurones convolutifs (Convolutional Neural Network (CNN)) afin d'identifier le candidat de meilleure qualité dans la liste de candidats décodables. Nous développons une mesure CNN basée sur IQA pour évaluer la qualité vidéo image par image sans tenir compte de la relation temporelle entre les images. Nous utilisons la structure proposée dans (Kang *et al.*, 2014) comme architecture de base de notre méthode. Le réseau original est conçu pour évaluer la qualité de l'image avec des distorsions locales, et nous utilisons des scores de fidélité au niveau de petites régions d'images, appelées patchs, et proposons une **normalisation locale améliorée** pour évaluer la qualité de l'image avec des distorsions non uniformes causées par des erreurs de transmission.
- Nous proposons ensuite le **premier cadre de décodage de liste vidéo assisté par transformeur** pour les systèmes de transmission vidéo sujets aux erreurs. Ce cadre identifie le candidat ayant la meilleure qualité visuelle parmi les multiples options générées au cours du processus de décodage de liste sur les réseaux non fiables, en veillant à ce que le décodage final soit le plus efficace possible en termes de qualité visuelle.
- Nous améliorons le cadre assisté proposé par DL en ajoutant les éléments suivants :

  a) Une fonction **Discriminant Color Texture Transformation (DCTT)** pour faire la distinction entre un patch uniforme bien reçu et un patch d'erreur initialisé comme un patch vert uniforme par le décodeur.

  b) Une fonction **Ranking-Constrained Penalty Loss function (RCPL)** est proposée pour pénaliser davantage les cas où un patch endommagé obtient un score plus élevé qu'un patch intact, ce qui est particulièrement important pour les images inter comportant des erreurs subtiles. Le cadre avancé proposé est conçu pour évaluer la qualité de l'image avec des distorsions locales. En d'autres termes, il est sensible à la détection des distorsions spatiales non uniformes causées par les erreurs de transmission. En ajoutant deux nouveaux

composants, notre schéma de décodage est nettement plus performant pour les images codées en intra et codées en inter.
- Nous améliorons le système en remplaçant le composant CNN du cadre de décodage de liste vidéo assisté DL proposé par un composant basé sur un transformeur. Cette méthode IQA basée sur transformeur utilise la structure proposée dans (Yang *et al.*, 2022) comme architecture de base, et nous proposons une nouvelle méthode appelée **Neighborhood-based Patch Fidelity Aggregation (NPFA)** pour mieux prendre en compte les discontinuités locales aux limites horizontales et verticales des blocs codés entre les patchs voisins.
- Nous construisons également une base de données utilisant la compression High Efficiency Video Coding (HEVC) sur des séquences vidéo YUV originales collectées à partir d'ensembles de données publiques (xip; Wang *et al.*, 2016; Pinson, 2013). La plupart des ensembles de données existants pour l'évaluation de la qualité de l'image se concentrent sur les pertes synthétisées artificiellement ou les pertes générées par l'utilisateur, mais n'incluent pas les différents types de distorsions non uniformes causées par les erreurs de transmission. Par conséquent, nous créons les scripts et les instructions pour régénérer une base de données, suivant la norme HEVC (Sze & al, 2014) et l'ajout d'erreurs de transmission pour obtenir des trames corrompues non uniformes. Des modèles d'erreur simples sont appliqués aux paquets vidéo encodés pour simuler l'effet de la transmission sur des réseaux sujets aux erreurs. Nous collectons la combinaison de $p \times p$ patchs, que nous appelons « super-patch », dans les images corrompues de ces flux vidéo décodés, pour effectuer l'entraînement et les tests avec les scores d'agrégation de la fidélité des patchs basés sur le voisinage.

Ce manuscrit est organisé comme suit. Dans le chapitre 1, nous passons en revue les méthodes utilisées dans la littérature pour le traitement des erreurs dans les communications vidéo, depuis la dissimulation des erreurs jusqu'à la correction proprement dite d'un paquet corrompu. Nous examinons également les méthodes VQA et tirons parti de leurs points forts pour identifier une stratégie permettant d'améliorer les performances. Les autres chapitres présentent les cadres de décodage de liste vidéo assistés par apprentissage profond proposés, notamment la méthode basée sur le CNN au chapitre 2 et la méthode basée sur le transformeur au chapitre 3. Les résultats expérimentaux et l'analyse des performances sont présentés au chapitre 4. Enfin, nous concluons la thèse.

Revue de la littérature

Dans ce chapitre, nous présentons le contexte des méthodes IQA présentées dans les chapitres suivants. Nous présentons d'abord différentes méthodes de gestion des erreurs dans les communications vidéo, notamment la dissimulation des erreurs vidéo et la correction des erreurs vidéo. Nous donnons également un aperçu détaillé des systèmes traditionnels de décodage de liste et soulignons leurs inconvénients, ce qui met en évidence la nécessité de l'approche assistée par apprentissage profond que nous proposons.

La dissimulation des erreurs est une technique utilisée du côté du décodeur pour régénérer les informations perdues dans le flux vidéo décodé, dans le but de reconstruire les parties

endommagées pendant la transmission et rejetées par la suite. Cette méthode tire parti de la corrélation des régions adjacentes dans l'image actuelle (dissimulation spatiale (Koloda *et al.*, 2013b)), des images reçues précédemment (dissimulation temporelle (Peng *et al.*, 2002)), ou d'une combinaison des deux (dissimulation spatio-temporelle (Kung *et al.*, 2006)) pour effectuer l'interpolation à partir des valeurs voisines dans l'espace et dans le temps afin de récupérer les zones perdues.

Plutôt que de dissimuler les paquets erronés, les méthodes de correction d'erreur permettent de corriger les paquets vidéo. Par exemple, les méthodes de correction d'erreurs au niveau du flux de bits visent à récupérer le paquet initialement transmis, ce qui est généralement effectué dans les couches inférieures de la pile de protocoles. Les méthodes de correction d'erreurs assistées par le protocole sont basées sur les caractéristiques du protocole utilisé pour transmettre les paquets vidéo et les informations supplémentaires, afin de reconstruire le paquet intact sans erreur. Ces approches sont principalement utilisées dans les couches de liaison et de transport. Dans un travail récent (Boussard *et al.*, 2021a), Boussard et al. ont proposé une méthode de correction vidéo inter-couches basée sur CRC, qui est efficace pour corriger les paquets vidéo déformés après la transmission sur des réseaux sujets aux erreurs.

Les méthodes de décodage par liste exploitent les paquets reçus corrompus pour générer plusieurs paquets transmis candidats à partir du paquet reçu corrompu. Sur la base de ces considérations, plusieurs chercheurs en décodage par liste tentent de trouver des solutions au problème de la correction des erreurs vidéo qui peuvent être intégrées de manière réaliste dans les systèmes de communication mobile actuels et de la prochaine génération.

La deuxième étape du module de décodage de liste, illustrée à la Figure 0.1(a), consiste à filtrer la liste de candidats, liste qui peut être volumineuse. Pour ce faire, des informations supplémentaires sont utilisées. Par exemple, le filtrage par somme de contrôle (Golaghazadeh *et al.*, 2018) utilise la somme de contrôle du *User Datagram Protocol (UDP)* côté récepteur pour éliminer les paquets dont la somme de contrôle indique une erreur. De même, la validation CRC est utilisée dans les approches Log-Likelihood Ratio (LLR) (Balatsoukas-Stimming *et al.*, 2015; Caron & Coulombe, 2015). La troisième étape du module de décodage de liste consiste à valider tous les paquets vidéo candidats par le biais de plusieurs opérations de décodage vidéo. Cela permet d'éliminer tout candidat syntaxiquement incorrect. Par la suite, plusieurs vidéos peuvent subsister, ce qui conduit à l'étape finale de sélection de la vidéo candidate de meilleure qualité en tant que vidéo reconstruite finale.

Le décodage traditionnel par liste présente des inconvénients pour la sélection des vidéos candidates. Comme l'illustre la Figure 0.1(b), la sélection finale des candidats dans ces méthodes est déterminée par le choix du premier candidat valide de la liste classée finale, au lieu de considérer la liste entière pour une évaluation plus complète. Bien que ce choix direct puisse sembler attrayant, il ne s'agit pas d'un processus rigoureux. La séquence vidéo en tête de liste peut ne pas avoir la meilleure qualité de reconstruction.

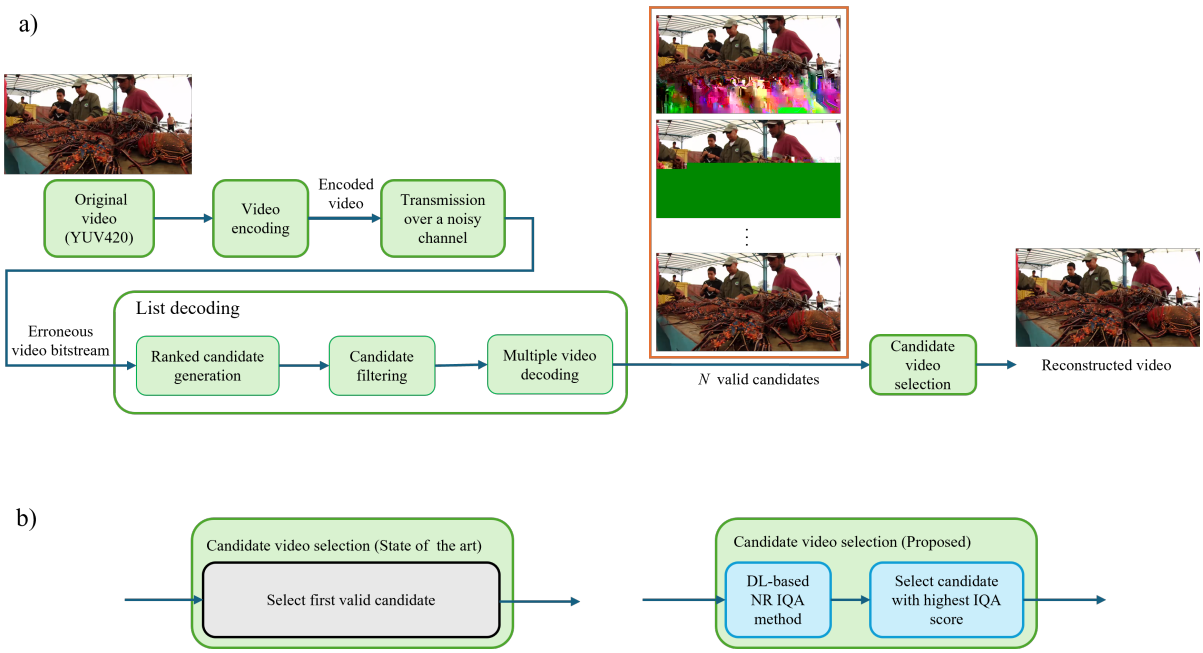


Figure 0.1 Le processus traditionnel de décodage de liste : a) processus général de décodage de liste ; b) différents critères de sélection de vidéos candidates : à gauche, critères de l'état de l'art ; à droite, le critère proposé basé sur l'apprentissage profond.

Il est donc nécessaire de développer une nouvelle méthode pour sélectionner automatiquement la vidéo de meilleure qualité dans la liste des candidats. Nous visons à atteindre cet objectif en utilisant un système d'apprentissage profond (DL) pour déterminer le meilleur candidat sur la base de la qualité visuelle.

Nous envisageons d'utiliser des méthodes objectives d'évaluation de la qualité vidéo pour évaluer la qualité des candidats. Dans la deuxième partie, nous examinons les méthodes objectives d'évaluation de la qualité vidéo, y compris les approches traditionnelles et celles basées sur l'apprentissage profond. Nous introduisons également les connaissances de base liées aux modèles d'apprentissage profond et discutons de l'application des architectures d'apprentissage profond dans l'évaluation de la qualité vidéo. En examinant ces méthodes et en soulignant leurs inconvénients, nous mettons en évidence la nécessité de la métrique assistée par transformeur, qui est bien adaptée à notre système de décodage de liste vidéo.

Bien que les méthodes avec référence puissent évaluer plus précisément la qualité vidéo, elles exigent que la vidéo originale soit disponible. Dans un contexte réel de transmission vidéo, la vidéo originale n'est pas disponible du côté de la réception, ce qui empêche l'utilisation de mesures de qualité vidéo avec référence. C'est pourquoi nos recherches se concentrent davantage sur les méthodes sans référence, NR IQA.

Plusieurs méthodes traditionnelles NR IQA, telles que BRISQUE (Mittal *et al.*, 2012), NIQE (Mittal *et al.*, 2013) et PIQE (Venkatanath *et al.*, 2015), utilisent des statistiques de scènes naturelles ou des caractéristiques basées sur la perception de vidéos naturelles pour évaluer la qualité de l'image en aveugle. Ces mesures NR sont efficaces pour évaluer la qualité des images soumises à des distorsions uniformes. Mais notre objectif est d'évaluer la qualité de l'image en cas de distorsions non uniformément réparties causées par des erreurs de transmission telles que celles illustrées dans les *N candidats valides* de la Figure 0.1(a). Malheureusement, comme nous le montrerons dans les résultats expérimentaux, ces mesures traditionnelles existantes ne permettent pas d'évaluer correctement la qualité de l'image en cas de distorsions non uniformément réparties causées par des erreurs binaires.

Avec le développement continu des technologies liées à l'intelligence artificielle, de plus en plus de recherches se concentrent sur l'application de l'apprentissage profond à l'évaluation de la qualité des images (Vega *et al.*, 2017). S'appuyant sur la capacité des réseaux neuronaux d'apprentissage profond à traiter les images, de nombreuses solutions d'évaluation d'images basées sur les technologies d'apprentissage profond ont vu le jour ces dernières années. Plusieurs études ont été proposées pour appliquer CNN à l'évaluation de la qualité des images (Kang *et al.*, 2014; Bosse *et al.*, 2016; Zhang *et al.*, 2020; Kossi *et al.*, 2022). Ces modèles d'apprentissage profond sont plus performants que les modèles traditionnels dans l'évaluation de la qualité des images. Par exemple, les auteurs de (Kang *et al.*, 2014) ont proposé une approche basée sur les patchs où tous les patchs de l'image se voient attribuer le même niveau de qualité que l'image entière lors de l'apprentissage. Dans (Bosse *et al.*, 2016), les auteurs ont proposé une approche axée sur les données s'appuyant aux une architecture CNN, où les caractéristiques et les statistiques de la scène naturelle sont apprises à partir des données et combinées avec la mise en commun et la régression dans un cadre unique. Un autre modèle bilinéaire profond est présenté dans (Zhang *et al.*, 2020), qui gère à la fois les distorsions synthétiques et authentiques. Ces mesures permettent d'utiliser des bases de données plus importantes pour la simulation et davantage de types d'images erronées peuvent être évalués sans référence. Cependant, ces méthodes ont tendance à évaluer la qualité globale de l'image entière plutôt que la qualité locale, négligeant souvent les distorsions locales dans des régions spécifiques. Ceci les rend inadaptées à notre problème sans, au moins, un réentraînement sur une base de données visuelle contenant des vidéos représentatives de celles décodées après des erreurs de bits.

Avec le grand nombre d'applications et le développement rapide du modèle de transformeur (Vaswani *et al.*, 2017; Dosovitskiy *et al.*, 2020) dans le domaine du traitement des images, de nombreuses méthodes d'évaluation de la qualité des images basées sur le modèle de transformeur sont apparues récemment (You & Korhonen, 2021; Cheon *et al.*, 2021; Golestaneh *et al.*, 2021; Chen *et al.*, 2020a; Yang *et al.*, 2022; Xu *et al.*, 2023). Les modèles basés sur les transformeurs divisent toujours l'image entière en plusieurs petits patchs, puis les aplatissent (*flatten*) et les introduisent dans le codeur du transformeur, afin qu'il puisse apprendre l'attention de ces patchs et évaluer la qualité de l'image. En utilisant un mécanisme d'attention pour calculer rapidement l'importance et la relation entre les patchs, ces méthodes améliorent l'efficacité du traitement de grandes quantités de données d'image et de l'évaluation de la qualité de l'image. Dans (You & Korhonen, 2021), les auteurs ont proposé une architecture consistant à

utiliser un encodeur transformeur peu profond au-dessus d'une carte de caractéristiques extraite par CNN. Les auteurs de (Xu *et al.*, 2023) ont proposé un extracteur de distorsion locale pour obtenir des caractéristiques de distorsion locale à partir d'un CNN pré-entraîné et un injecteur de distorsion locale pour injecter les caractéristiques de distorsion locale dans ViT (Dosovitskiy *et al.*, 2020). Les expériences présentées dans ces articles démontrent également que les résultats de l'évaluation de ce modèle sont plus cohérents avec la perception visuelle humaine.

Système proposé de décodage de liste vidéo assisté par CNN

Comme nous l'avons mentionné précédemment, le cadre traditionnel de décodage de liste qui sélectionne le premier candidat valide n'est pas une méthode idéale. Il est nécessaire de développer une nouvelle méthode pour sélectionner automatiquement la vidéo de meilleure qualité dans la liste des candidats. Nous visons à atteindre cet objectif en utilisant un système d'apprentissage profond (DL) pour déterminer le meilleur candidat sur la base de la qualité visuelle. Étant donné que les erreurs seront traitées par image, nous nous concentrerons sur l'évaluation de la qualité de l'image, et non de la qualité de la vidéo. Plus précisément, chaque candidat sera évalué par une méthode DL basée sur NR IQA pour obtenir un score. Le système sélectionnera ensuite le candidat ayant le score IQA le plus élevé.

Inspirés par les travaux de la littérature, nous avons d'abord envisagé une méthode basée sur CNN. Nous avons proposé une métrique d'estimation de la qualité d'image basée sur CNN appliquée à un système de décodage de liste de vidéos assisté par apprentissage profond. Le processus utilisant notre système comprend la conversion d'image (du format YUV au format utilisé pendant l'apprentissage), la génération de patchs et l'apprentissage du réseau de neurones supervisée par un score de fidélité au niveau du patch. Le choix du meilleur candidat se fait en identifiant le candidat ayant la meilleure qualité. Nous choisissons l'architecture CNN proposée dans (Kang *et al.*, 2014) comme architecture de base et l'améliorons en fonction de notre objectif. Nous proposons d'utiliser des scores locaux afin que chaque patch ait son propre score de qualité. Cela peut aider le réseau de neurones à apprendre plus efficacement les distorsions locales. L'architecture d'origine est un réseau de neurones à 5 couches. Au lieu d'effectuer la simulation uniquement sur le canal de luminance comme dans (Kang *et al.*, 2014), nous étendons nos expériences à trois canaux YUV ou RGB. Notre architecture CNN proposée utilise également une méthode améliorée de normalisation du contraste local.

En considérant un seul canal, par exemple Y, on ne peut pas faire la distinction entre un patch uniforme bien reçu dont la valeur est normalisée à 0 et un patch erroné qui est uniforme car initialisé à 0 par le décodeur. Cette situation est problématique lorsqu'elle se produit dans la base de données d'apprentissage, car le réseau de neurones devient confus lors de l'apprentissage. En effet, après normalisation, un patch uniforme et un patch erroné deviennent identiques et entrent dans les couches CNN avec des scores de référence différents pour l'apprentissage.

Pour éviter ce problème, nous améliorons la normalisation locale en séparant ces deux situations. Lorsque nous détectons que l'écart type est de 0 dans les patchs d'entrée, nous calculons la

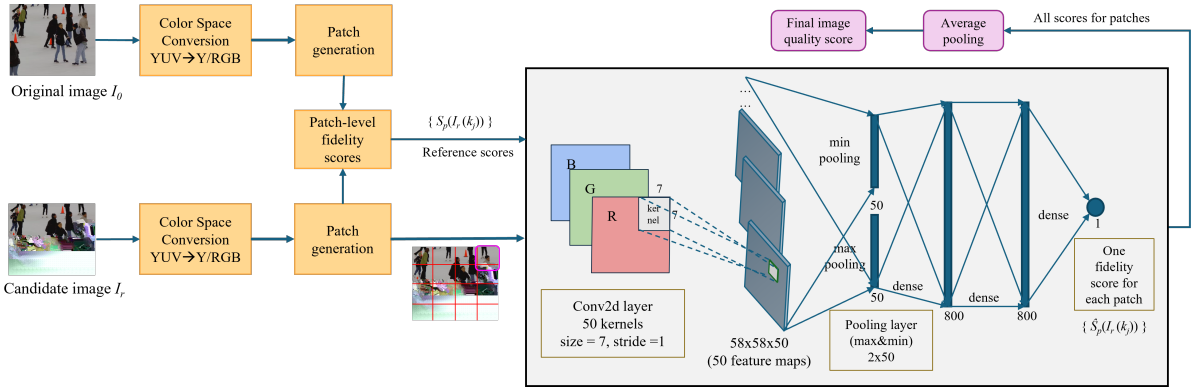


Figure 0.2 La mesure IQA basée sur CNN proposée précédemment pour classer les candidats vidéo dans le décodage de liste. L'image d'origine n'est utilisée que pendant la formation.

valeur moyenne. Si la moyenne n'est pas de 0, nous forçons la valeur de normalisation à être égale à une petite valeur epsilon après normalisation. Nous utilisons cette approche sur chaque canal d'une image YUV ou RGB.

Notre réseau de neurones est d'abord formé sur des patchs non superposés de 64×64 pixels, correspondant à un Coding Tree Unit (CTU) en HEVC. Pour l'entraînement, nous calculons pour chaque patch le score de qualité PSNR, qui est calculé entre le patch corrompu et le patch correspondant dans l'image d'origine avant l'encodage. Pour les tests, nous utilisons la moyenne des scores de patch prédits pour chaque image afin d'obtenir le score de qualité au niveau de l'image.

Système proposé de décodage de liste vidéo assisté par transformeur

Nous avons constaté que le système proposé, assisté par CNN, présente encore certaines limites, comme l'incapacité à détecter les discontinuités entre les blocs CTU voisins lorsque la taille du patch formé est la même que la taille du CTU de codage, et qu'il reste encore à faire pour améliorer la précision des images codées en inter. Nous souhaitons donc créer un système plus avancé, capable d'analyser plus efficacement les distorsions localisées résultant de la propagation d'erreurs dans le voisinage d'un bloc donné, de mieux distinguer un patch uniforme bien reçu d'un patch erroné qui semble uniforme, et d'améliorer les performances des images codées en inter.

Nous proposons une version avancée basée sur un transformeur au lieu du schéma assisté par CNN proposé auparavant. Ce cadre utilise une métrique sans référence basée sur l'architecture transformeur pour évaluer la qualité de l'image candidate. Notre métrique est basée sur MANIQA (Yang *et al.*, 2022), visant à estimer la qualité de l'image pour notre cas spécifique.

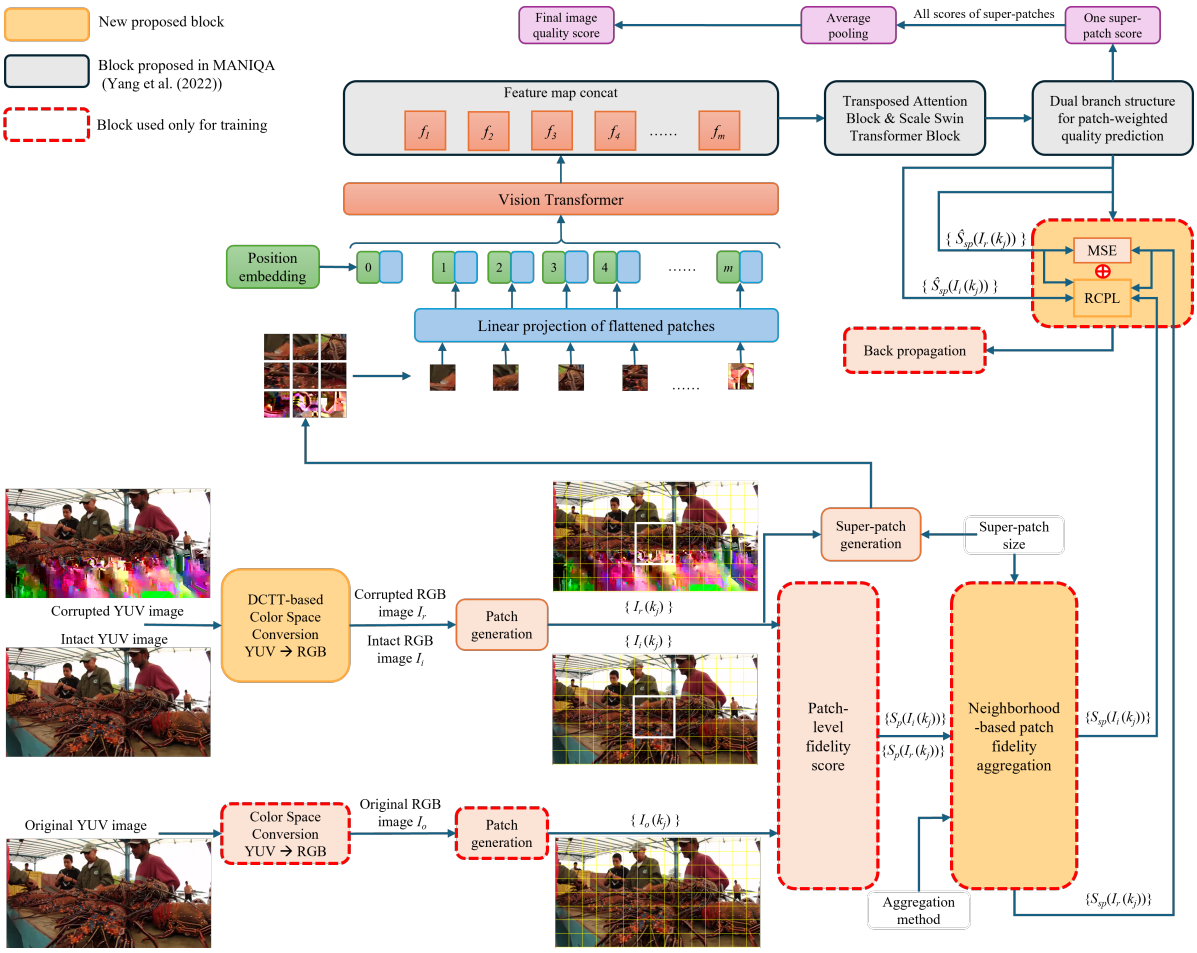


Figure 0.3 La mesure proposée d'estimation de la qualité d'image assistée par transformeur

L'architecture de notre métrique se compose de plusieurs blocs, dont le prétraitement des images, l'entraînement du réseau et le calcul du score final de l'image. Pour chaque séquence vidéo YUV originale, une liste de vidéos candidates est générée et les premières images corrompues de chaque séquence candidate sont extraites pour entrer dans notre réseau. L'image candidate est représentée par I_r, et chaque liste de vidéos candidates comprend une version intacte I_i, qui est reçue sans erreur. Nous préparons également la version de l'image originale correspondante I_o. Avant d'alimenter les images vers le réseau d'entraînement, le prétraitement des images est appliqué. Il se compose de plusieurs étapes :

- Tout d'abord, nous appliquons la conversion de l'espace colorimétrique de l'image pour changer le format de l'image de YUV420 à RGB, avec un composant Discriminant Color Texture Transformation (DCTT) (voir Section 3.3.3) pour bien distinguer entre les régions uniformes causées par des erreurs et des zones d'images réellement uniformes.
- Deuxièmement, nous générons les patchs $I(k_j)$ pour chaque version de l'image et calculons les scores de fidélité avec référence au niveau du patch $S_p(I(k_j))$. Ensuite, nous combinons

$p \times p$ patchs normaux pour générer des *super-patchs* et utilisons le Neighborhood-based Patch Fidelity Aggregation (NPFA) proposé (voir Section 3.3.2) pour générer les scores de référence des super-patchs afin de mieux prendre en compte les discontinuités locales aux limites horizontales et verticales des HEVC CTU (blocs de codec) entre les patchs de voisinage. La taille des super-patchs et la méthode d'agrégation sont des paramètres variables qui pourraient être modifiés à l'avenir.

Le composant NPFA est proposé pour détecter les discontinuités entre les blocs CTU voisins. De plus, le remplacement de la métrique CNN par une métrique basée sur transformeur pose un défi en raison de la petite taille des patchs individuels utilisés dans le modèle CNN, ce qui n'est pas compatible avec l'architecture de base du transformeur. Ici, nous introduisons des super-patchs au lieu de simples patchs individuels. Les super-patchs sont la combinaison de patchs $p \times p$ de telle sorte qu'une image complète soit divisée en super-patchs superposés. Les super-patchs contiennent plusieurs blocs CTU voisins, ce qui permet au modèle d'analyser plus efficacement les distorsions localisées résultant de la propagation d'erreurs dans le voisinage d'un patch donné.

Notre composant NPFA proposé présente plusieurs avantages. Il accorde une plus grande importance aux erreurs locales et améliore encore les performances par rapport aux patchs individuels classiques. Il garantit également que le modèle apprend complètement les distorsions locales dans différentes régions de l'image. Et en utilisant le super-patch comme entrée du modèle transformeur, nous augmentons également la quantité de données d'apprentissage par rapport à l'utilisation des images entières.

Nous avons aussi constaté que le simple fait de modifier la normalisation locale ne permet pas de faire la différence entre un patch uniforme bien reçu et un patch erroné qui semble uniforme. Par conséquent, nous proposons une conversion de l'espace colorimétrique *Transformation discriminante de texture de couleur* (DCTT) pour résoudre le problème. Elle crée un motif totalement différent pour chaque canal d'une image RGB, ce qui ne se produira pas dans une image RGB naturelle où les canaux RGB sont fortement corrélés. Comme le décodeur initialise chaque pixel YUV d'une image à (0,0,0) avant le décodage, la valeur d'un pixel perdu reste nulle lorsqu'une erreur se produit. Par conséquent, les régions perdues sont faciles à identifier. Pour chaque pixel (i, j) où les valeurs YUV sont toutes détectées comme 0 dans le patch k de l'image reconstruite I_r, nous appliquons les équations suivantes :

$$
\begin{aligned}
I_{r,R}(k, i, j) &= 255((-1)^{i+j} + 1)/2 \\
I_{r,G}(k, i, j) &= 255((-1)^{i} + 1)/2 \\
I_{r,B}(k, i, j) &= 255((-1)^{j} + 1)/2
\end{aligned}
$$

Ces motifs n'existent pas dans les images RGB naturelles. Nous nous attendons à ce que les patchs présentant ces motifs, étant très différents des autres patchs, se voient attribuer un score de qualité extrêmement faible au cours du processus d'apprentissage.

Le modèle MANIQA (Yang *et al.*, 2022) original n'utilisait que la fonction de perte d'erreur quadratique moyenne. Ces fonctions de perte simples fonctionnaient bien sur les images codées en intra, mais il reste une marge d'amélioration lorsqu'elles sont appliquées aux images codées en inter. Par conséquent, nous envisageons d'améliorer la fonction de perte de notre système pour garantir qu'un super-patch intact reçoive un score de qualité supérieur à une version corrompue, ce qui est particulièrement important pour les images codées en inter où les distorsions résultant d'une erreur de transmission ne sont pas aussi importantes que pour les images codées en intra. Nous proposons la fonction de perte de pénalité RCPL (Ranking-constrained penalty loss function) pour y parvenir.

Nous avons évalué les performances par patch individual et par super-patch avec la méthode NPFA basée sur différentes fonctions d'agrégation, respectivement. Cette étude a montré que l'agrégation "minimum" conduit à moins de situations négatives, où la situation négative représente que le score prédit de patch corrompu est plus élevé que le score prédit de patch intact. Nous remarquons que la sélection appropriée de la fonction d'agrégation est importante pour réduire les situations négatives, mais elle n'est toujours pas suffisante pour éviter totalement ce cas. Nous devrions également pousser le système à éviter de telles situations négatives en premier lieu en ajoutant un terme de pénalité à la fonction de perte. En d'autres termes, nous voulons ajouter un terme F_2 à la fonction de perte en guise de pénalité quand un patch endommagé obtient un score plus élevé qu'un patch intact, ce qui est particulièrement important pour les images inter comportant des erreurs subtiles.

F_1 est la fonction de perte originale (erreur quadratique moyenne) utilisée dans le système MANIQA. F_2 est le nouveau terme de perte que nous ajoutons à la fonction de perte F pendant l'entraînement. En ajoutant F_2, nous imposons une pénalité lorsque le score prédit du super-patch corrompu est supérieur à celui du super-patch intact correspondant. Nous ajoutons une condition de F_2 : lorsque nous constatons que le super-patch candidat et la version intacte ont les mêmes scores de référence, nous ne considérons que F_1. Sinon, nous considérons les deux fonctions de perte ensemble, où nous utilisons α dans la plage [0,1] pour représenter le coefficient de la fonction de perte.

Pour le calcul du score final de l'image, nous collectons tous les scores de super-patch prédits pour chaque image et utilisons un pooling moyen pour obtenir le score de qualité au niveau de l'image.

Résultats expérimentaux

Dans cette partie, nous présentons nos résultats expérimentaux. Nous commençons par décrire les bases de données utilisées pour la formation et les tests. Ensuite, nous décrivons la méthodologie de formation et les critères d'évaluation. Enfin, nous fournissons une évaluation complète des performances, comprenant les résultats, une étude d'ablation, une analyse de sensibilité des paramètres et une discussion.

La plupart des bases de données existants pour l'évaluation de la qualité d'image se concentrent sur les distorsions synthétisées artificiellement (simulées) dans le contenu généré par l'utilisateur (Sheikh, 2005; Ponomarenko *et al.*, 2013; Lin *et al.*, 2019). Cependant, il manque des ensembles de données qui englobent divers types de distorsions non uniformes résultant d'erreurs de transmission, où le contenu est décodé sans dissimulation d'erreur. Par exemple, la base de données LIVE contient des distorsions résultant d'erreurs de transmission, mais les régions erronées sont rejetées et masquées plutôt que décodées et rendues comme dans notre cas. Par conséquent, nous avons développé des scripts et des instructions pour générer la base de données souhaitée. Le processus de génération de la base de données comprend plusieurs étapes: codage vidéo selon la norme HEVC, génération d'erreurs en flippant des bits dans des positions spécifiques dans les flux binaires vidéo et décodage de liste vidéo, sans masquage d'erreur, pour obtenir les N candidats représentant diverses tentatives infructueuses de correction de la vidéo. Après avoir décodé tous les candidats vidéo, nous extrayons les images corrompues de chaque séquence vidéo, si elles sont décodables, et les ajoutons à notre base de données.

Nous utilisons les séquences originales des bases de données publiques (xip; Wang *et al.*, 2016; Pinson, 2013). Les vidéos collectées sont au format YUV420 avec une résolution de 1920×1024 pixels. Nous extrayons les 10 premières images de chaque vidéo pour les encoder avec la norme HEVC (Sze & al, 2014) avec le profil P à faible délai. Parmi les différentes valeurs possibles de pas de quantification (Quantization Parameter (QP)), nous avons choisi 37 et 22, qui correspondent respectivement aux points de fonctionnement à bas et haut débit des conditions de test associées au logiciel de référence standard HEVC (HM) que l'on trouve dans les Common Test Conditions (Bossen, 2013). Nous supposons que chaque image encodée est contenue dans un seul paquet vidéo. La première image de la vidéo encodée est une image codée en intra (I), et les 9 images suivantes sont des images codée en inter (P).

Nous voulons simuler la combinaison d'une erreur de transmission suivie d'un décodage de liste où les bits sont inversés à différents endroits, pour ainsi répartir l'erreur sur toute la trame vidéo du début à la fin. Pour simplifier le processus, nous plaçons les erreurs binaires à divers endroits du paquet. Cette méthode est compatible avec les scénarios de décodage de liste, où les bits modifiés pour générer des candidats apparaissent dans des emplacements aléatoires et imprévisibles. Par exemple, dans la correction d'erreur basée sur le CRC, comme mentionné dans (Boussard *et al.*, 2020a), les candidats présentent des patterns avec des bits modifiés dans de tels emplacements imprévisibles.

Par conséquent, nous avons sélectionné des positions de bits inversées en fonction de l'équation $pos = \beta \times M$, où pos indique la position du bit inversée, $\beta = \{0.1, 0.2, 0.3, ..., 0.9, 0.99\}$ et M est la taille de chaque paquet. Cette approche assure une diversité significative dans les modèles d'erreurs de transmission pour entraîner le système. Nous incorporons les modèles d'erreurs de transmission séparément pour les trames codées en intra et les trames codées en inter, en fonction du scénario étudié. Par conséquent, les erreurs dans les trames codées en inter sont directement appliquées à la trame elle-même, plutôt que d'être propagées à partir des erreurs des trames précédentes. Les trames candidates sujettes à ces diverses erreurs sont décodées et ajoutées, si elles sont décodables, à notre base de données. Pour chaque séquence et chaque type

de trame, nous générons 11 candidats, dont un candidat sans erreur (intact). Il en résulte 990 images corrompues à partir de 90 images de référence, formant notre base de données basée sur les séquences.

Les configurations de nos expériences sont les suivants : Pour le système assisté par transformeur, notre base de données contient environ 830 000 super-patchs superposés pour chaque type d'image dans notre simulation, avec une taille de super-patch de 224×224 pixels. Pour le système assisté par CNN, nous avons environ 475 200 patchs pour chaque type d'image avec une taille de patch de 64×64 pixels. Afin de mieux évaluer les performances de nos modèles, nous définissons plusieurs métriques.

Nous améliorons le schéma basé CNN proposé en ajoutant les composants suivants proposés:
- Transformation discriminante de texture de couleur (DCTT) pour distinguer entre un patch uniforme bien reçu et un patch d'erreur initialisé pour être uniforme par le décodeur.
- Fonction de perte de pénalité contrainte par classement (RCPL) pour pénaliser davantage les cas où un patch endommagé obtient un score plus élevé qu'un patch intact.

En ajoutant deux nouveaux composants, notre schéma basé par CNN fonctionne également mieux dans les images codées en intra et codées en inter.

Nous présentont les résultats expérimentaux comparant nos approches aux méthodes à l'état de l'art, utilisant des images codées respectivement en mode intra et en mode inter. Comparé avec les autres modèles pré-entraînés, l'application des composants DCTT et Ranking-Constrained Penalty Loss function (RCPL) proposés aux systèmes assistés par transformeur et assistés par CNN pour les images codées en intra et codées en inter se traduit par une précision améliorée et une différence de qualité réduite entre les qualités moyennes d'images intactes et d'images choisies par nos modèles. Les avantages de l'utilisation de ces composants nouvellement proposés dans notre système assisté par transformeur et du réentraînement sur nos bases de données proposées sont évidents.

Par rapport au modèle basé CNN, le modèle basé transformeur montre une plus grande sensibilité dans la détection de petites distorsions dans les images codées en inter lors de l'utilisation des blocs nouvellement proposés. Ces composants permettent à notre modèle de mieux apprendre les dégradations de qualité causées par les distorsions locales dues aux erreurs de transmission dans les images corrompues.

Nous fournissons des expériences d'ablation pour expliquer l'effet de différents paramètres et de chaque composant de notre méthode proposée en comparant les résultats sur notre ensemble de données proposé. Nous effectuons également des expériences de sensibilité des paramètres pour évaluer la sensibilité de la sortie de notre modèle proposé aux changements des variables de la base de données.

Conclusion

En conclusion, au cours de notre recherche, nous avons présenté un cadre de décodage de liste de vidéos assisté par apprentissage profond pour identifier le meilleur candidat d'une liste dans le contexte de communications vidéo peu fiables. Ce cadre sélectionne la vidéo candidate avec la qualité visuelle évaluée la plus élevée dans la liste des candidats. Nous avons proposé deux mesures IQA sans référence différentes pour évaluer la qualité du contenu vidéo endommagé par des erreurs de transmission:

- La première est une méthode basée sur CNN. Nous améliorons la normalisation locale puis appliquons les nouveaux composants DCTT et RCPL proposés dans l'architecture.
- La deuxième approche est basée sur le modèle transformeur pour améliorer les performances de notre système IQA. Nous réentraînons le modèle de base transformeur MANIQA avec des «super-patchs» supervisés par des scores générés par la méthode NPFA, et aboutissons à de meilleurs résultats en présence de discontinuités locales aux limites horizontales et verticales des blocs codés entre les patchs voisins. Nous appliquons également les fonctions améliorées de conversion de l'espace colorimétrique DCTT et de perte RCPL dans le modèle transformeur, ce qui améliore les performances du modèle.

Nous avons construit une nouvelle base de données contenant des vidéos corrompues en utilisant des vidéos originales provenant de bases de données publiques. Des modèles d'erreur simples ont été appliqués aux paquets vidéo codés pour simuler l'effet de la transmission sur des réseaux sujets aux erreurs. Grâce à des simulations et des expériences complètes, nous montrons que notre solution assistée par apprentissage profond est performante. Notre approche atteint une précision de décision remarquable, atteignant 100% pour les erreurs sur images intra et 96% pour les erreurs sur images inter, ce qui constitue une amélioration significative par rapport aux autres méthodes évaluées.

L'une des futures orientations de recherche possibles est d'intégrer des informations temporelles dans les métriques. Il serait également intéressant d'appliquer le modèle proposé à une gamme plus large de normes de compression vidéo et de scénarios d'erreur pour développer des systèmes de transmission vidéo plus robustes et plus intelligents.

A propos de la production scientifique associée à notre projet, un article a été publié dans la conférence CORESA en 2023 (Zhang *et al.*, 2023), ainsi qu'un poster lors de la journée du Mardi des chercheurs organisée à l'UPHF en 2023. De plus, nous avons rédigé un article de revue qui a été soumis à IEEE Access et est en cours d'évaluation (Zhang *et al.*, 2024).

Mots-clés: Transmission vidéo, décodage par liste, correction d'erreur vidéo basée sur contrôle de redondance cyclique, distorsions non uniformes, évaluation de la qualité de l'image, réseau neuronal convolutif, transformeur de vision

TABLE OF CONTENTS

Page

INTRODUCTION 1
0.1 Context 1
0.2 Objectives and contributions of our research 4
0.3 Thesis organization 6

CHAPTER 1 LITERATURE REVIEW 7
1.1 Introduction 7
1.2 Error management in video communications 7
1.2.1 Video error concealment 7
1.2.1.1 Spatial Error Concealment Methods 8
1.2.1.2 Temporal Concealment Methods 11
1.2.1.3 Spatio-Temporal Concealment Methods 13
1.2.2 Video error correction methods 16
1.2.2.1 CRC-based Error Correction 17
1.2.3 List decoding 20
1.3 Objective video quality assessment methods 21
1.3.1 Traditional VQA methods 23
1.3.1.1 Full-reference VQA metrics 23
1.3.1.2 No-reference VQA metrics 28
1.3.2 Deep-learning based VQA methods 32
1.3.2.1 CNN-based VQA methods 33
1.3.2.2 Transformer based VQA methods 37
1.3.3 Analysis of the existing VQA methods 40
1.3.3.1 Examples with the distortion caused by the transmission error 40
1.3.3.2 Analysis and the experimental results of the existing methods 41

CHAPTER 2 PROPOSED CNN-ASSISTED VIDEO LIST DECODING SYSTEM 47
2.1 Introduction 47
2.2 Proposed DL-assisted video list decoding system 48
2.3 Proposed CNN-based visual quality evaluation method 49
2.3.1 The inability to distinguish between various sources of uniform blocks 51
2.3.2 Proposed CNN-based method with improved local normalization 52
2.4 The video database 54
2.4.1 Creation of the database 54
2.4.2 Examples from our database 56
2.5 Experimental results 57

2.5.1 Training and testing methodologies 57
2.5.2 Performance evaluation criteria 58
2.5.3 Results and sensitivity analysis 58
2.5.4 Concluding remarks 62

CHAPTER 3 PROPOSED TRANSFORMER-ASSISTED VIDEO LIST DECODING SYSTEM 65
3.1 Introduction 65
3.2 Improved DL-assisted video list decoding system 66
3.3 Proposed Transformer-based visual quality evaluation method 67
3.3.1 The original Transformer architecture 68
3.3.2 Application of Neighborhood-based patch fidelity aggregation 71
3.3.2.1 Analysis of individual patches 72
3.3.2.2 Justification for the utilization of NPFA 74
3.3.3 Discriminant Color Texture Transformation 81
3.3.4 Ranking-constrained penalty loss function 82

CHAPTER 4 EXPERIMENTAL RESULTS 85
4.1 Introduction 85
4.2 Experimental results for the CNN-assisted IQA system component 85
4.2.1 Training and testing methodologies 85
4.2.2 Application of DCTT and RCPL to CNN-based method 86
4.2.3 Simulation results and analysis 87
4.2.4 Ablation study 89
4.2.4.1 Parameters sensitivity analysis 91
4.3 Experimental results for the Transformer-assisted IQA system component 92
4.3.1 Training and testing methodologies 92
4.3.2 Simulation results and analysis 93
4.3.3 Ablation studies 96
4.3.4 Parameter sensitivity analysis 98

CONCLUSION AND RECOMMENDATIONS 101
5.1 Conclusion 101
5.2 Further works and perspectives 102

BIBLIOGRAPHY 104

LIST OF TABLES

Page

Table 1.1 Performance on intra-coded images with the existing methods. 45

Table 1.2 Performance on inter-coded images with the existing methods. 45

Table 2.1 Performance on intra-coded images. 59

Table 2.2 Performance on inter-coded images. 59

Table 2.3 Performance on intra-coded images with CNN-assisted system. 60

Table 2.4 Performance on inter-coded images with CNN-assisted system. 61

Table 3.1 Analysis of patch results with CNN predicted scores. 74

Table 3.2 Analysis of negative difference results. 74

Table 3.3 Analysis of super-patch results with CNN predicted scores. 77

Table 3.4 Analysis of super-patch results with ground-truth PSNR scores. 80

Table 4.1 Performance on intra-coded images with CNN models applied newly proposed components DCTT and RCPL. 87

Table 4.2 Performance on inter-coded images with CNN models applied newly proposed components DCTT and RCPL. 88

Table 4.3 Performance on intra-coded images with CNN model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 89

Table 4.4 Performance on inter-coded images with CNN model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 89

Table 4.5 Performance on inter-coded images with different coefficients in loss function on CNN model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 91

Table 4.6 Performance on intra-coded images with CNN model by changing the patch size. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 91

Table 4.7 Performance on inter-coded images with CNN model by changing the patch size. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 92

Table 4.8 Performance on intra-coded images with Transformer model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 94

Table 4.9 Performance on inter-coded images with Transformer model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 94

Table 4.10 Performance on intra-coded images with Transformer model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 96

Table 4.11 Performance on inter-coded images with Transformer model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 97

Table 4.12 Performance on inter-coded images with different coefficients in loss function on Transformer model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB 97

Table 4.13 Performance on intra-coded images with our model by changing QP. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 98

Table 4.14 Performance on inter-coded images with our model by changing QP. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB. 98

LIST OF FIGURES

Page

Figure 0.1 Le processus traditionnel de décodage de liste : a) processus général de décodage de liste ; b) différents critères de sélection de vidéos candidates : à gauche, critères de l'état de l'art ; à droite, le critère proposé basé sur l'apprentissage profond. XIV

Figure 0.2 La mesure IQA basée sur CNN proposée précédemment pour classer les candidats vidéo dans le décodage de liste. L'image d'origine n'est utilisée que pendant la formation. XVII

Figure 0.3 La mesure proposée d'estimation de la qualité d'image assistée par transformeur XVIII

Figure 1.1 The region of adjacent macroblock (Ni & Li, 2017) 10

Figure 1.2 The classification of edge direction (Ni & Li, 2017) 10

Figure 1.3 Illustration of the boundary matching relationship (Liu *et al.*, 2012) 13

Figure 1.4 Illustration of the search area around the reference macroblock (MB), (Zabihi *et al.*, 2021) 14

Figure 1.5 Outer boundaries of the degraded MB and the candidate MB (Zabihi *et al.*, 2021) 16

Figure 1.6 Flowchart of the proposed method's algorithm to correct a single error in a packet (Boussard *et al.*, 2020a) 19

Figure 1.7 The traditional process of list decoding: a) general process of list decoding; b) different criterion of candidate video selection: left, criteria in state of the art; right, the proposed deep-learning based criterion. 22

Figure 1.8 Multi-scale structural similarity measurement system. L: low-pass filtering; 2↓: downsampling by 2. (Wang *et al.*, 2003) 26

Figure 1.9 How Video Multi-Method Assessment Fusion (VMAF) works: pixel level data are pooled to create frame-level features; different spatial and temporal features are fused using SVM regression to create frame-level quality score; consecutive frame scores are pooled to produce final video sequence VMAF score (Ioannis *et al.*, 2018) 28

Figure 1.10 A simple CNN architecture, comprised of just five layers (O'Shea & Nash, 2015) 33

Figure 1.11 A visual representation of a convolutional layer. The centre element of the kernel is placed over the input vector, of which is then calculated and replaced with a weighted sum of itself and any nearby pixels. (O'Shea & Nash, 2015) 35

Figure 1.12 The Transformer-model architecture (Vaswani *et al.*, 2017) 38

Figure 1.13 Scaled Dot-Product Attention (left). Multi-Head Attention consists of several attention layers running in parallel (right) (Vaswani *et al.*, 2017) 38

Figure 1.14 Examples of the sequences with the distortion caused by the transmission error 41

Figure 2.1 Proposed framework to optimize list decoding of corrupted videos during transmission (Zhang *et al.*, 2023). 49

Figure 2.2 The general inference process of the deep-learning based candidate selection within the proposed video list decoding framework. 50

Figure 2.3 The proposed CNN-based IQA metric for ranking video candidates in list decoding(Zhang *et al.*, 2023). 50

Figure 2.4 An example of two different uniformed patches with color space conversion from YUV420 to RGB 52

Figure 2.5 The problem of the original local normalization method in Kang *et al.* (2014) 53

Figure 2.6 The improved local normalization method in Zhang *et al.* (2023) 53

Figure 2.7 The database generation process. 54

Figure 2.8 Examples of the candidates in our database (Zhang *et al.*, 2023) 56

Figure 2.9 Example of bad decision (intra frame, CNN_YUV_NL proposed): choose the *best* decoded version. The system selects c) while the intact version is d) 63

Figure 2.10 Example of bad decision (inter frame, CNN_YUV_NL proposed): choose the *best* decoded version. The system selects c) while the intact version is d) 63

Figure 3.1 The proposed Transformer-assisted image quality estimation metric 66

Figure 3.2 The architecture of MANIQA (Yang *et al.*, 2022) 69

Figure 3.3 Transposed Attention Block of MANIQA (Yang *et al.*, 2022) 70

Figure 3.4 Scale Swin Transformer Block of MANIQA (Yang *et al.*, 2022) 71

Figure 3.5 Dual branch structure for patch-weighted quality prediction of MANIQA (Yang *et al.*, 2022) 71

Figure 3.6 Analysis of patches in training set with CNN predicted scores (top: all scores, bottom left: negative scores, bottom right: positive scores) 73

Figure 3.7 Example of super-patches (left: from intact frame, right: from corrupted frame) 75

Figure 3.8 Analysis of super-patches in training set with CNN predicted scores and average combinations (top: all scores, bottom left: negative scores, bottom right: positive scores) 77

Figure 3.9 Analysis of super-patches in training set with CNN predicted scores and min combinations (top: all scores, bottom left: negative scores, bottom right: positive scores) 78

Figure 3.10 Analysis of super-patches in training set with CNN predicted scores and squared error combinations (top: all scores, bottom left: negative scores, bottom right: positive scores) 79

Figure 3.11 Example of the application of the proposed Discriminant Color Texture Transformation (DCTT), as part of the YUV to RGB conversion, to an erroneous green patch (top) and to an intact uniform patch (bottom) 83

LIST OF ABBREVIATIONS

AGGD	Asymmetric Generalized Gaussian Distribution
AI	Artificial Intelligence
BLE	Bluetooth Low Energy
CNN	Convolutional Neural Network
CRC	Cyclic Redundancy Check
CTU	Coding Tree Unit
CV	Checksum Validation
DCTT	Discriminant Color Texture Transformation
DL	Deep Learning
DLM	Detail Loss Metric
DMOS	Difference Mean Opinion Score
FR	Full Reference
HD	High Definition
HEVC	High Efficiency Video Coding
HVS	Human Visual System
IoT	Internet of Things
IQA	Image Quality Assessment
ITS	Intelligent Transport Systems
LSTM	Long Short-Term Memory

MLP	Multilayer Perceptron
MOS	Mean Opinion Score
MSCN	Mean Subtracted Contrast Normalized
MSE	Mean Squared Error
MS-SSIM	Multi Scale Structural Similarity Index Method
MVG	Multivariate Gaussian
NALU	Network Abstraction Layer unit
NIQE	Natural Image Quality Evaluator
NLP	Natural Language Processing
NPFA	Neighborhood-based Patch Fidelity Aggregation
NR	No-Reference
NSS	Natural Scene Statistics
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QP	Quantization Parameter
RCPL	Ranking-Constrained Penalty Loss Function
RNN	Recurrent Neural Networks
RR	Reduced Reference
SEC	Spatial Error Concealment
SGD	Stochastic Gradient Descent

SROCC	Spearman Rank Order Correlation Coefficient
SSIM	Structural Similarity Index Method
SSTB	Scale Swin Transformer Block
STEC	Spatio-Temporal Error Concealment
STL	Swin Transformer Layers
SVM	Support Vector Machine
SVR	Support Vector Regressor
TAB	Transposed Attention Block
TCP	Transmission Control Protocol
TEC	Temporal Error Concealment
UDP	User Datagram Protocol
VIF	Visual Information Fidelity
VMAF	Video Multi-Method Assessment Fusion
VQA	Video Quality Assessment

INTRODUCTION

0.1 Context

Over the past few years, there has been an extremely rapid development of video devices, systems and applications. This growth in video will only intensify in the years to come, as video content already accounts for almost 80% of current internet traffic (Laghari *et al.*, 2023; Systems, 2016). Real-time video is increasingly popular and video content transmission constitutes the main category of data transmitted in the world nowadays. In the context of the emergence of the IoT, the targeted applications are ever more numerous for which visual information can considerably enrich the knowledge of the environment: video systems for remote surveillance and machine control, e-health devices, virtual and augmented reality. ITS are also directly concerned; the video makes it possible to communicate information on the driving environment or the state of the transport network between vehicles and the infrastructure.

Moreover, video quality experience has greatly improved over the past few years, thanks to the advent of HD video and the emergence of 4K content. As a consequence, video streams tend to represent a larger amount of data. To significantly reduce the size of such video streams, new video compression solutions were developed (Sullivan & Wiegand, 2005; Wiegand *et al.*, 2003; Sullivan *et al.*, 2012).

However, transmission errors occur on unreliable and error-prone networks, such as sensor networks and video internet objects (WiFi (IEEE, 2016), BLE (Collotta *et al.*, 2018), etc.). Such errors can significantly degrade the user experience by causing unwanted distortions like blur, geometric patterns, or green screen effects (see Figure 1.7a, center part). When erroneous packets are retransmitted, it also reduces network efficiency. In addition to being time-consuming and resource-intensive, such retransmissions may be incompatible with certain application domains, such as transportation and immersive video applications where video information must arrive

reliably and in real-time with very low latency. In such application domains, it is preferable to keep erroneous packets, even if they may lead to some visual artifacts, rather than discard them and request new ones.

Various approaches have been proposed in the literature to find effective, low-complexity solutions for repairing video packets containing bit errors. Error concealment and error correction methods are the two main classes of approaches used to handle damaged packets. Error concealment (Shirani *et al.*, 1999; Wang & Zhu, 1998) is a technology applied on the decoder side to regenerate the lost information in the decoded video stream (i.e. attempt to reconstruct the parts that were damaged in the transport and discarded). Error concealment leverages the correlation of adjacent regions in the current frame (spatial concealment (Koloda *et al.*, 2013b)) or previously received frames (temporal concealment (Peng *et al.*, 2002)) or both (spatio-temporal concealment (Kung *et al.*, 2006)) to recover lost areas.

On the other hand, error correction involves identifying and correcting the erroneous bits in a packet using various strategies, such as error-correcting codes (e.g., Reed-Solomon in Digital Video Broadcasting (Jokela & Lehtonen, 2007)), leveraging the reliability information of each received bit (e.g. LLR (Balatsoukas-Stimming *et al.*, 2015)) or utilizing newly proposed CRC-based error correction methods (Boussard *et al.*, 2020a). Unfortunately, the reliability information of each bit is usually not available at the video decoder. Furthermore, error-correcting codes add undesirable overhead to the communications. Therefore, since CRC is already widely used in IP communications (e.g. in UDP and TCP packets) and accessible at the application layer, CRC-based error correction is a promising approach. It utilizes readily available information and does not add overhead. However, in practice, CRC-based error correction can only correct a limited number of errors. Indeed, depending on the generator polynomial, packet size, and the maximum number of errors considered, the method may not lead to a unique corrected packet but, rather, a list of potentially corrected packets. This is where list decoding becomes relevant

in combination with CRC-based error correction as well as with those leveraging received bit reliability.

In list decoding, a limited number of potential candidates are sequentially presented to the decoder until a valid video is generated (i.e. without generating syntax errors or crashing the decoder). Each candidate is a variant of the received packet where one or several bits have been altered as an attempt to correct it. For instance, a candidate may be obtained by altering some of the least reliable bits identified from the LLRs. The candidates are typically ranked and presented to the decoder from most likely to least likely. In the case of CRC-based error correction, it is not possible to rank the candidates if LLRs are not available. Furthermore, even if LLRs were available, many candidates may be decoded without error and the first one decoded (most likely) is not necessarily the corrected one. Consequently, it is highly desirable to develop a system that can assess the visual quality of the candidates to select the best one.

Therefore, the ideal way to select the version is not taking the first decodable candidate or checking LLRs but assessing the visual quality of each candidate. Since the original video sequence is not available, we need to explore various NR visual quality assessment approaches. Considering that the error will be managed on each frame, we will consider IQA rather than VQA. We propose to use a deep-learning based NR IQA system to select the best candidate video with the highest quality score.

This system assesses the quality of videos subjected to transmission errors without discarding lost packets or concealing lost regions. The distortions caused by transmission errors significantly differ from those considered by traditional visual quality metrics, which typically address global, uniform distortions in the image and, as we will show, these metrics fail to distinguish the corrected version from various corrupted video versions. This comprehensive approach, combining traditional, but revisited, list decoding techniques with a deep-learning architecture

to assess visual quality and identify the best candidate, is unprecedented and offers excellent performance.

0.2 Objectives and contributions of our research

The main objective of this research project is to develop, in the context of video list decoding for error-prone video transmissions, a new method for automatically selecting the highest quality video from a candidate list generated by a correction method such as the CRC-based error correction. Consequently, we aim to select the best candidate based on the visual quality determined by a deep-learning (DL) system. Assuming that the error will be assessed on each frame individually, we consider image quality assessment rather than video quality assessment. More specifically, each candidate will undergo processing by a DL-based NR IQA method to obtain a score. Subsequently, the system selects the candidate with the highest IQA score. Therefore, the main contributions of our research are as follows:

- We first propose a CNN-assisted video list decoding framework to identify the best quality candidate in the decodable candidate list. We develop a CNN-based IQA metric to evaluate video quality image by image without considering the temporal relation between the frames. We use the structure proposed in (Kang *et al.*, 2014) as the backbone of our method. The original network is designed to evaluate image quality with local distortions, and we use patch-level fidelity scores and propose an improved local normalization to evaluate image quality with non-uniform distortions caused by transmission errors.
- We then propose the first Transformer-assisted video list decoding framework for error-prone video transmission systems. This framework identifies the candidate with the highest visual quality from multiple options generated during the list decoding process in unreliable networks, ensuring the final decoding is optimally efficient in terms of visual quality.
- We enhance the proposed DL-assisted framework by adding the following components:

a) A Discriminant Color Texture Transformation (DCTT) to distinguish between a well-received uniform patch and an error patch initialized as a uniform green patch by the decoder.

b) A Ranking-Constrained Penalty Loss function (RCPL) function is proposed to further penalize cases where a damaged patch obtains a higher score than an intact one, which is particularly important for inter images with subtle errors.

The proposed advanced framework is designed to evaluate image quality with local distortions. In other words, it is sensitive to detect the non-uniform spatial distortions caused by transmission errors. By adding two new components, our framework performs significantly better both for intra-coded and inter-coded images.

- We improve the system by replacing the CNN component in the proposed DL-assisted video list decoding framework by a Transformer-based component. This IQA method based on Transformer uses the structure proposed in (Yang *et al.*, 2022) as the backbone, and we propose a Neighborhood-based Patch Fidelity Aggregation (NPFA) to better consider the local discontinuities at horizontal and vertical boundaries of coded blocks between neighbouring patches.
- We also build a database using HEVC compression on original YUV video sequences collected from public datasets (xip; Wang *et al.*, 2016; Pinson, 2013). Most existing datasets for image quality assessment focus on artificially synthesized losses or user-generated losses, but do not include different types of non-uniform distortions caused by transmission errors. Therefore, we create the scripts and instructions to regenerate a database, including the standard HEVC (Sze & al, 2014) encoding and addition of transmission errors to obtain non-uniform corrupted frames. Simple error patterns are applied to the encoded video packets to simulate the effect of transmission over error-prone networks. And we collect the combination of $p \times p$ patches, which we call "super-patch", in the corrupted frames from

these decoded video bitstreams, to perform the training and testing with the ground-truth neighborhood-based patch fidelity aggregation scores.

This research led to a published conference paper (Zhang *et al.*, 2023) and a journal submission (Zhang *et al.*, 2024).

0.3 Thesis organization

This manuscript is organized as follows. In Chapter 1, we review methods in the literature for error handling in video communications, from the concealment of errors to the actual correction of a corrupted packet. We also review the VQA methods and take advantage of their strengths to identify a strategy to improve performance. The remaining chapters introduce the proposed deep-learning assisted video list decoding frameworks, including the CNN-based method in Chapter 2, and the Transformer-based method in Chapter 3. The experimental results and the analysis of the performances are presented in Chapter 4. We then conclude the thesis.

CHAPTER 1

LITERATURE REVIEW

1.1 Introduction

In this chapter, we introduce the background of the IQA methods presented in the next chapters. We first introduce different error management methods in video communications, including video error concealment and video error correction. We also provide a detailed overview of traditional list decoding systems and highlight their disadvantages, thereby underscoring the necessity of the deep-learning-assisted approach we propose. In the second part, we discuss the objective video quality assessment methods, including both traditional and deep-learning based approaches. We also introduce the basic knowledge related to deep learning models and discuss the application of deep learning architectures in video quality assessment. By discussing these methods and highlighting their disadvantages, we underscore the necessity of the Transformer-assisted metric which is well-suited for our video list decoding system.

1.2 Error management in video communications

Various approaches have been proposed in the literature to find solutions to effectively manage errors that can occur when transmitting video content over networks. Error concealment and error correction methods are the two main classes of approaches used to handle damaged packets. In this section, we introduce the different error concealment approaches and video error correction approaches. We also present the general view of the traditional list decoding approach.

1.2.1 Video error concealment

Error concealment (Shirani *et al.*, 1999; Wang & Zhu, 1998) is a technique utilized on the decoder side to regenerate lost information in the decoded video stream, aiming to reconstruct the parts damaged during transmission and subsequently discarded. This method leverages

the correlation of adjacent regions within the current frame (spatial concealment (Koloda *et al.*, 2013b)), previously received frames (temporal concealment (Peng *et al.*, 2002)), or a combination of both (spatio-temporal concealment (Kung *et al.*, 2006)) to recover lost areas. In the following section, we will discuss these principal methods of error concealment in detail.

1.2.1.1 Spatial Error Concealment Methods

The spatial error (SE) concealment method uses the information in the current frame to restore the missing data. This type of error concealment method normally recovers the corrupted region based on the surrounding blocks (Rongfu *et al.*, 2004) or the available groups of pixels (Koloda *et al.*, 2013a).

W. Kwok and H. Sun first proposed a multi-directional interpolation method for spatial error concealment in 1993 (Kwok & Sun, 1993). This method is a typical block-based method where the interpolation algorithm utilizes spatially correlated edge information from a large local neighbourhood of surrounding pixels and performs multi-directional interpolation to restore the missing block. The process of this method is to first classify the surrounding neighbourhood blocks and determine which directions characterize the strongest edges. According to each classified direction, spatial directional interpolation is used to create a set of blocks, each block has a strong edge in its respective direction. Then the blocks in the set are mixed so that all the strong features of each block are extracted and combined into one block. Therefore, it can recover detailed blocks containing multiple interpolated edges.

With the continuous development of technology, the current spatial error concealment methods have better adaptability to serve different coding standards and transmission standards. We select one of them here to introduce the principle in detail.

A spatial error concealment algorithm based on adaptive edge threshold and directional weight is defined by Ni and Li in 2017 (Ni & Li, 2017). This method proposes a three steps interpolation process:

- Use a novel technique to estimate the significant edges of missing areas after performing a directional edge analysis on the correctly received neighbouring blocks of the missing areas.
- Obtain an approximation along the direction of each significant edge.
- For each pixel, compute a weighted average by using two edge correspondence measures as weighting factors. The strength of each significant edge, i.e. its magnitude, is used as the first measure for weighted averaging. The similarity of two boundary pixels located at the ends of each significant direction is used as the second weighting measure.

This article uses a Sobel operator for edge gradient detection for each pixel in the adjacent macroblock. The Sobel operator is introduced in the following. S_v is the vertical operator and S_h is the horizontal operator.

$$S_h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_v = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{1.1}$$

If $p(x, y)$ represents the pixel value in the correctly received macro-blocks near the damaged macroblocks, the gradients of horizontal $g_h(x, y)$ and vertical $g_v(x, y)$ are:

$$\begin{aligned} g_h(x, y) &= S_h \otimes p(x, y), \\ g_v(x, y) &= S_v \otimes p(x, y). \end{aligned} \tag{1.2}$$

$\otimes$ represents the two-dimensional convolution operator. Thus, the gradient's magnitude $G(x, y)$ and direction $\theta(x, y)$ of pixel $p(x, y)$ are:

$$\begin{aligned} G(x, y) &= \sqrt{g_h^2(x, y) + g_v^2(x, y)}, \\ \theta(x, y) &= -\tan^{-1}\left(\frac{g_v(x, y)}{g_h(x, y)}\right). \end{aligned} \tag{1.3}$$

This article takes into account the fact that the correlation between adjacent macroblocks in the up, down, left, and right directions of the damaged macroblock is stronger. At the same time,

considering the reduction in the amount of calculations, it uses pixels of eight rows or columns in the above four directions near the damaged macroblock (as shown in Figure 1.1). Meanwhile, the edge direction is divided into eight directions 1–8 as illustrated in Figure 1.2.

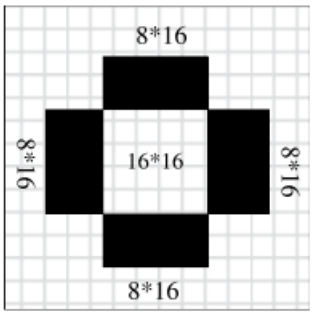


Figure 1.1 The region of adjacent macroblock (Ni & Li, 2017)

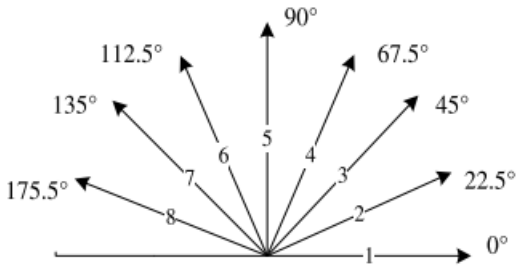


Figure 1.2 The classification of edge direction (Ni & Li, 2017)

Then the authors proposed an adaptive pixel interpolation algorithm to reconstruct damaged macroblocks. The proposed algorithm is composed of three steps. Firstly, the approach of adaptive edge detection threshold is introduced. The threshold is configured adaptively according to the specific content of the adjacent macroblocks. Secondly, the relevant pixels and the direction of weighting are determined according to the information of the adjacent macroblocks. Finally, the damaged macroblock is rebuilt by the adaptive interpolation.

1.2.1.2 Temporal Concealment Methods

The Temporal Error Concealment (TEC) methods use the temporal similarity between consecutive frames to hide erroneous MBs. The key operation in this method is to estimate the motion vector of the missing MB by using the information from the Motion Vector (MV) of the neighbouring MB to detect errors and hide them (Wu *et al.*, 2008).

The Boundary Matching Algorithm (BMA) (Lee *et al.*, 2004) is often used to select MVs among candidate MVs so that the internal and external boundaries of the missing MB maintain maximum smoothness. This algorithm was described in 1993 by Lam et al. (Lam *et al.*, 1993) and has been extended throughout the years (Chen *et al.*, 2003) to be adapted to recent video coding techniques.

The smoothness of the video signal makes the possibility of drastic changes between adjacent pixels in the picture less likely, so the main purpose of the boundary matching algorithm is to minimize the side matching distortion D_{sm} between the internal and external boundaries of the reconstructed MBs (Wang *et al.*, 2002). As shown in Figure 1.3, the internal boundary represents the boundary pixels of the MB, and the external boundary represents the surrounding pixels in the corresponding spatially adjacent MBs. D_{sm} is defined as the sum of the absolute difference between the internal boundaries of the candidate block in the reference frame and the external boundaries of the missing block in the current frame:

$$
\begin{aligned}
D_{\mathrm{sm}} &= \frac{1}{(w_N + w_S + w_W + w_E)M} \\
&\times \Bigg[w_N \sum_{i=0}^{M-1} |f(x+i, y-1, t) \\
&\quad - f(x + mv_x + i, y + mv_y, t-1)| \\
&\quad + w_S \sum_{i=0}^{M-1} |f(x+i, y+M, t) \\
&\quad - f(x + mv_x + i, y + mv_y + M - 1, t-1)| \\
&\quad + w_W \sum_{i=0}^{M-1} |f(x-1, y+i, t) \\
&\quad - f(x + mv_x, y + mv_y + i, t-1)| \\
&\quad + w_E \sum_{i=0}^{M-1} |f(x+M, y+i, t) \\
&\quad - f(x + mv_x + M - 1, y + mv_y + i, t-1)| \Bigg]
\end{aligned}
\tag{1.4}
$$

where M is the size of MB, the subscripts N, S, W, E are short for North, South, West, and East respectively, (x, y) is the location of the top-left pixel in the current lost block, (mv_x, mv_y) is the candidate MV which could be zero MV or the MVs of neighbouring adjacent blocks. And $f(\cdot, \cdot, t)$ stands for current frame, $f(\cdot, \cdot, t-1)$ is the corresponding previous reference frame.

Motion Vector Extrapolation (MVE) concealment techniques was developed by Peng et al. in 2002 (Peng *et al.*, 2002). This method tries to reduce the computational complexity of recovering lost frames. In this method, the lost frame is divided into contiguous blocks. For each of these contiguous blocks, a motion vector is calculated from the extrapolated motion vectors using a weighted average.

Liu et al. (Liu *et al.*, 2012) proposed an adapted version of the MVE algorithm to fit with HEVC, a compression standard which allows more flexible block management. In this method,

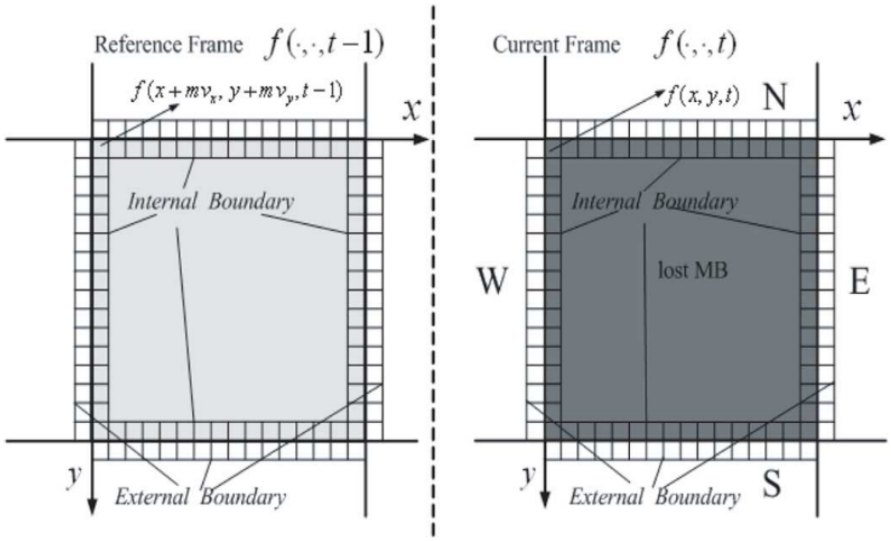


Figure 1.3 Illustration of the boundary matching relationship (Liu *et al.*, 2012)

firstly, by analyzing the texture and motion information of the missing blocks, strong correlations between the time domain are used to judge variable size blocks. Then, the regions with simple textures are concealed. Finally, an error concealment method based on variable size motion vector extrapolation is proposed, which can greatly reduce the complexity while guaranteeing better reconstruction quality.

1.2.1.3 Spatio-Temporal Concealment Methods

The spatio-temporal error concealment method combines Spatial Error Concealment (SEC) method and TEC method, in other words, both intra-frame and inter-frame information is used to recover the lost macroblocks in the video. Most parts of these methods consist of the adaptation of a temporal and a spatial error concealment method so that they can be used together and increase the accuracy of the concealment.

Traditional spatio-temporal error concealment methods often have high computational complexity. With the avancement of more and more emerging technologies, some more flexible methods (Seiler *et al.*, 2013; Shih *et al.*, 2018) are gradually being more widely used. In 2021, Zabihi et al. (Zabihi *et al.*, 2021) proposed a novel Spatio-Temporal Error Concealment (STEC) method

based on the non-local means (Buades *et al.*, 2011) concept. In this method, due to the small MB-to-picture-size ratio in high-resolution videos, the authors proposed to recover the lost pixels of the degraded MBs simultaneously in order to alleviate the high amount of calculation.

Since the consecutive frames in a video sequence are usually strongly correlated, appropriate MBs are more probable to be found in the previous frame than in the current frame. In the proposed method the search area is considered in the previous frame and so, the reconstructed MB is given as follows:

$$\boldsymbol{W}_p^t = \frac{\sum_{q \in H(p)} s_q^{t-1} \boldsymbol{W}_q^{t-1}}{\sum_{q \in H(p)} s_q^{t-1}} \tag{1.5}$$

where $\boldsymbol{W}_p^t$ and $\boldsymbol{W}_q^{t-1}$ represent the reconstructed MB in the current frame and the candidate MB in the previous frame, respectively. p and q represent the coordinates of the top-left corner of reconstructed and candidate MBs in the frame, respectively. s_q^{t-1} represents the weight assigned to the candidate MB, $\boldsymbol{W}_q^{t-1}$, in the search area, $\boldsymbol{H(p)}$, in the previous frame, $t-1$.

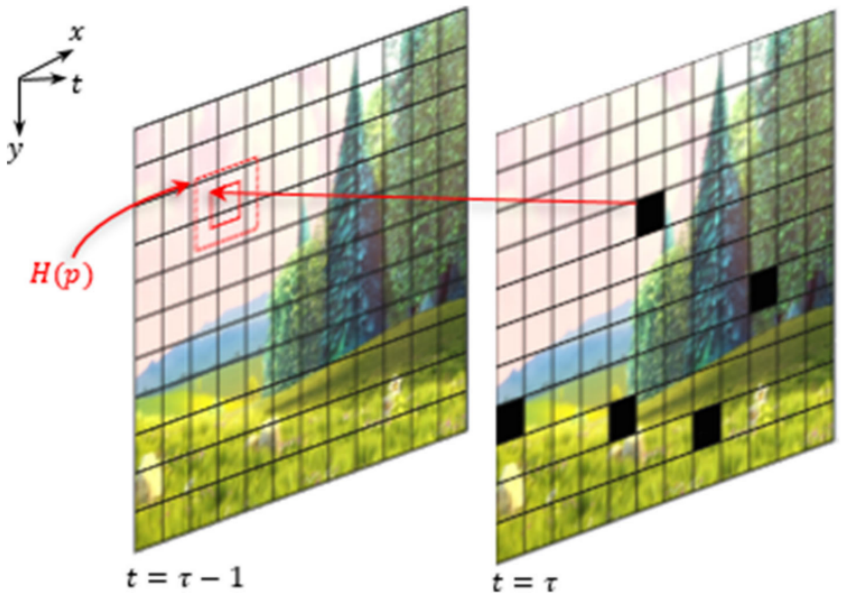


Figure 1.4 Illustration of the search area around the reference MB, (Zabihi *et al.*, 2021)

In this method, the search area is restricted to a reference MB and its available R_s-pixels width neighbourhood. The search area is illustrated in Figure 1.4. In this way, the degraded MB is recovered as the weighted average of only the MBs within the search area and not all the

available MBs in the reference frame. The average of the available neighbouring MVs is used to determine the position of the reference MB in the previous frame. Accordingly, the coordinates of the top-left pixels of the degraded MB and the candidate MB, p and q, are related to each other by:

$$q = p + (MV_x + x_d, MV_y + y_d), x_d, y_d \in \{-R_s, ..., R_s\} \tag{1.6}$$

where MV_x and MV_y represent the horizontal and vertical components of the estimated MV respectively. x_d and y_d indicate the horizontal and vertical displacements around the reference MB respectively. R_s determines the maximum horizontal and vertical displacements around the reference MB.

In order to calculate the weight for each candidate MB, the sum of absolute differences between the available outer boundaries of the degraded MB and the candidate MB is calculated as follows:

$$d(\boldsymbol{W}_p^t, \boldsymbol{W}_q^{t-1}) = \frac{1}{|\mathcal{N}(p)|} \sum_{(x,y)\in\mathcal{N}(p)} |u_1(x, y, t) - u_1(x + MV_x + x_d, y + MV_y + y_d, t - 1)| \tag{1.7}$$

In 1.7, $\mathcal{N}(p)$ contains the coordinates of the available pixels on the outer boundaries of the degraded MB. $|\mathcal{N}(p)|$ denotes the cardinality of the $\mathcal{N}(p)$. The outer boundaries of the lost MB and the candidate MB are also illustrated in Figure 1.5. Then, the weight assigned to the MB W_q^{t-1} can be calculated as:

$$s_q^{t-1} = \exp\left(\frac{-d(\boldsymbol{W}_p^t, \boldsymbol{W}_q^{t-1})}{h}\right) \tag{1.8}$$

where h is the filtering parameter which controls the strength of the weighting as a function of the similarity degree so that for small values of h, only the candidate MBs with high similarity degree get significant weights. After determining the weights for all of the candidate MBs in the search area, the degraded MB is reconstructed by 1.5.

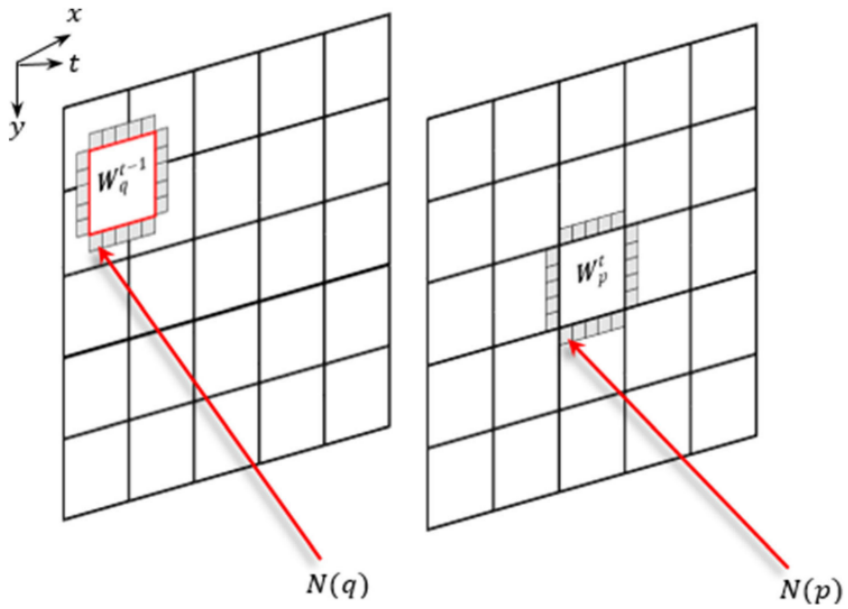


Figure 1.5 Outer boundaries of the degraded MB and the candidate MB (Zabihi *et al.*, 2021)

In this STEC method, the whole MBs corresponding to the MVs in the candidate MVs set take part in reconstructing the degraded MB through the Non-Local Means concept. This allows the reconstructed MB to be closer to the original MB, further improves the quality of the received video, and makes the viewer's viewing experience better. Besides, the homography-based STEC method proposed in (Chung & Yim, 2019) gives comparably good reconstruction results for large-size unknown regions with low complexity. By considering both global and local registration models, such an approach is more efficient to reconstruct complex and less natural movements in the scene.

1.2.2 Video error correction methods

Video error correction methods are commonly used at the decoder side in order to correct the bits in error within an erroneously received video packet, in order to improve the video quality and the user's viewing experience. Rather than concealing erroneous packets, error correction methods allow to correct the video packets. For example, bit-stream level error correction methods aim at

recovering the originally transmitted packet, which is generally performed at lower layers of the protocol stack. Protocol-aided error correction methods are based on the characteristics of the protocol used to transmit the video packets, and the supplementary information, to reconstruct the intact error-free packet. Those approaches are mainly used from the Link to Transport layers of the protocol stack. In a previous work (Boussard *et al.*, 2021a), Boussard et al. proposed a cross-layer video correction method based on CRC, which is a new and more efficient method to correct the distorted video packets after transmission over error-prone networks. In the following part, we will present this CRC-based video error correction method.

1.2.2.1 CRC-based Error Correction

The method proposed in Boussard *et al.* (2020a) uses the CRC syndrome present in low layers of protocol stacks in order to correct a given number of errors in data packets. The proposed algorithm generates the whole list of error patterns, leading to a given received syndrome containing up to a given maximum number of errors. This approach can instantly correct erroneous packets when the output list contains a single element. It can correct all single and double error patterns as well as most triple error cases when considering small payloads as used for example in IoT applications (Boussard *et al.*, 2020a).

CRC codes are typically used for error detection in lower layers of the protocol stack of a wired or wireless transmission. The CRC field is computed at the transmitter as the remainder of the long division of the protected data by a generator polynomial $g(x)$. The resulting remainder $r(x)$ is then appended to the packet and sent through the communication channel. At the receiver side, the long division by the generator polynomial is performed again on the received packet $p_R(x)$ and the appended remainder. At the end of that process, the newly computed remainder is known as the syndrome, denoted $s(x)$. Given the definition of the CRC computation, the syndrome can be expressed as follows (Boussard *et al.*, 2020b):

$$s(x) = p_R(x) \bmod g(x) \tag{1.9}$$

which is equal to zero when no error occurs during the packet's transmission. Otherwise, when errors occur, the syndrome $s(x)$ will differ from zero. If we consider an error pattern $e(x)$ with non-null coefficients at error positions, we have:

$$s(x) = (p_R(x) + e(x)) \bmod g(x) \tag{1.10}$$

$s(x)$ is a non-null syndrome. But a given syndrome value can be the result of several different error patterns $e(x)$, containing different numbers of errors. $E_M(s(x))$ is the set of all valid error patterns leading to the syndrome $s(x)$. The error patterns in E_M contain between 1 and M errors (all bits of the packet are erroneous in the latter case). All error patterns of E_M are defined as:

$$E_M = \{e(x) \in GF(2^M) | e(x) = s(x) + q(x)g(x) \quad \text{with} \quad q(x) \in GF(2^m)\} \tag{1.11}$$

where m is the payload length and $GF(2^m)$ is the Galois Field of order 2^m (i.e., the set of binary polynomials of length m). In other words, the error pattern corresponding to the syndrome can be any binary polynomial of the highest degree $m-1$ (that we denoted as $q(x)$) multiplied by the generator polynomial, with $s(x)$ added. The set E_M is called the equivalence class containing $s(x)$. Every possible value of $q(x)$ in this equation will produce a CRC-compliant error pattern $e(x)$. The degrees of the non-zero coefficients in the resulting $e(x)$ correspond to the erroneous positions in the corrupted packet.

The search process for single-error patterns is illustrated in Figure 1.6. Each step is identified in Figure 1.6 using binary notations. Based on this algorithm, we can find all single-error pattern candidates (usually only one) and correct them to recover a single error in the received video packet.

The correction method for the case with one error cannot be used entirely for the case with $N \geq 1$ errors. However, we use the same principle by applying the logical operator between the error vector and the generator polynomial, but by forcing the error positions in order to scan all possible cases. To fix a bit in a location, it must be done by adding, if necessary, the generator polynomial shifted to this position in order to maintain the equivalence, i.e. generate solutions

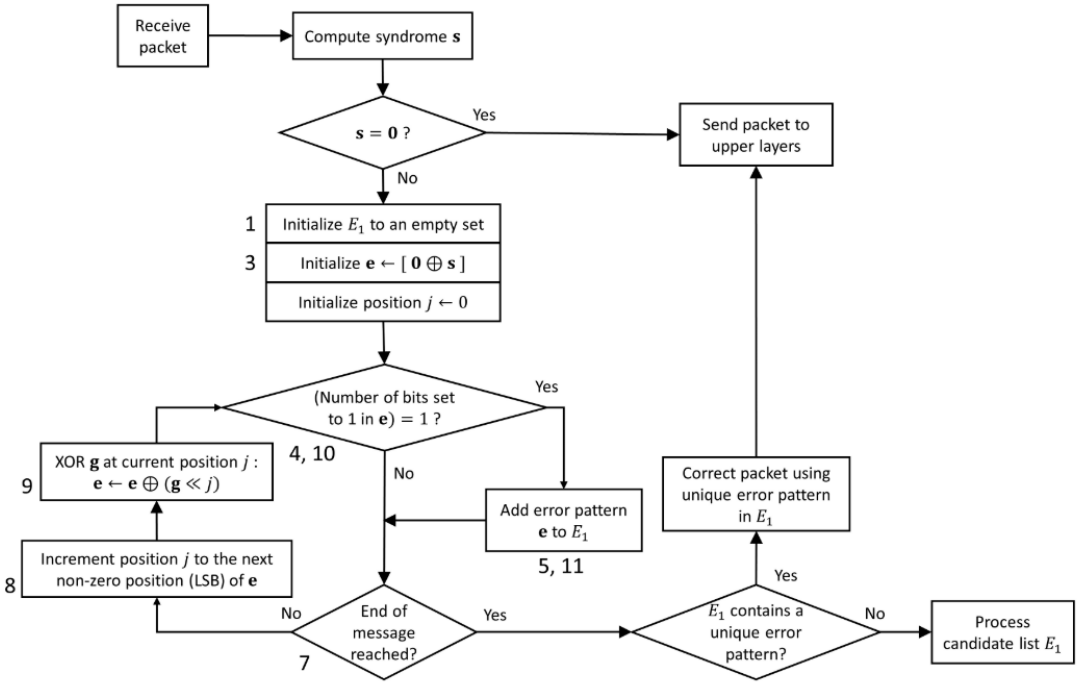


Figure 1.6 Flowchart of the proposed method's algorithm to correct a single error in a packet (Boussard *et al.*, 2020a)

leading to the same syndrome. During this first iteration, we fixed the first position of the error at position 0. We apply previously the generating polynomial to the entire message. Knowing that we are looking here for the case with 2 errors, we recover all the error vectors where there are 2 errors or less. We then apply the same process by forcing the position of the first error to a new position where we force position 1, and we recover the set of error vectors containing 2 errors or less (Boussard *et al.*, 2021b).

On this basis, we obtained the candidate list for the N-errors patterns. However, the number of candidates increases with both the number of errors considered and the size of the packet. When we consider N errors, it will provide a very large number of candidates. The size of the list of potential candidates can be reduced by applying UDP/Transmission Control Protocol (TCP) checksum validation to each candidate (Golaghazadeh *et al.*, 2018). Every candidate which passes this verification will be retained. Otherwise, it will be rejected. This makes it possible to significantly reduce the number of possible candidates. The list can contain a single candidate

or even several candidates. For each candidate packet, it should be decoded. If the decoding operation fails, no candidate is selected. So at the end of the process, we can get one or more candidates after decoding in order to compare the quality of the resulting reconstructed videos or images.

1.2.3 List decoding

List decoding methods exploit corrupted received packets to generate multiple candidate transmitted packets from the corrupted received packet. According to this consideration, several researchers of list decoding try to find solutions to the video error correction problem that can be realistically integrated into current and next-generation mobile communication systems.

Figure 1.7a) shows the traditional process of list decoding proposed in previous works. In general, during video transmission, a raw YUV video sequence is firstly encoded, complying with a video compression standard such as H.264 or HEVC, and then transmitted over a communication channel. However, variable channel conditions, especially on a noisy channel, may cause the erroneous video bitstream after transmitting over it. At this point, to repair the corrupted video bitstream for more complete video content and better user experience, the list decoding approach is one of the existing methods to find the closest candidate to the original video quality. Traditional list decoding methods first reconstruct the corrupted video packets and generate an ordered list of candidates, represented by methods such as LLR based formulation (Balatsoukas-Stimming *et al.*, 2015), which generates ranked candidate list based on bit reliability, also CRC-based multi-error correction (Boussard *et al.*, 2020a, 2021b), which generates the most probable CRC-compliant candidate video packets.

Further filtering of the large list of candidates continues to use extra information in the next steps, for example, Checksum-Filtering exploiting the receiver side user datagram protocol checksum (Golaghazadeh *et al.*, 2018), CRC validation used in LLR approach (Balatsoukas-Stimming *et al.*, 2015; Caron & Coulombe, 2015), etc. All of the candidate video packets should be then validated by multiple video decoding, and the final candidates can be decoded normally.

The final list after the list decoding process may contain several different valid reconstructed candidate sequences, and the last step is to select the candidate video of the best visual quality as the final reconstructed video after video transmission.

The existing methods of list decoding have disadvantages in candidate video selection. As shown in Figure 1.7.b), because of the ranked candidate generation step during the list decoding process in state-of-the-art methods, the final candidate selection will take the first valid candidate in the final list as the reconstructed video. However, this simple selection is not rigorous, and the candidate video sequence at the top of the list is not necessarily the candidate with the highest reconstructed visual quality.

Therefore, the ideal way to select the version is not taking the first decodable candidate or checking LLR but assessing the visual quality of each candidate. Since the original video sequence is not available, we need to explore various NR visual quality assessment approaches, which will be discussed in the next subsection. Considering that the error will be managed on each frame, we will consider image quality assessment rather than video quality assessment. Consequently, we aim to select the best candidate based on its visual quality. This visual quality is determined by a DL system as illustrated in the right part in Figure 1.7b). This constitutes the first originality of our robust decoding approach. More specifically, each candidate will undergo processing by a DL-based NR IQA method to obtain a score. Subsequently, the system will select the candidate with the highest IQA score.

1.3 Objective video quality assessment methods

Video quality assessment refers to the perception and evaluation of changes and distortions between two different versions of the same video content. Video quality assessment methods are divided into two categories: subjective assessment methods and objective assessment methods (Chikkerur *et al.*, 2011).

In the case of subjective video quality assessment, the subjective score is generally expressed by the Mean Opinion Score (MOS) or the Difference Mean Opinion Score (DMOS) (Seshadrinathan

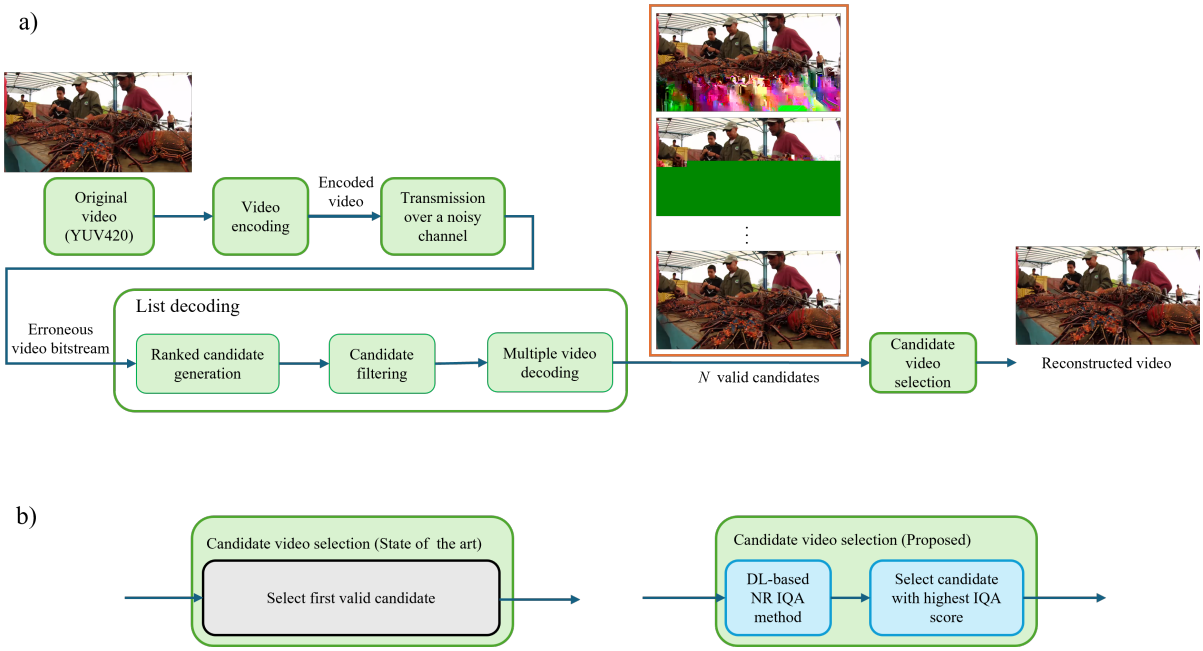


Figure 1.7 The traditional process of list decoding: a) general process of list decoding; b) different criterion of candidate video selection: left, criteria in state of the art; right, the proposed deep-learning based criterion.

et al., 2010). The former is to judge the video quality by normalizing the observer's score, and the latter is to judge the video quality by normalizing the difference between the observer's evaluation scores of the undistorted video and the distorted video.

In the case of objective video quality assessment, the computer calculates a quality index of the video according to a given algorithm. Although objective assessment metrics often try to mimic the human visual system in order to predict a score which reflects the viewer's experience, there is still a gap between existing objective indicators and subjective evaluations.

Objective IQA methods can be divided into three categories: Full Reference, Reduced Reference, and No Reference methods. The full reference method needs to provide a distortion-free original video, and after comparing the two video contents, an evaluation result of the distorted video is obtained. The reduced reference method refers to the reference is not the original video, but some features extracted from the original video or some added information are transmitted to the sink through a non-destructive auxiliary channel. Then, feature extraction is performed

on the video transmitted through the damaged main channel. Analyzing the degree of loss of this characteristic information reflects the degree of damage to the video quality. finally, the non-reference method is to evaluate the video quality without the original video at all. The general method is to decompose the quality factor into different types of distortion, effect or noise, and then establish a corresponding evaluation model (Then, 2019). In the following part, we will introduce several commonly used video assessment indicators, including the traditional methods and deep-learning based methods.

1.3.1 Traditional VQA methods

Traditional IQA metrics work well when evaluating certain artificially synthesized distortions and uniformly distributed distortions (Sara *et al.*, 2019; Wang *et al.*, 2004a; Li *et al.*, 2016; Mittal *et al.*, 2012, 2013). As a video sequence can be considered as a serie of video frames, each frame being a picture, image quality metrics are widely used to evaluate digital video quality on a frame-by-frame basis. In this section, we will introduce some traditional Full Reference (FR) and NR IQA metrics which will be used as reference indicators in the future experiment process.

1.3.1.1 Full-reference VQA metrics

The full-reference objective video quality evaluation method refers to comparing the original reference video and the distorted video between each corresponding pixel in each corresponding frame. To be precise, this method does not get the real video quality, but the similarity or fidelity of the distorted video relative to the original video.

A. Peak Signal-to-Noise Ratio

The Peak Signal-to-Noise Ratio (PSNR) is the most commonly used objective quality assessment technique to measure the quality of a given image compared to a reference one. The signal is considered as the original data and the noise is the error yielded by any image processing including compression or channel distortion. The PSNR is known to be an approximate estimation of the human perception of image quality (Sara *et al.*, 2019). It uses the mean square

error to calculate the image distortion. The larger the PSNR value, the closer the distorted image is to the reference image, that is, the better the image quality.

The Mean Squared Error (MSE) between an original image $g(x, y)$ of resolution $N \times M$ and its distroted version $\hat{g}(x, y)$ is defined as (Søgaard *et al.*, 2016):

$$MSE = \frac{1}{MN} \sum_{n=1}^{M} \sum_{m=1}^{N} [\hat{g}(n, m) - g(n, m)]^2 \tag{1.12}$$

And PSNR is expressed as:

$$\text{PSNR} = 10 \log_{10} \left(\frac{peakval^2}{MSE} \right) \tag{1.13}$$

where $peakval$ (Peak Value) is the maximal in the image data. If it is an 8-bit unsigned integer data type, the $peakval$ is 255 (Deshpande *et al.*, 2018). From Equation 1.13, we can see that it is a representation of absolute error in dB.

Although PSNR is a simple image quality metric which is not always well suited with human visual judgment, it is often used in video quality assessment, reflecting the quality of each frame or calculating the average PSNR of the entire video. Moreover, some recent research developed that PSNR shows a more reliable performance in assessing the quality of error-concealed videos than some other metrics, such as Structural Similarity Index Method (SSIM) (Kazemi *et al.*, 2020).

B. Structural Similarity Index Measurement (SSIM)

SSIM is a popular method for quality assessment of still images (Wang *et al.*, 2004a), that was extended to video in (Wang *et al.*, 2004b). The SSIM index method is calculated based on the computation of three major aspects termed as luminance, contrast and structural or correlation term. This index is a combination of multiplication of these three aspects (Brooks *et al.*, 2008) and can be expressed as:

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{1.14}$$

Here, l is the luminance (used to compare the brightness between two images), c is the contrast (used to differ the ranges between the brightest and darkest regions of two images) and s is the structure (used to compare the local luminance pattern between two images to find the similarity or dissimilarity of the images) (Sara *et al.*, 2019) and the α, β and γ parameters are used to define the relative importance of the three components (Wang *et al.*, 2003).

The luminance, contrast and structure of an image can be expressed separately as (Brooks *et al.*, 2008):

$$\begin{aligned} l(x, y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x, y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x, y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \tag{1.15}$$

where μ_x and μ_y are the local means, σ_x and σ_y are the standard deviations and σ_{xy} is the cross-covariance for images x and y sequentially. C_1, C_2 and C_3 in the above calculation equations are constant terms added to avoid instability caused when the denominator is close to zero. If $\alpha = \beta = \gamma = 1$, then the index is simplified as the following form using Equations (1.15):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{1.16}$$

SSIM has symmetry, boundedness (not more than 1) and maximum uniqueness properties. It measures the similarity between the original image/video and the recovered image/video. There is an advanced version of SSIM called Multi Scale Structural Similarity Index Method (MS-SSIM) that evaluates various structural similarity images at different image scales (Wang *et al.*, 2003).

In MS-SSIM, two images are compared to the scale of same size and resolutions (Sara *et al.*, 2019). Like SSIM, changes in luminance, contrast and structure are considered to calculate multi scale structural similarity between two images (Dosselmann & Yang, 2011). The MS-SSIM index is also extended to video by applying it frame-by-frame on the luminance component of the video and the overall MS-SSIM index for the video was computed as the average of the frame level quality scores (Seshadrinathan *et al.*, 2010).

THe block diagram for the MS-SSIM method is illustrated in Figure 1.8. Taking the reference and distorted image signals as the input, the system iteratively applies a low-pass filter and downsamples the filtered image by a factor of 2. We index the original image as Scale 1, and the highest scale as Scale M, which is obtained after $M-1$ iterations. At the j-th scale, the contrast comparison and the structure comparison are calculated and denoted as $c_j(x,y)$ and $s_j(x,y)$, respectively. The luminance comparison is computed only at Scale M and is denoted as $l_M(x,y)$. The overall MS-SSIM evaluation is obtained by combining the measurement at different scales using:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l_m(\mathbf{x},\mathbf{y})]^{\alpha M} \cdot \prod_{j=1}^{M} [\mathrm{c}_j(\mathbf{x},\mathbf{y})]^{\beta_j} [s_j(\mathbf{x},\mathbf{y})]^{\gamma_j} \tag{1.17}$$

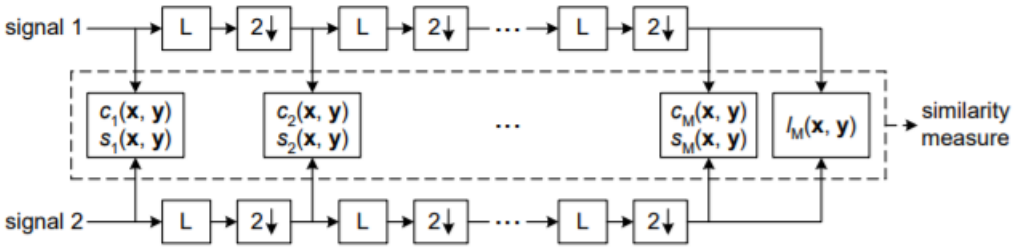


Figure 1.8 Multi-scale structural similarity measurement system. L: low-pass filtering; 2↓: downsampling by 2. (Wang *et al.*, 2003)

Similar to 1.14, the exponents α_M, β_j and γ_j are used to adjust the relative importance of different components.

C. Video Multi-Method Assessment Fusion (VMAF)

The VMAF is a set of objective evaluation indicators developed by Netflix to solve the situation that traditional indicators cannot deal efficiently with video contents which are generally composed of multiple scenes with variable spatio-temporal characteristics. The first to disclose this indicator was on Netflix's technical blog (Li *et al.*, 2016). The VMAF indicator is based on machine learning methods for classification and prediction. By fusing the results of basic indicators with different weights and the opinion scores of subjective experiments, the machine learning model is trained and tested for supervised learning. Finally, a objective metric is generated and adapted to a variety of sources, underlying characteristics, and distortion types.

The current version of the VMAF metric (Li *et al.*, 2016) uses the two following elementary metrics fused by Support Vector Machine (SVM) regression (Cortes & Vapnik, 1995):

- Visual Information Fidelity (VIF) (Sheikh & Bovik, 2006). VIF is a well-adopted image quality metric based on the premise that quality is complementary to the measure of information fidelity loss. In its original form, the VIF score is measured as a loss of fidelity combining four scales. In VMAF, a modified version of VIF is adopted where the loss of fidelity in each scale is included as an elementary metric.
- Detail Loss Metric (DLM) (Li *et al.*, 2011). DLM is an image quality metric based on the rationale of separately measuring the loss of details which affects the content visibility, and the redundant impairment which distracts viewer attention. The original metric combines both DLM and additive impairment measure (AIM) to yield a final score. In VMAF, only the DLM is considered as an elementary metric. Particular care was taken for special cases, such as black frames, where numerical calculations for the original formulation break down.

The final VMAF metric is a full-reference, perceptual video quality metric that aims to approximate human perception of video quality. This metric is mainly focused on quality degradation due to compression and rescaling (Rassool, 2017). The VMAF framework is general and allows for others to retrain it for their own use-case. In the Figure 1.9, it shows how the VMAF indicator work generally.

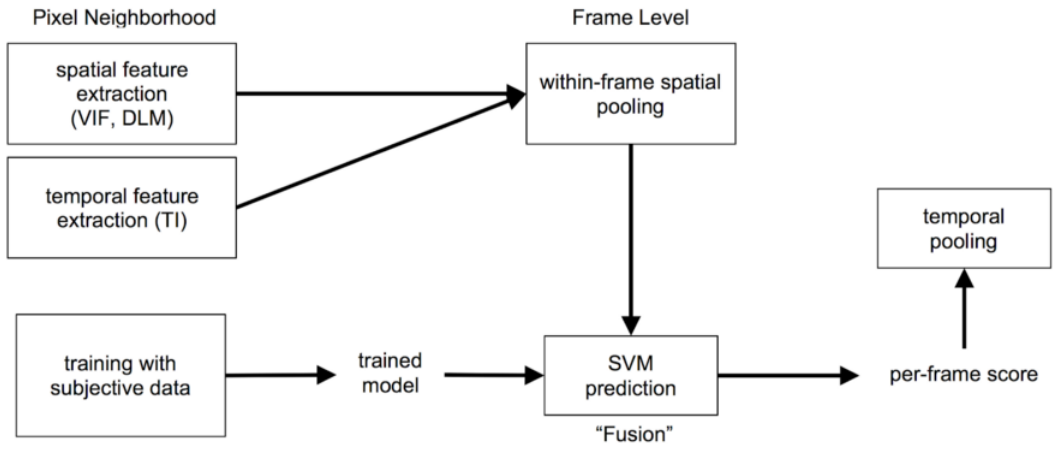


Figure 1.9 How VMAF works: pixel level data are pooled to create frame-level features; different spatial and temporal features are fused using SVM regression to create frame-level quality score; consecutive frame scores are pooled to produce final video sequence VMAF score (Ioannis *et al.*, 2018)

VMAF is based on machine learning learning and is able to integrate a variety of objective indicators. Moreover, the metric's accuracy can be improved through continuous training with continuously expanding data sets, and it can also replace various machines learning methods to reduce regression errors (Xinghao). This makes its performance better than traditional video quality assessment indicators in some cases, which also makes it one of the most popular quality assessment indicators in recent years.

1.3.1.2 No-reference VQA metrics

The NR video quality assessment method does not require any information from the original reference video when evaluating the video quality. It extracts the characteristics of the distorted video by processing and analyzing the distorted video in the spatial and temporal domains, or obtains the video quality based on a predefined quality modelin the pixel domain. No-reference IQA metrics are suitable for wireless and IP video services for which no original reference video sequences are available online.

A. Blind image spatial quality evaluator

Blind/referenceless image spatial quality evaluator (BRISQUE) is a spatial approach to NR IQA developed by Mittal *et al.* in 2012. Like other IQA metrics, it also can be used to evaluate video quality on a frame-by-frame basis. It can be summarized as follows. Given an image (possibly distorted), it first computes locally normalized luminances via local mean subtraction and normalization. Applying a local non-linear operation to log-contrast luminances to remove local mean displacements from zero log-contrast and to normalize the local variance of the log contrast has a decorrelating effect as explained in (Ruderman, 1994). Such an operation (Mittal *et al.*, 2012) may be applied to a given intensity image $I(i, j)$ to produce

$$\hat{I}(i, j) = \frac{I(i, j) - \mu(i, j)}{\sigma(i, j) + C} \tag{1.18}$$

where, $i \in \{1, 2...M\}$, $j \in \{1, 2...N\}$ are spatial indices, $\mu(i, j)$ and $\sigma(i, j)$ are the local mean and the local variance, respectively. M and N are the image height and width respectively, $C = 1$ is a constant that prevents instabilities from occurring when the denominator tends towards zero (e.g., in the case of an image patch corresponding to the plain sky) and

$$\mu(i, j) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} \omega_{k,l} I_{k,l}(i, j) \tag{1.19}$$

$$\sigma(i, j) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} \omega_{k,l} (I_{k,l}(i, j) - \mu(i, j))^2} \tag{1.20}$$

where $\omega = \{\omega_{k,l} | k = -K, ...K, l = -L, ...L\}$ is a 2D circularly-symmetric Gaussian weighting function sampled out to 3 standard deviations and rescaled to unit volume. In this implementation, $K = L = 3$. The author used the pre-processing model (1.18) in the quality assessment model development and refer to the transformed luminances $\hat{I}(i, j)$ as Mean Subtracted Contrast Normalized (MSCN) coefficients.

The author's hypothesis is that the MSCN coefficients have characteristic statistical properties that are changed by the presence of distortion, and that quantifying these changes will make it possible to predict the type of distortion affecting an image as well as its perceptual quality. They

adopt a very general Asymmetric Generalized Gaussian Distribution (AGGD) model (Lasmar *et al.*, 2009) and the AGGD with zero mode is given by:

$$f(x; v, \sigma_l^2, \sigma_r^2) = \begin{cases} \frac{v}{(\beta_l + \beta_r)\Gamma(\frac{1}{v})} exp(-(\frac{-x}{\beta_l})^v) & x < 0 \\ \frac{v}{(\beta_l + \beta_r)\Gamma(\frac{1}{v})} exp(-(\frac{x}{\beta_r})^v) & x \geq 0 \end{cases} \tag{1.21}$$

where

$$\beta_l = \sigma_l \sqrt{\frac{\Gamma(\frac{1}{v})}{\Gamma(\frac{3}{v})}} \tag{1.22}$$

$$\beta_r = \sigma_r \sqrt{\frac{\Gamma(\frac{1}{v})}{\Gamma(\frac{3}{v})}} \tag{1.23}$$

and $\Gamma(\cdot)$ is gamma function:

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \quad a > 0. \tag{1.24}$$

The shape parameter v controls the 'shape' of the distribution while σ_l^2 and σ_r^2 are scale parameters that control the spread on each side of the mode, respectively. The parameters of the AGGD (v, σ_l, σ_r), are estimated using the moment-matching based approach proposed in Lasmar *et al.* (2009). Also, we expect images to have a better separation when modeled in the high dimensional space of parameters obtained by fitting AGGD distributions to paired products from different orientations and scales together. This figure also motivates the use of (1.21) to better capture the finite empirical density function. And the parameters $(\eta, v, \sigma_l, \sigma_r)$ of the best AGGD fit are extracted where η is given by:

$$\eta = (\beta_r - \beta_l) \frac{\Gamma(\frac{2}{v})}{\Gamma(\frac{1}{v})} \tag{1.25}$$

Thus for each paired product, 16 parameters (4 parameters/orientation $\times$ 4 orientations) are computed, yielding the next set of features. Images are naturally multiscale, and distortions

affect image structure across scales. Hence, the authors extract all features listed at two scales - the original image scale, and at a reduced resolution (low pass filtered and downsampled by a factor of 2). They observed that increasing the number of scales beyond 2 did not contribute to performance much. Thus, 36 features at each scale, are used to identify distortions and to perform distortion-specific quality assessment. A mapping is learned from feature space to quality scores using a regression module, yielding a measure of image quality. The framework is generic enough to allow for the use of any regressor. In their implementation, a Support Vector Regressor (SVR) (Schölkopf *et al.*, 2000) is used. SVR has previously been applied to image quality assessment problems (Narwaria & Lin, 2010). They utilize the LIBSVM package (Chang & Lin, 2011) to implement the SVR with a radial basis function kernel.

B. Natural Image Quality Evaluator (Natural Image Quality Evaluator (NIQE))

Objective image quality assessment refers to automatically predict the quality of distorted images as would be perceived by an average human. NR IQA models assume that only the distorted image whose quality is being assessed is available. Many general purpose NR IQA algorithms are based on models that can learn to predict human judgments of image quality from databases of human-rated distorted images. These kinds of IQA models are necessarily limited, since they can only assess quality degradation arising from the distortion types that they have been trained on. NIQE (Mittal *et al.*, 2013) is a blind IQA model that only makes use of measurable deviations from statistical regularities observed in natural images, without training on human-rated distorted images, and, indeed without any exposure to distorted images. This IQA model is based on the construction of a “quality aware” collection of statistical features based on a simple and successful space domain Natural Scene Statistics (NSS) model. These features are derived from a corpus of natural, undistorted images. The authors (Mittal *et al.*, 2013) developed a NSS-based modeling framework for “opinion unaware”(OU) - “distortion unaware”(DU) NR IQA design, resulting in a first of a kind NSS-driven blind OU-DU IQA model which does not require exposure to distorted images a priori, nor any training on human opinion scores.

NIQE model is based on constructing a collection of “quality aware” features and fitting them to a Multivariate Gaussian (MVG) model. The quality aware features are derived from a simple but highly regular NSS model. The quality of a given test image is then expressed as the distance between a MVG fit of the NSS features extracted from the test image, and a MVG model of the quality aware features extracted from the corpus of natural images.

The NSS features used in the NIQE index are similar to those used in BRISQUE model (Mittal *et al.*, 2012). However, NIQE only uses the NSS features from a corpus of natural images while BRISQUE is trained on features obtained from both natural and distorted images and also on human judgments of the quality of these images. By comparison, the NIQE Index is not tied to any specific distortion type.

Then a simple model of the NSS features computed from natural image patches can be obtained by fitting them with an MVG density. NIQE index is applied by computing the 36 identical NSS features from patches of the same size from the image to be quality analyzed, fitting them with the MVG model, then comparing its MVG fit to the natural MVG model. Finally, the quality of the distorted image is expressed as the distance between the quality aware NSS feature model and the MVG fit to the features extracted from the distorted image:

$$D(\nu_1, \nu_2, \Sigma_1, \Sigma_2) = \sqrt{\left((\nu_1 - \nu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\nu_1 - \nu_2)\right)} \tag{1.26}$$

where ν_1, ν_2 and Σ_1, Σ_2 are the mean vectors and covariance matrices of the natural MVG model and the distorted image’s MVG model.

1.3.2 Deep-learning based VQA methods

With the continuous development of deep learning technology, more and more researches focus on applying deep learning to video quality assessment (Vega *et al.*, 2017). Relying on the ability of deep learning neural networks to process images and videos, many video evaluation indicators based on deep learning technologies have emerged in recent years.

Several studies have been proposed aiming for applying CNN in video quality assessment (Giannopoulos *et al.*, 2018; Liu *et al.*, 2018). And some researches also proposed to apply other deep learning models into VQA, such as Recurrent Neural Networks (RNN) model (Chen *et al.*, 2020b) and Transformer (Xing *et al.*, 2021). These deep learning models generally lead to better performance than traditional models in video quality assessment.

1.3.2.1 CNN-based VQA methods

A CNN (LeCun *et al.*, 1995) is an efficient recognition method that has been developed in recent years and has attracted widespread attention. This type of neural network was first introduced by LeCun *et al.* in 1995. Now, CNN has become one of the research hotspots in many scientific fields. Because the network avoids the complicated pre-processing of the image and can directly input the original image, it has been more widely used.

Generally, a CNN model contains three types of layers: convolutional layers, pooling layers and fully-connected layers. Figure 1.10 from O'Shea & Nash (2015) shows a simple architecture of CNN for MNIST database classification which is formed by stacking these layers together.

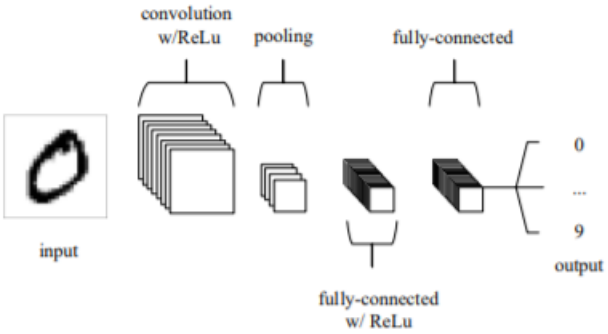


Figure 1.10 A simple CNN architecture, comprised of just five layers (O'Shea & Nash, 2015)

The basic functionality of the CNN example above can be broken down into four key areas.

- The input layer will hold the pixel values of the image.

- The convolutional layer will determine the output of neurons which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. The rectified linear unit (commonly shortened to ReLu) aims to apply an 'elementwise' activation function such as sigmoid to the output of the activation produced by the previous layer.
- The pooling layer will simply perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation.
- The fully connected layers will attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance.

CNNs are able to transform the original input layer by layer using convolutional and downsampling techniques to produce class scores for classification and regression purposes. In what follows, we will explore in detail the individual layers, detailing their parameters and connectivities.

A. Convolutional Layer

The convolutional layers parameters focus on the use of kernels. These kernels are usually small in spatial dimension but expand over the entire depth of the input. When the data reaches the convolutional layer, the layer will convolve each filter in the input spatial dimension to generate a 2D activation map.

As we browse the input, a scalar product will be calculated for each value in that kernel. As shown in Figure 1.11, from this, the network will learn the kernels that will "launch" when a particular feature is seen at a given spatial position of the input. These are usually called activations.

Each kernel will have a corresponding activation map, which is stacked along the depth dimension to form the entire output of the convolutional layer. Each neuron in the convolutional layer is only connected to a small area of the input. The size of this area is usually called the receptive field size of the neuron. The size of the connectivity through the depth is almost always equal to

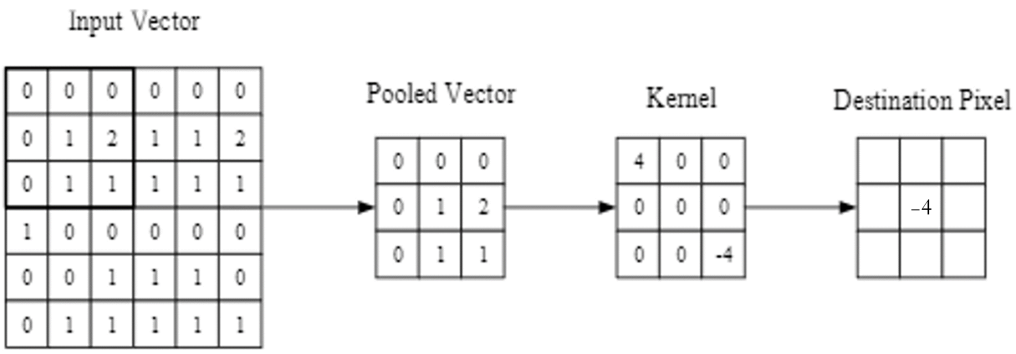


Figure 1.11 A visual representation of a convolutional layer. The centre element of the kernel is placed over the input vector, of which is then calculated and replaced with a weighted sum of itself and any nearby pixels. (O'Shea & Nash, 2015)

the depth of the input. These are optimized by three hyperparameters, depth, stride and setting zero padding.

B. Pooling Layer

The pooling layer aims to gradually reduce the dimensionality of the representation, thereby further reducing the number of parameters and the computational complexity of the model. The pooling layer operates on each activation map in the input and uses the pooling function to scale the dimension. In most CNNs, they appear in the form of the largest pooling layer, the size of the core is 2×2, and the span of the input space size is 2. This reduces the activation map to 25% of its original size while keeping the amount of depth at its standard size. In addition to the maximum pool, the CNN architecture may also include general pools, which perform many common operations, including average pooling and so on.

C. Fully-connected Layer

The fully-connected layer contains neurons of which are directly connected to the neurons in the two adjacent layers, without being connected to any layers within them. The data processed by the convolutional layer and the pooling layer are input into the fully connected layer, and the processed data are classified using softmax to achieve the desired final result. After the data is

reduced by the convolutional layer and the pooling layer, the fully connected layer can "work", otherwise the amount of data is too large, the calculation cost is high, and the efficiency is low.

Different implementations of CNN have shown continuous improvement of accuracy in computer vision (Shrestha & Mahmood, 2019). Here are the well-known variations and implementations of the CNN architecture:

- LeNet-5: devised by LeCun et al. (LeCun *et al.*, 1998) for digit recognition.
- ALexNet: developed by Alex Krizhevsky et al. (Krizhevsky *et al.*, 2012) in 2012 to compete in the ImageNet competition.
- Inception (Szegedy *et al.*, 2015): deep CNN developed by Google.
- VGG (Simonyan & Zisserman, 2014): very deep CNN developed for large scale image recognition.
- ResNet (He *et al.*, 2016): very deep Residual network developed by Microsoft.

Existing deep learning driven models are mainly based on CNN architectures. A typical example is to use CNN as a feature extractor and Multilayer Perceptron (MLP) on the top to predict image/video quality. Other CNN-based metrics (Kang *et al.*, 2014; Zhang *et al.*, 2020) tend to separate an image/video into many small patches and extract the features of each patch to evaluate their quality. Patch-based models often assume that image/video patches share the same quality level as their original full image/video, when training the models (You & Korhonen, 2021).

Although CNN models are very widely used in vision tasks, the convolutional neural networks often require deep networks and a lot of calculations to process video information. Especially for temporal dimension, they often need to be combined with recurrent neural networks, which further deepens the complexity of the model and makes the calculation cost and the time cost rises a lot.

1.3.2.2 Transformer based VQA methods

In 2017, Google proposed a new model called Transformer (Vaswani *et al.*, 2017), instead of the CNN and RNN used in previous deep learning tasks. This model is widely used in the field of Natural Language Processing (NLP), such as machine translation, question answering systems, text summarization and speech recognition, etc.

Transformer is a new simple network architecture, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. The so-called attention mechanism aims to filter out a small amount of important information from the input data and then focus on this important information, ignoring most of the unimportant remaining information. Transformer is a model architecture avoiding recurrence and instead relying entirely on an attention mechanism to establish global dependencies between input and output.

Most competitive neural sequence transduction models have an encoder-decoder structure. Here, the encoder maps an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. Given z, the decoder then generates an output sequence $(y_1, ..., y_m)$ of symbols one element at a time. At each step the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, as illustrated in the left and right halves of Figure 1.12, respectively.

The encoder is composed of a stack of N identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network. The decoder is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

The attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed

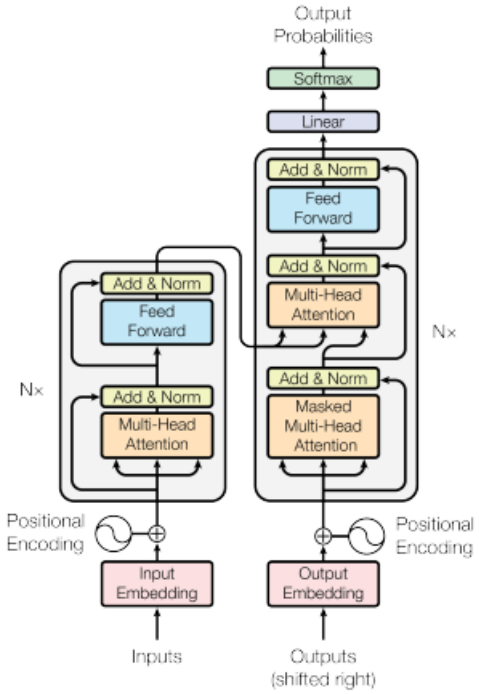


Figure 1.12 The Transformer-model architecture (Vaswani *et al.*, 2017)

as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. In this model, there are 2 attention used: Scaled Dot-Product Attention and Multi-Head Attention (Vaswani *et al.*, 2017). The structures of these two attention mechanisms are shown in the Figure 1.13.

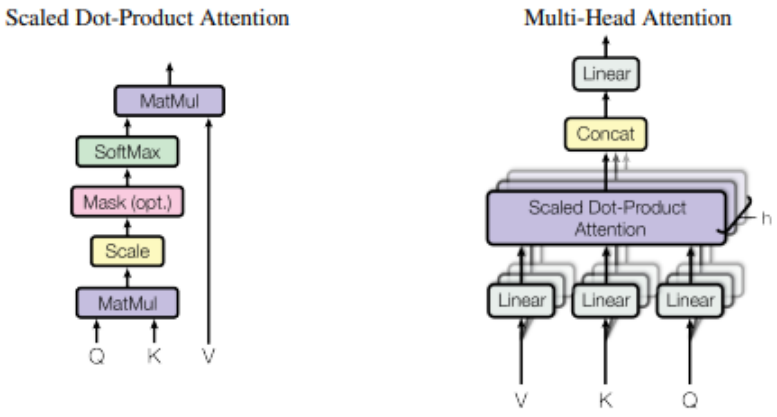


Figure 1.13 Scaled Dot-Product Attention (left). Multi-Head Attention consists of several attention layers running in parallel (right) (Vaswani *et al.*, 2017)

In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between. The model use learned embeddings to convert the input tokens and output tokens to vectors of dimension d_{model}. It also uses the usual learned linear transformation and softmax function to convert the decoder output to predicted next-token probabilities.

This model is the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention. For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers. In the past two years, more and more transformer structures have been applied to image and video processing, and some image/video quality evaluation models based on the transformer model have also emerged.

In particular, ViT (Dosovitskiy *et al.*, 2020) has proposed a very direct application of the Transformer model in image recognition, which allows the Transformer model to be used independently instead of a model mixed with other neural networks such as CNN. A hybrid model is also mentioned in ViT (Dosovitskiy *et al.*, 2020), using CNN to extract feature maps, and then inputting the feature maps into the Transformer encoder for processing.

With the large number of applications and the rapid development of the Transformer model in the field of image processing, numerous methods for evaluating image/video quality based on the Transformer model have appeared over the last three years (Chen *et al.*, 2020a; Cheon *et al.*, 2021; Golestaneh *et al.*, 2021; You & Korhonen, 2021; Yang *et al.*, 2022; Xu *et al.*, 2023). These models proposed to cut the whole image on several patches, or use CNN convolution process on the image to extract feature maps, then flatten them and enter them into the Transformer encoder so that the Transformer can learn these feature maps and assess image quality. By using CNN to rapidly extract the features and attention mechanism included in the Transformer, these methods improved the efficiency of processing small amounts of image data and assessing image quality.

At the same time, the experiments presented in these papers also demonstrate that the results of evaluating this mixed objective model are more consistent with human visual perception.

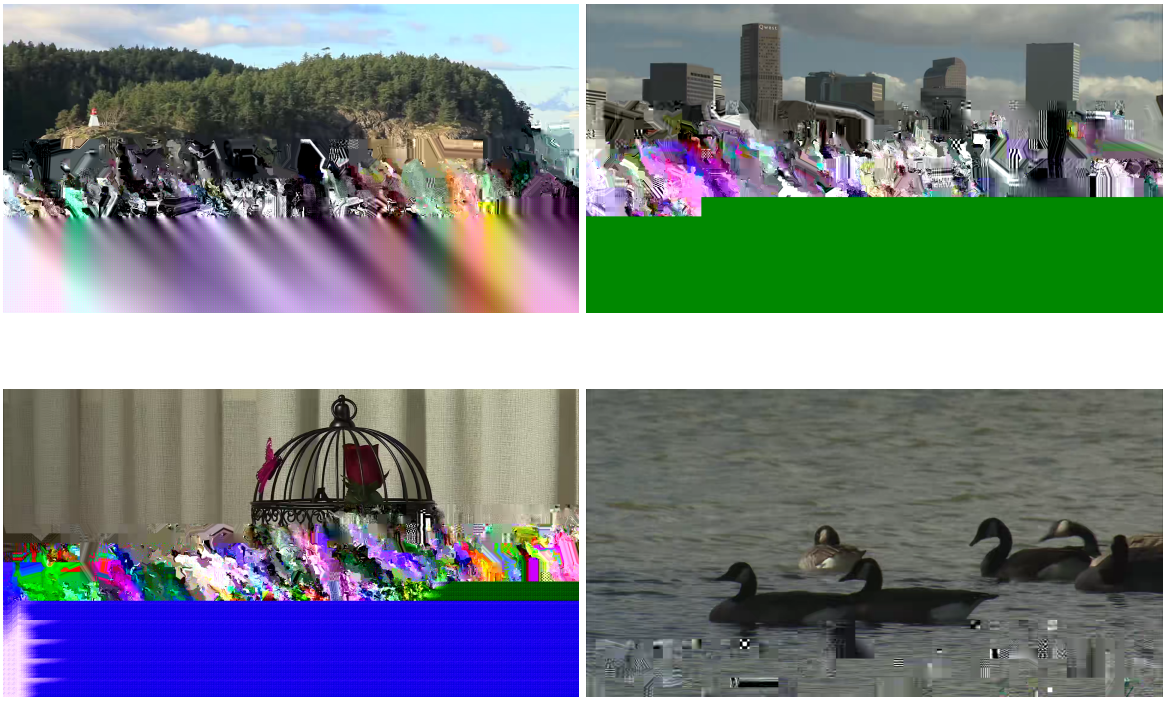
Xing *et al.* (2021) proposed a novel space-time attention network for the VQA problem, named StarVQA. StarVQA builds a Transformer by alternately concatenating the divided space-time attention. To adapt the Transformer architecture for training, StarVQA designs a vectorized regression loss by encoding the MOS to the probability vector and embedding a special vectorized label token as the learnable variable. To capture the long-range spatiotemporal dependencies of a video sequence, StarVQA encodes the space-time position information of each patch to the input of the Transformer. The authors proved that StarVQA is very situable for the defacto in-the-wild video datasets and high-resolution videos.

1.3.3 Analysis of the existing VQA methods

Although the full reference VQA methods can more accurately evaluate the video quality, they require that the original video is available. In a real video transmission context, the original video is not available on the receiving side, which avoids the use of full-reference video quality metrics. Therefore, our research concentrates more on the NR VQA methods. In this Section, we first show the distortion caused by the transmission error, then we show the simulation results based on the existing methods to describe they are not the methods we want to find. Finally, we conclude that we need to propose a new proper VQA method for our research.

1.3.3.1 Examples with the distortion caused by the transmission error

As shown in Figure 1.14, we notice that the great diversity of visual artifacts due to transmission errors, with totally different and unpredictable visual renderings, makes them difficult to predict and modelable.



1.3.3.2 Analysis and the experimental results of the existing methods

For the NR metrics mentioned in the previous sections, BRISQUE (Mittal *et al.*, 2012) and NIQE (Mittal *et al.*, 2013), are two typical well-known non-reference metrics which do not use deep-learning algorithms. Both use the natural scene statistics model to extract video features from the natural videos. One of the main ideas of NR VQA by using natural scene statistics is that natural videos exhibit certain statistical regularities that can be altered in the presence of



Figure 1.14 Examples of the sequences with the distortion caused by the transmission error

distortions. The video quality can be estimated by extracting features that indicate how far these statistics deviate in the distorted videos. This also raises the question of how many exemplar videos are needed to design an accurate natural video model, and how diverse and distinctive these need to be relative to each other.

With the development of deep learning architectures, there are plenty of different no-reference VQA metrics proposed in these years. There are also NR VQA algorithms based on generic types that do not detect a specific type of distortion, they usually transform the VQA problem into a classification or regression problem, where the classification and regression are trained using specific features. Relevant features are either extracted using natural scene statistics or discovered through machine learning and deep learning.

Patch-based models often assign all patches in the image the same quality level as the full image when learning (Kang *et al.*, 2014; You & Korhonen, 2021). These patch-based solutions give good results when the distortions are evenly across the entire visual signal, but are not a desirable approach when considering non-uniform distortions. Because these methods tend to calculate a global rather than local quality of the video content, they sometimes minimize the visual impact of some local errors that strongly degrade the visual quality. In a video with uneven error distribution, when these methods integrate the quality scores of all patches and perform regression analysis to obtain an overall score, some local errors in the patch will be underestimated, and the overall quality score of the video may be very high, but these local errors may be very obvious in human vision, thus strongly affecting the final viewer's judgment.

As mentioned in sections before, there are some new deep-learning architectures applied in video quality assessment in recent years, such as Transformer (Vaswani *et al.*, 2017) proposed by the Google team in 2017. Transformers utilize an encoder-decoder architecture fully relying on self-attention to compute latent representations without the need for recurrent mechanisms or convolutions. This type of metric has the advantage of dealing with temporal information, which is particularly effective in modelling the long-term dependency of sequential languages, such as videos or sentences.

Transformer models have advantages in processing the time series and scale well for large datasets and complex models. They can be easily trained in parallel, making up for their natural applicability for a wide variety of tasks (Chubarau & Clark, 2021). But these types of metrics are also "data hungry" models, that is to say, they need plenty of data to train the model to get better performance. It performs very well on vision tasks if the training dataset is sufficiently large. Transformer models achieve higher accuracy than CNN structures when increasing the model size and allowing for higher computational cost. But convolution is still the best choice in designing light weight models (Zhao *et al.*, 2021). Due to the great success of the Transformer model in the natural language processing domain, there are several works on applying Transformers to the computer vision field. Some researchers proposed VQA metrics based on the combination of Transformer and CNN. CNN and Transformer are complementary in the sense that the convolution structure has the best generalization capability while the Transformer structure has the largest model capacity among the three structures.

However, most NR VQA algorithms focus on detecting specific types of distortions such as blur, blockiness, and various forms of noise. These algorithms have demonstrated strong performance in assessing the quality of videos with specific, uniformly distributed distortions. As previously analyzed, these metrics often excel at evaluating the quality of videos with uniform errors, as they are sensitive to artificially synthesized and evenly distributed distortions. Our research aims to better evaluate video quality with non-uniformly distributed distortions, which differ significantly from uniform distortions. In such cases, visual distortions appear randomly and different video blocks typically exhibit varying degrees of distortion. For example, videos transmitted over unreliable networks are prone to non-uniform transmission errors. Unfortunately, most existing metrics do not perform well in evaluating video quality under these conditions. Severe local distorted information is often ignored during evaluation, leading to inconsistencies with human visual perception and resulting in deviations.

In our research, we focus on the distortions introduced by channel errors during video transmission. Transmission errors caused by bit flipping or packet loss generally do not affect the entire visual content in the same manner after video decoding, but result in random inhomogeneous

visual artifacts which are difficult to predict or model. In this case, most global-evaluated VQA models will produce large errors or fail to detect such transmission errors during video quality evaluation.

In order to develop our solution, we build a database with defined HEVC encoding based on the original YUV video sequences collected from public datasets (xip; Wang *et al.*, 2016; Pinson, 2013). Most existing datasets for image quality assessment focus on artificially synthesized losses or user-generated losses, but lack datasets that include different types of non-uniform distortions caused by a large number of transmission errors. Therefore, we create the scripts and instructions to regenerate the database, including the standard HEVC (Sze & al, 2014)) encoding and addition of transmission errors to obtain non-uniform corrupted frames. Simple error patterns are applied to the encoded video packets to simulate transmission over error-prone networks. We collect the combination of $p \times p$ patches, which is called "super-patch" in the corrupted frames from these decoded video bitstreams, to do the training and testing with the groud-truth neighbourhood-based patch fidelity aggregation scores. We collect 90 original videos with 1920×1080 resolution from public datasets. All videos are encoded and decoded without error concealment to show the different distortions.

We do the test on the existing models and on our database which has the distortion caused by transmission error. We use several metrics to evaluate the performance of the models with different configurations. As shown below, $\overline{S}_{\text{intact}}$ indicates the average PSNR, relative to the original versions, of all intact frames, which are compressed but received without transmission errors. Here S is the PSNR score calculated on the RGB color space. $\overline{S}_{\text{system}}$ represents the average PSNR, relative to the original versions, of all images selected by a given method. $\overline{S}_{\text{diff}}$ is, for a method, the absolute difference between the average quality of the selected images and that of the intact images.

From Table 1.1 and 1.2, we can clearly see that the fact that traditional metrics cannot distinguish distortions distributed "uniformly" over the entire image which may not visually interfere too much with more severe errors such as transmission errors.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
PIQE (Venkatanath *et al.* (2015))	30.4%	34.76	19.84	14.92
NIQE (Mittal *et al.* (2013))	5.4%		16.30	18.46
BRISQUE (Mittal *et al.* (2012))	19.6%		18.29	16.47
CNN_NR_IQA (Kang *et al.* (2014))	46.4%		24.78	9.98
MANIQA (Yang *et al.* (2022))	75.0%		28.84	5.92

Table 1.1 Performance on intra-coded images with the existing methods.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
PIQE (Venkatanath *et al.* (2015))	28.6%	34.22	24.38	9.84
NIQE (Mittal *et al.* (2013))	17.9%		26.48	7.74
BRISQUE (Mittal *et al.* (2012))	28.6%		22.75	11.47
CNN_NR_IQA (Kang *et al.* (2014))	33.9%		28.78	5.44
MANIQA (Yang *et al.* (2022))	55.4%		27.67	6.55

Table 1.2 Performance on inter-coded images with the existing methods.

Therefore, it is necessary to develop a new method for automatically selecting the highest quality video from the candidate list. Consequently, we aim to select the best candidate based on the visual quality determined by a deep-learning (DL) system as illustrated in the right part in Figure 1.7b). Considering that the error will be managed on each frame, we will consider image quality assessment rather than video quality assessment. Our proposed system is sensitive to the non-uniform distributed distortions in the image. More specifically, each candidate will undergo processing by a DL-based NR IQA method to obtain a score. Subsequently, the system will select the candidate with the highest IQA score.

This new deep-learning assisted video list decoding system which has a visual quality evaluation framework using deep-learning metrics to identify the best candidate from the candidate list. It is for assessing the quality of videos subject to transmission errors where we do not discard lost packets, conceal lost regions and assess quality.

CHAPTER 2

PROPOSED CNN-ASSISTED VIDEO LIST DECODING SYSTEM

2.1 Introduction

We are witnessing a very rapid development of applications involving the transmission of video contents. However, transmission errors over wireless networks seriously compromise the visual quality of reconstructed video, which results in a poor quality of experience for the end user. Different approaches exist in the literature to repair erroneous video packets received (Wang & Zhu, 1998; Kung *et al.*, 2006; Liu *et al.*, 2020; Lin *et al.*, 2022; Boussard *et al.*, 2020a; Sabeva & al, 2006). Among these, we are interested in list decoding approaches which exploit corrupted received packets. From each corrupted packet, the method generates several *candidate* packets. These candidates represent various attempts to correct the erroneous packet. The challenge consists of estimating at the receiver side the quality of each of these candidates without reference and then choosing the best. The latter will ideally correspond to the intact version originally transmitted.

Previous list decoding has use various criteria such as LLR (Balatsoukas-Stimming *et al.*, 2015), and first decodable stream. Typically list decoding, which generates a list of candidate corrected videos from the corrupted received packets and decides the best one based on criteria such as bit reliability (e.g. LLR), first decodable stream, etc., but there is no certainty or validation that the selected version is actually of good quality. For example, in previous work, the authors proposed applying checksum validation methods (Golaghazadeh *et al.*, 2018) to reduce the number of candidates in the list, but were unable to identify the best candidate. They still selected the first decodable candidate in the list after selection as the best video candidate. It is therefore necessary to build a new method for selecting the best quality video from the candidate list, which can now be decided with the proposed method on a new criterion which is the visual quality as assessed by a deep learning system. Therefore, we want to select the right version based on the resulting visual quality as assessed by a deep-learning system.

In this chapter, we propose a DL-assisted video list decoding framework where a reference-free evaluation of visual quality makes it possible to identify the best candidate among a list of several ones. Our approach is based on the use of a modified CNN to allow the consideration of non-uniform distortions due to transmission errors. The new quality assessment method is based on the CNN presented in(Kang *et al.*, 2014), but improved in several respects, including patch-based local normalization and quality measurement to support non- uniform in the images.

We also introduce a new database made up of videos encoded with the HEVC (Sze & al, 2014) standard and to which we have injected transmission errors. This leads to images with non-uniformly spatially distributed artifacts on which our system can train.

We first introduce our proposed DL-assisted video list decoding framework in section 2.2. In Section 2.3, we present the proposed CNN-based IQA method. We present our database created with transmission error patterns in Section 2.4. In Section 2.5, we present our experimental results with this CNN-assisted system.

2.2 Proposed DL-assisted video list decoding system

Inspired by the previous works (Kang *et al.*, 2014; Bosse *et al.*, 2016; Zhang *et al.*, 2020; Kossi *et al.*, 2022), we propose a deep-learning assisted video list decoding framework where a CNN-based image quality estimation metric is applied, as shown in Figure 2.1. The proposed DL-assisted framework consists of: 1) generating a database of images with non-uniform distortions, 2) learning for quality assessment (training), and 3) selection of the best candidate (inference). The database generation process includes different steps: video encoding by the HEVC standard, generation of transmission errors and video decoding by list, without error concealment, to obtain the N candidates representing mostly unsuccessful attempts to fix the video. The training process to assess quality includes converting images (from YUV format to the format used during training), generating patches and training the neural network in a supervised manner using a quality metric with complete reference.

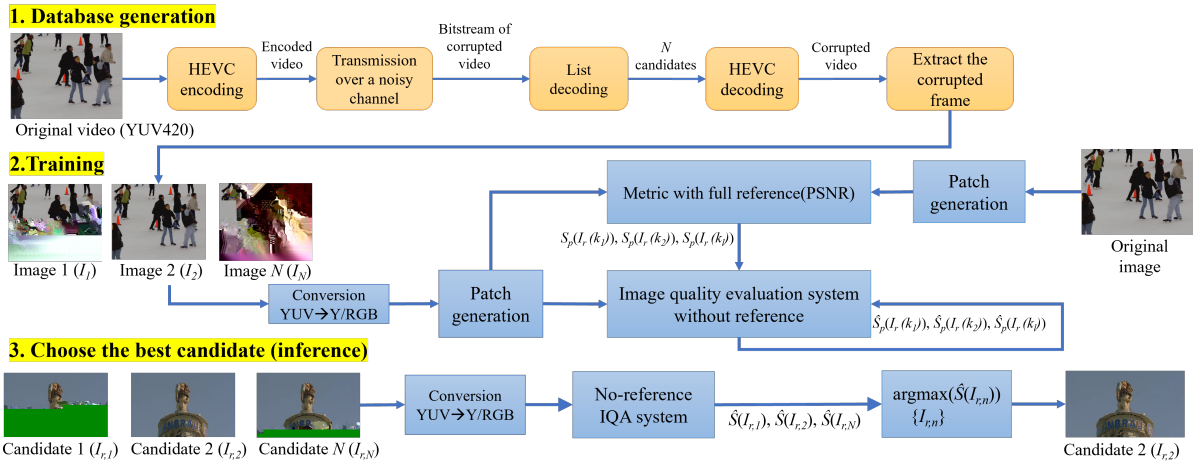


Figure 2.1 Proposed framework to optimize list decoding of corrupted videos during transmission (Zhang *et al.*, 2023).

The general inference process of the deep-learning based candidate selection within list decoding is shown in Figure 2.2. Given a list of reconstructed candidate images $I_{r,n}$, for $n \in [1, N]$, where N is the total number of candidates in this list. The NR deep-learning based IQA system is used to estimate the quality score of each candidate image, which is represented by $\hat{S}(I_{r,n})$. The candidate with the highest estimated quality is calculated by:

$$I_{r,\text{best}} = \underset{\{I_{r,n},\ 1 \le n \le N\}}{\arg\max}\ \hat{S}(I_{r,n}) \tag{2.1}$$

2.3 Proposed CNN-based visual quality evaluation method

Several CNN-based metrics (Kang *et al.*, 2014; Bosse *et al.*, 2016) separate an image into several small patches and extract features from each patch to evaluate their quality. Patch-based models often assign all patches in the image the same quality level as the full image when trained (You & Korhonen, 2021), which gives good results for uniform distortions, but is not a desirable approach when considering non-uniform distortions. We therefore propose to use local scores so that each patch has its own quality score. This can help the neural network learn local distortions more efficiently.

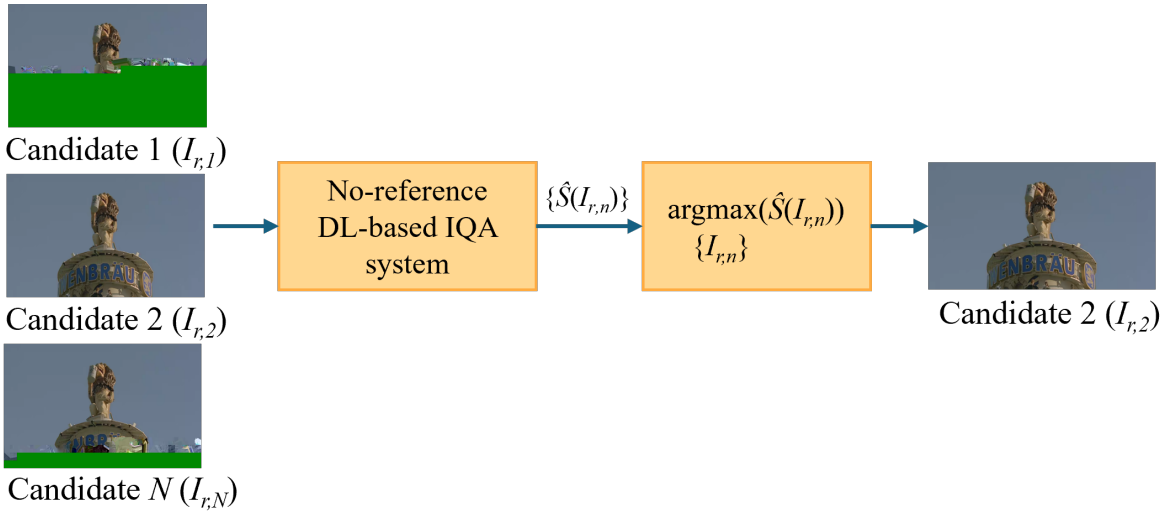


Figure 2.2 The general inference process of the deep-learning based candidate selection within the proposed video list decoding framework.

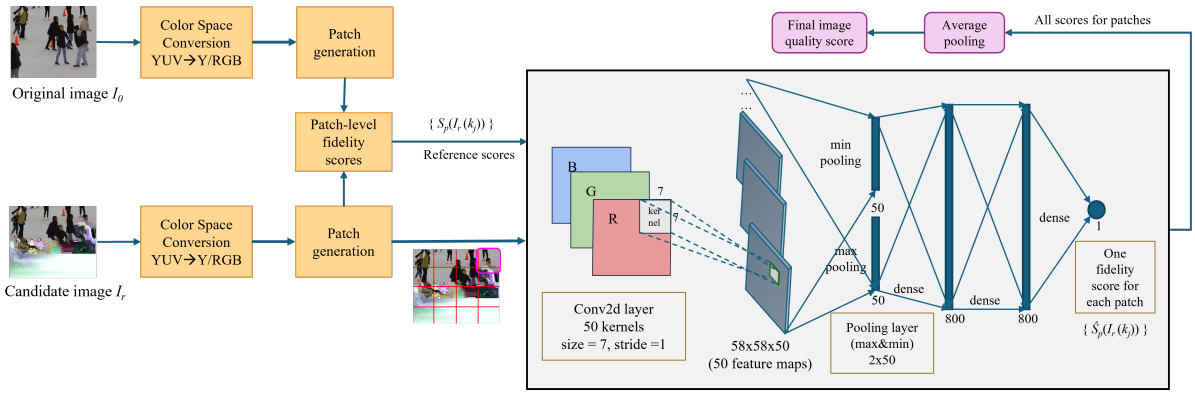


Figure 2.3 The proposed CNN-based IQA metric for ranking video candidates in list decoding(Zhang *et al.*, 2023).

We choose the CNN architecture proposed in Kang *et al.* (2014) as the base architecture and improve it in order to adapt it to our objective. The original architecture is a 5-layer neural network, including 1 convolutional layer, 2 pooling layers and 2 fully connected layers. As shown in Figure 2.3, in this article we use the following network structure: $64\times64\times3 - 58\times58\times50 - 2\times50 - 800 - 800 - 1$. Instead of performing the simulation only on the luminance channel as in Kang *et al.* (2014), we extend our experiments the three R, G, B channels.

2.3.1 The inability to distinguish between various sources of uniform blocks

The basic architecture (Kang *et al.*, 2014) uses a local contrast normalization method. Suppose the intensity value of a pixel at location (i, j) is $v(i, j)$, then the authors calculate its normalized value $v_n(i, j)$ as follows:

$$v_n(i,j) = \frac{v(i,j) - \mu(i,j)}{\sigma(i,j) + C}, \text{with}$$
$$\mu(i,j) = \frac{1}{(2W+1)^2} \sum_{p=-W}^{p=W} \sum_{q=-W}^{q=W} v(i+p, j+q)$$
$$\sigma(i,j) = \sqrt{\frac{1}{(2W+1)^2} \sum_{p=-W}^{p=W} \sum_{q=-W}^{q=W} [v(i+p, j+q) - \mu(i,j)]^2}$$

where C is a positive constant that prevents division by zero. The size of the normalization window is $(2W + 1) \times (2W + 1)$ pixels with $W = 3$.

However, this method poses a problem when applied to uniform patches. As shown in Figure 2.4, *visual patch 1* is an erroneous patch initialized to 0 by the decoder and *visual patch 2* is a well-received uniform patch. Both patches are uniform but with different ground-truth quality scores.

As shown in Figure 2.5, considering a single channel, for example, Y, we cannot distinguish between a well-received uniform patch whose value is normalized to 0 and an erroneous patch which is uniform because it was initialized to 0 by the decoder. This situation is problematic when it occurs in the training database because the neural network becomes confused during training. Indeed, after normalization, a uniform patch and an error patch become identical and enter the layers of the CNN with different reference scores for learning.

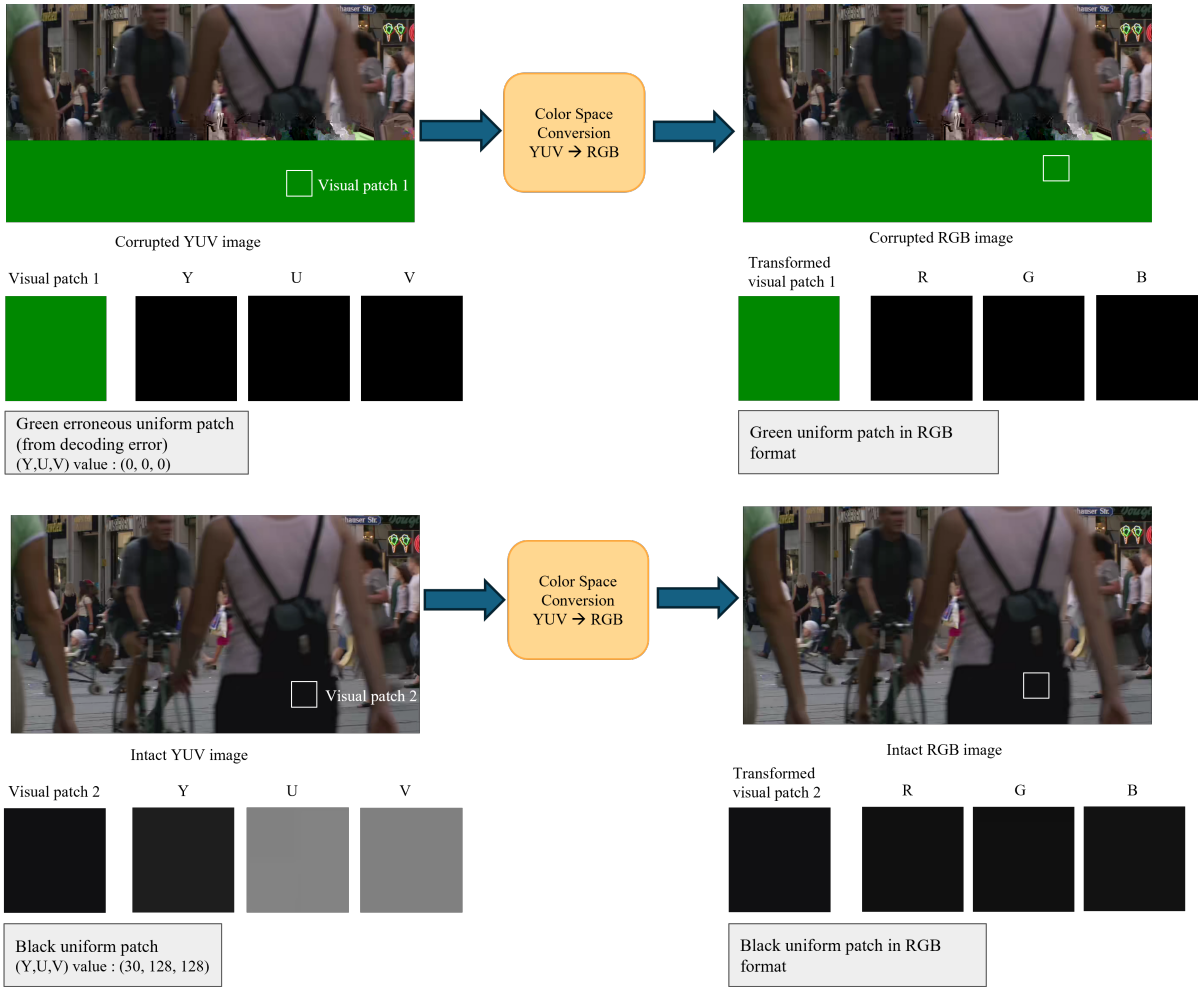


Figure 2.4 An example of two different uniformed patches with color space conversion from YUV420 to RGB

2.3.2 Proposed CNN-based method with improved local normalization

To avoid this problem, as shown in Figure 2.6, we improve local normalization by separating these two situations: a well-received uniform patch whose value is normalized to 0 and an erroneous patch that is uniform because it has been initialized to 0 by the decoder. When we detect $\sigma(i, j) = 0$ in the input patches, we calculate $\mu(i, j)$. If $\mu(i, j) \neq 0$, we force $v_n(i, j)$ to be equal to $\epsilon \neq 0$ after normalization. We use Eq.(2.2) on each channel of an image in RGB format where we force the value to (0,0,0) when the decoder recovers YUV at (0,0,0) following an error.

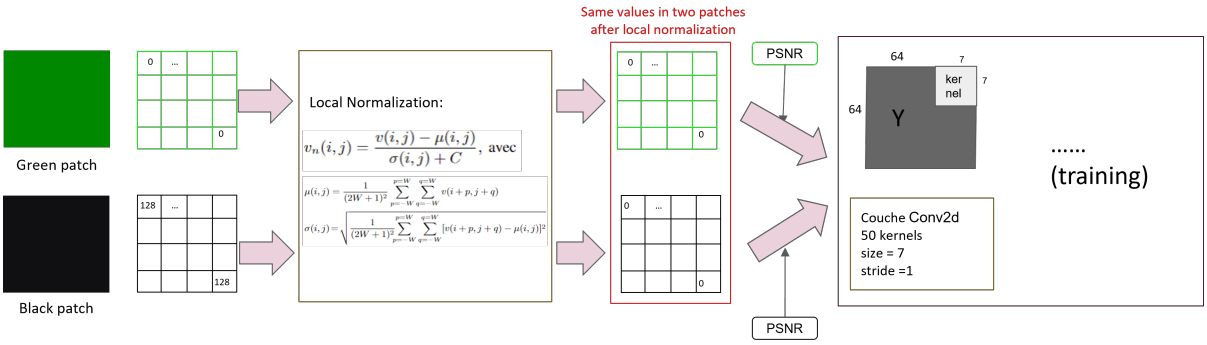


Figure 2.5 The problem of the original local normalization method in Kang *et al.* (2014)

$$v_n(i,j) = \begin{cases} 0, & \text{if } \sigma(i,j) = 0 \text{ and } \mu(i,j) = 0 \\ \epsilon, & \text{si } \sigma(i,j) = 0 \text{ and } \mu(i,j) \neq 0 \end{cases} \tag{2.2}$$

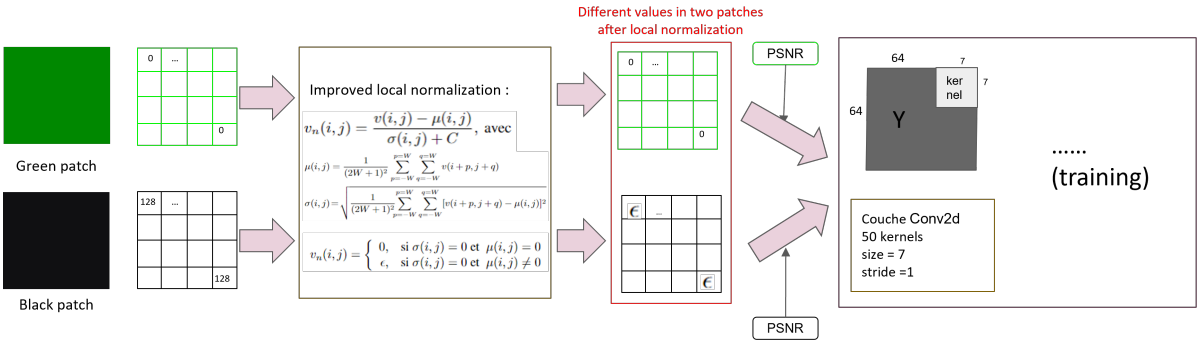


Figure 2.6 The improved local normalization method in Zhang *et al.* (2023)

Our neural network is first trained on non-overlapping patches of 64×64 pixels, corresponding to a CTU in HEVC (Sze & al, 2014), coming from high definition images. For training, we assign to each patch a PSNR quality score, calculated between the corrupted patch and the corresponding patch in the original image before encoding (metric with reference). For testing, we use the average of the predicted patch scores for each image I_r to obtain the image-level quality score $\hat{S}(I_r)$:

$$\hat{S}(I_r) = \frac{1}{L} \sum_{l=1}^{L} \hat{S}_p(I_r(k)) \tag{2.3}$$

where $\hat{S}_p(I_r(k))$ indicates the predicted quality score for patch k of image I_r by the CNN (reference-free metric), and L is the total number of patches in the image. Using small patches as input greatly enlarges the training sample for the CNN and avoids the lack of data problem encountered when using a full image set. We use the (Kang *et al.*, 2014) loss function, stochastic gradient descent and back-propagation are used during training. We use a validation set to avoid over-fitting and keep the model parameters that generate the highest Spearman Rank Order Correlation Coefficient (SROCC) value on the validation set.

2.4 The video database

2.4.1 Creation of the database

Most existing datasets for image quality assessment focus on artificially synthesized (simulated) distortions found in user-generated content (Sheikh, 2005; Ponomarenko *et al.*, 2013; Lin *et al.*, 2019). However, they lack datasets that encompass various types of non-uniform distortions resulting from transmission errors, where the content is decoded without error concealment. Indeed, the LIVE database (Sheikh, 2005) contains distortions resulting from transmission errors, but the erroneous regions are discarded and concealed rather than decoded and rendered as in our case. Therefore, we developed some scripts and instructions for the desired database. The database generation process is shown in Figure 2.7. The process includes several steps: video encoding by the HEVC standard, generating transmission errors by flipping bits in specific positions in the video bitstreams and video list decoding, without error concealment, to obtain the N candidates representing various unsuccessful attempts to correct the video. After decoding all the video candidates, we extract the corrupted frames in each video sequence if decodable and add them to our database.



Figure 2.7 The database generation process.

We now provide more details on the generation process. We use the original sequences from the public datasets (xip; Wang *et al.*, 2016; Pinson, 2013). The collected videos are in YUV format with a resolution of 1920×1024. We extract the first 10 frames from each video to encode them with the HEVC standard (Sze & al, 2014). Among the different possible QP values, we chose 37 and 22, which correspond, respectively, to the low and high bit rate operating points of the HEVC standard reference software (HM) Common Test Conditions (Bossen, 2013). We assume that each encoded frame is contained in a single video packet. The first frame of the encoded video is an intra (I) frame, and the next 9 frames are inter (P) frames.

We want to simulate the combination of a transmission error followed by list decoding where bits are inverted at different locations, i.e., to spread the error throughout the video frame from the beginning to the end. However, using a network simulator is highly complex, and the results vary significantly depending on the type of network and transmission conditions. To simplify the process, we place bit errors at various locations in the packet. This method is compatible with list decoding scenarios, where the bits altered to generate candidates appear in random-like, unpredictable locations. For instance, in CRC-based error correction, as mentioned in (Boussard *et al.*, 2020a), the candidates exhibit patterns with bits altered in such unpredictable locations. The main objective in developing the database is not to obtain an accurate simulation of erroneous packets generated by specific wireless networks but rather to ensure sufficient diversity in the transmission error patterns to adequately train the system. Therefore, we selected flipped (inverted) bit positions based on the equation $pos = \beta \times M$, where $\beta = \{0.1, 0.2, ..., 0.9, 0.99\}$ and M is the size of each packet. We incorporate transmission error patterns separately for intra-coded frames and inter-coded frames, depending on the scenario under study. Consequently, errors in inter-coded frames are directly applied to the frame itself, rather than being propagated from errors in previous frames. The candidate frames subject to these various errors are decoded and added, if decodable, to our database. For each sequence and each frame type, we generate 11 candidates, including one error-free (intact) candidate. This results in 990 corrupted images from 90 reference images.

2.4.2 Examples from our database

Here are examples of images extracted from the video database with non-uniform distortions, which are caused by transmission errors. We can see that the visual impact of channel disturbance can be very different: some images are severely damaged and the details or even structures are no longer recognizable, while others contain subtle degradations that are barely perceptible. The images on the right side exhibit large green flat areas, such visual artifact is typical in digital TV impairments due to transmission errors (green screen effect).



Figure 2.8 Examples of the candidates in our database (Zhang *et al.*, 2023)

2.5 Experimental results

In this section, we present our experimental results. We start by describing the training methodology and evaluation criteria. Then we provide a comprehensive performance evaluation, including results, parameter sensitivity analysis, and discussion. Finally, we give an example of a situation where our system fails to detect the error-free version of the image, showing that even in this case, our system delivers an image of satisfying visual quality.

2.5.1 Training and testing methodologies

Our experiments are conducted using the NVIDIA Quadro RTX 5000 GPU with PyTorch 2.3.0 and CUDA 12.1 for training and testing. We trained and tested our model on our proposed patch based datasets. Our database has about 475 200 non-overlapping patches for each frame type in our simulation, with a patch size of 64×64 pixels. Each patch is associated with a PSNR (Sara *et al.*, 2019) score between the reconstructed version and the original one in the interval [0, 50] dB, which is normalized to the interval [0, 1] during training. PSNR and SSIM scores are both adequate choices for patch-level reference scores. For simplicity, we choose PSNR as the full-reference patch-level fidelity score, as it is a low-complexity widely recognized distortion metric. Following the standard training strategy outlined in existing IQA algorithms (Kang *et al.*, 2014), we randomly split each dataset into 60:40 ratio, with 60% allocated for training, 20% for validation and the remaining 20% for testing. During training, we set the learning rate l to 0.001 and the batch size B to 128. We utilized the ADAM optimizer. Based on empirical simulation results, we set $\epsilon = -0.013$ in Eq.(2.2). The training loss used is the L1 loss (Kang *et al.*, 2014). The final score is generated by averaging the scores predicted for all patches predicted in each image.

Note that we train our system on QP=37 because it represents a higher quantization parameter, which corresponds to a frequently used value in low-bitrate real-time video applications. It also introduces more compression artifacts and creates more severe degradation compared to low QP. This challenging scenario allows the model to learn how to handle significant visual degradation

and error propagation, making it robust and effective in improving visual quality under difficult conditions.

2.5.2 Performance evaluation criteria

We train and test the original CNN model and our improved version on the newly developed database. We use the metrics described in Eq. (2.4) to evaluate the performance of the various IQA models with different configurations. In the equation, $\overline{S}_{\text{intact}}$ indicates the average PSNR between the intact images and the original versions, of all N video sequences, where intact versions are compressed but received without transmission errors. Here S is the PSNR score calculated on the YUV color space, and N represents the total number of video sequences. $\overline{S}_{\text{system}}$ represents the average quality returned by the system, which is calculated by the average PSNR value between the system selected version and the original version, of all N sequences. $\overline{S}_{\text{diff}}$ gives the difference between the quality returned by the system when intact images are selected and by the proposed deep-learning based list decoding system.

$$\begin{aligned} \overline{S}_{\text{intact}} &= \frac{1}{N}\sum_{n=1}^{N} S(I_{\text{o},n}, I_{\text{i},n}), \\ \overline{S}_{\text{system}} &= \frac{1}{N}\sum_{n=1}^{N} S(I_{\text{o},n}, I_{\text{s},n}), \\ \text{where } I_{\text{s}} &= \underset{\{I_{\text{r},i},\ 1\le i\le R\}}{\arg\max}\, \hat{S}(I_{\text{r},i}), \\ \overline{S}_{\text{diff}} &= \left|\overline{S}_{\text{intact}} - \overline{S}_{\text{system}}\right| \end{aligned} \tag{2.4}$$

2.5.3 Results and sensitivity analysis

Tables 2.1 and 2.2 present the experimental results obtained for images encoded in *intra coding* mode and *inter coding* mode respectively, comparing the original CNN method to different simulation configurations. For the results on image *inter*, the error directly hits the inter frame in

question and does not correspond to an error propagation occurring in the previous intra frame. The best results are shown in bold.

Methods	Accurancy	$\overline{S}_{\text{intact}}$ (dB)	$\overline{S}_{\text{system}}$ (dB)	$\overline{S}_{\text{diff}}$ (dB)
CNN_Y_G pre-trained (Kang *et al.*, 2014)	45.6%	39.18	28.69	10.49
CNN_Y proposed	93.0%		38.39	0.79
CNN_RGB proposed	96.5%		38.88	0.30
CNN_Y_NL proposed	94.7%		38.60	0.58
CNN_RGB_NL proposed	**100%**		39.18	**0.00**

Table 2.1 Performance on intra-coded images.

Methods	Accuracy	$\overline{S}_{\text{intact}}$ (dB)	$\overline{S}_{\text{system}}$ (dB)	$\overline{S}_{\text{diff}}$ (dB)
CNN_Y_G pre-trained (Kang *et al.*, 2014)	33.3%	38.62	32.99	5.63
CNN_Y proposed	60.0%		30.19	8.43
CNN_RGB proposed	66.7%		36.49	2.13
CNN_Y_NL proposed	65.0%		34.02	4.60
CNN_RGB_NL proposed	**73.7%**		36.65	**1.97**

Table 2.2 Performance on inter-coded images.

The first row of each table, pre-trained CNN_Y_G, uses the already trained model from the article(Kang *et al.*, 2014), which applies the same score for each patch of the Y component of the image (score global), and tests with our database with non-uniform distortions. Proposed CNN_Y indicates the proposed solution where we use a different score per luminance patch and where we retrain and test on our database (like all proposed methods). We can see the benefit of using a local score per patch and re-training on the proposed database since the precision goes from 46% to 93% on intra-coded images and from 33% to 60% on inter-coded images. We believe that this local score allows us to better learn the characteristics of distortions originating from transmission errors. Proposed CNN_RGB indicates that images initially used in YUV format are converted to RGB format for training and inference. A method with the suffix _NL indicates a configuration that applies our improved local normalization method (Eq.2.2). For intra-encoded images, using local normalization achieves better accuracy and less difference in quality when the CNN uses the Y channel with a precision that goes from 93% to 95% and

from 96.5% to 100% for RGB, which shows an excellent performance on RGB format. For inter images, the use of local normalization makes it possible to significantly improve performance both for Y and for RGB with a precision which goes from 60% to 65% for Y and from 67% to 74% for RGB. Finally, although the performances are similar for Y and RGB in intra frame, the use of RGB format performs better in inter frame. We believe it has the advantage of being able to identify color distortions. We note that the precision is much lower for inter-coded images than for intra-coded images. Indeed, transmission errors in inter-coded images do not generate as significant losses of quality as in intra-coded images, which makes learning the model more difficult.

We can see the benefit of local scoring, re-training on our database and local normalization. However, the performances are not optimal and several works are planned to improve them. For example, we could think about modifying the size of the patches to be able to detect discontinuities at the borders of HEVC CTUs, made visible following the presence of transmission errors. Also, we could adapt our system by operating directly in YUV rather than in Y or RGB to detect color distortions while avoiding additional conversions. We realize the simulations with more configurations and obtain the results in table 2.3 and 2.4.

Methods	Accuracy	$\overline{S}_{\text{intact}}$ (dB)	$\overline{S}_{\text{system}}$ (dB)	$\overline{S}_{\text{diff}}$ (dB)
CNN_Y_G pre-trained (Kang *et al.*, 2014)	45.6%	39.18	28.69	10.49
CNN_Y proposed	93.0%		38.39	0.79
CNN_RGB proposed	96.5%		38.88	0.30
CNN_YUV proposed	94.7%		38.73	0.45
CNN_Y_NL proposed	94.7%		38.60	0.58
CNN_RGB_NL proposed	**100%**		39.18	**0.00**
CNN_YUV_NL proposed	98.2%		39.04	0.14
CNN_Y_NL proposed (patch65)	98.2%		39.04	0.14
CNN_RGB_NL proposed (patch65)	96.5%		38.88	0.30
CNN_YUV_NL proposed (patch65)	94.7%		38.73	0.45

Table 2.3 Performance on intra-coded images with CNN-assisted system.

Methods	Accuracy	$\overline{S}_{\text{intact}}$ (dB)	$\overline{S}_{\text{system}}$ (dB)	$\overline{S}_{\text{diff}}$ (dB)
CNN_Y_G pre-trained (Kang *et al.*, 2014)	33.3%	38.62	32.99	5.63
CNN_Y proposed	60.0%		30.19	8.43
CNN_RGB proposed	66.7%		36.49	2.13
CNN_YUV proposed	79.0%		36.79	1.83
CNN_Y_NL proposed	65.0%		34.02	4.60
CNN_RGB_NL proposed	73.7%		36.65	1.97
CNN_YUV_NL proposed	79.0%		37.11	1.51
CNN_Y_NL proposed (patch65)	77.0%		36.55	2.07
CNN_RGB_NL proposed (patch65)	79.0%		36.71	1.91
CNN_YUV_NL proposed (patch65)	**82.5%**		37.14	**1.48**

Table 2.4 Performance on inter-coded images with CNN-assisted system.

The proposed CNN_YUV configuration indicates that images are used directly in YUV format. For intra-encoded images, using local normalization provides better accuracy and less difference in quality when the CNN uses the Y, YUV or RGB channel. To better detect the local discontinuities in the boundaries of CTU blocks, we also modify the patch size from 64 to 65 and re-train our proposed CNN-based metric. Modifying the size of the patches from 64×64 pixels to 65×65 pixels shows improvements on the Y channel, but for YUV and RGB format images, it does not show better performance.

For inter images, the use of local normalization makes it possible to significantly improve performance both for Y and for RGB, but no improvement is achieved when YUV is used. However, when we apply the modification of the patch size, we detect the improvement in all 3 image formats, with a precision which goes from 65% to 77% for Y, from 79% to 82.5% for YUV and from 74% to 79% for RGB.

Finally, although the performances are similar for YUV and RGB in intra frame, the use of YUV performs better for intra and inter-coded images. We believe it has the advantage of being able to identify color distortions and avoid additional conversions which can cause image degradation. We also see that the modification of the patch size is able to detect discontinuities at the borders

of HEVC CTUs, which shows almost the same performances in intra frame. However, an improvement is shown in the inter frame simulation.

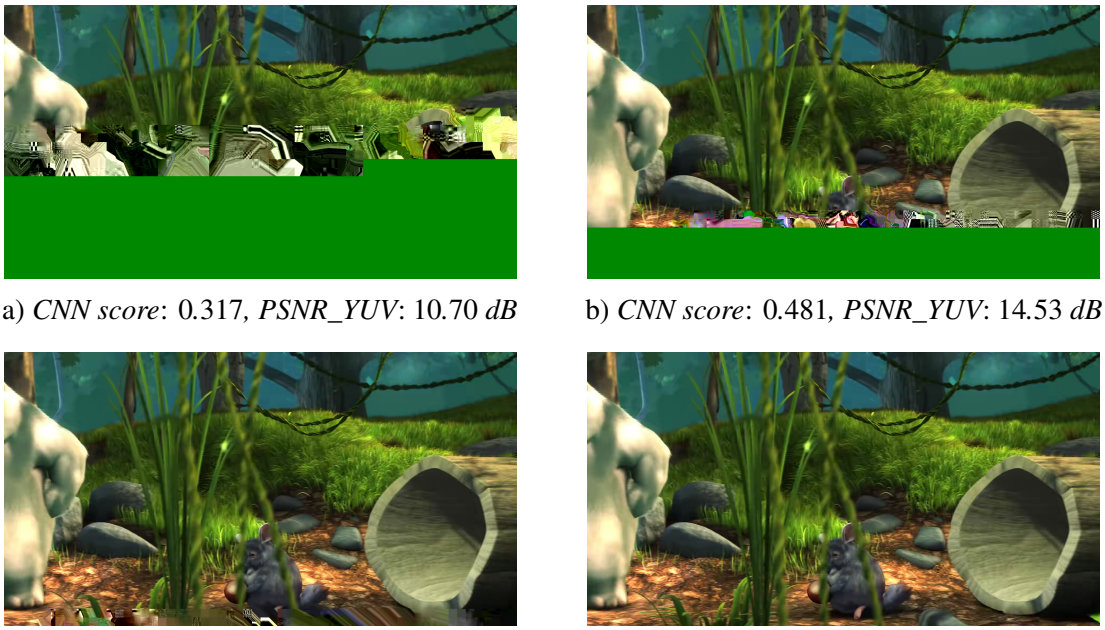
We note that the precision is much lower for inter-coded images than for intra-coded images. In fact, transmission errors in inter-coded images do not cause such significant losses of quality as in intra-coded images, which makes learning the model more difficult.

2.5.4 Concluding remarks

To conclude this chapter, we give some examples of bad decision situations encountered by our CNN-assisted video list decoding system. Figure 2.9 gives an example of a bad decision in intra frame simulations, i.e. in this case, the model did not choose the error-free image in the candidate list, but chose a corrupted version. Nevertheless, this chosen version is the one with the highest PSNR score among the three corrupted images. For the candidates selected by CNN, the errors are barely visible to the human eye at the bottom of the image as compared to the error-free version. Therefore, we can see that our model offers a good performance on intra-coded images.

Figure 2.10 shows an example of a bad decision in inter frame simulations. Although the classification accuracy on inter-coded images is much lower than that on intra-coded images in our simulations, we found that most of the incorrectly decided situations always chose the corrupted version with the highest PSNR score among the corrupted candidates. We know that transmission errors in inter frames do not cause as significant quality loss as in intra frames, which makes training the model more difficult. Therefore, we believe that although the performance of the proposed model on inter-coded images is insufficient, it is still a great improvement compared to the original model.

In the next chapter, we will introduce a new video list decoding system with the aim of increasing the performance of our solution by exploiting the properties of the Transformer architecture.



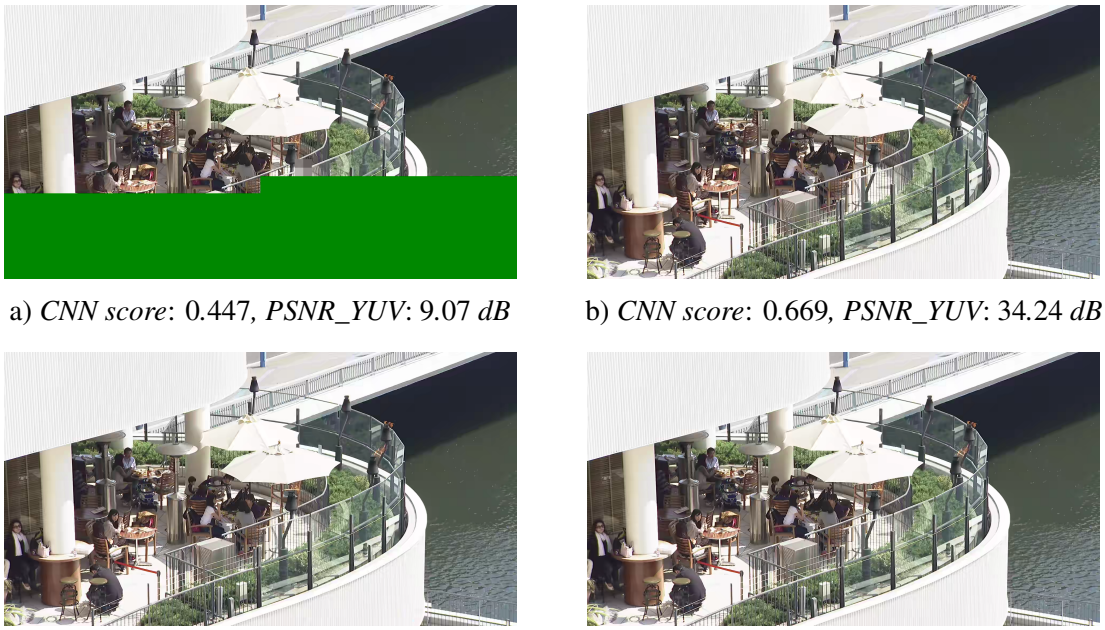
a) *CNN score*: 0.317, *PSNR_YUV*: 10.70 *dB*

b) *CNN score*: 0.481, *PSNR_YUV*: 14.53 *dB*

c) *CNN score*: 0.598, *PSNR_YUV*: 29.60 *dB*

d) *CNN score*: 0.594, *PSNR_YUV*: 37.65 *dB*

Figure 2.9 Example of bad decision (intra frame, CNN_YUV_NL proposed): choose the *best* decoded version. The system selects c) while the intact version is d)



a) *CNN score*: 0.447, *PSNR_YUV*: 9.07 *dB*

b) *CNN score*: 0.669, *PSNR_YUV*: 34.24 *dB*

c) *CNN score*: 0.673, *PSNR_YUV*: 34.79 *dB*

d) *CNN score*: 0.672, *PSNR_YUV*: 35.26 *dB*

Figure 2.10 Example of bad decision (inter frame, CNN_YUV_NL proposed): choose the *best* decoded version. The system selects c) while the intact version is d)

CHAPTER 3

PROPOSED TRANSFORMER-ASSISTED VIDEO LIST DECODING SYSTEM

3.1 Introduction

After the first simulations with the proposed CNN-assisted video list decoding system, we noticed that the CNN-based IQA method performed well on intra-coded images, to the point that it achieves 100% candidate selection accuracy on the proposed test set. However, for inter-frame encoded images, we noted that the precision was around 80%, which is significantly lower than for intra-coded images and still leaves much room for improvement. We found that simply changing local normalization, as done in Section 2.3, had its limitations. We aim to improve further our system by better distinguishing between a well-received uniform patch and an erroneous patch that appears uniform, and by improving the performance for inter-coded images by penalizing cases where a damaged patch obtains a higher score than an intact one, which is particularly important for inter-coded images with small distortions.

Therefore, in this chapter, we propose a new Transformer-assisted video list decoding system that includes a visual quality evaluation framework using a Transformer-based metric to identify the best candidate from the list. This new framework features a NR IQA metric based on a Vision Transformer to evaluate the quality of candidate videos, incorporating three new components: Neighborhood-based Patch Fidelity Aggregation (NPFA), Discriminant Color Texture Transformation (DCTT) and Ranking-Constrained Penalty Loss function (RCPL) to address the previous shortcomings. These improvements can also be applied to a CNN-based metric framework.

In this chapter, we firstly present the proposed transformer-assisted video list decoding framework. Then, we focus on the transformer-based image quality assessment process and explain all its new components: DCTT, NPFA, and RCPL.

3.2 Improved DL-assisted video list decoding system

We propose an enhanced and transformer-based version of the CNN-assisted video list decoding framework proposed in Section 2.1. This framework is the first Transformer-assisted video list decoding framework for error-prone video transmission systems. It utilizes a no-reference IQA metric based on Transformer architecture (Vaswani *et al.*, 2017) to identify the candidate with the highest visual quality from multiple options generated during the list decoding process in communications over unreliable networks. This aims to ensure that the final rendered image is optimal in terms of visual quality.

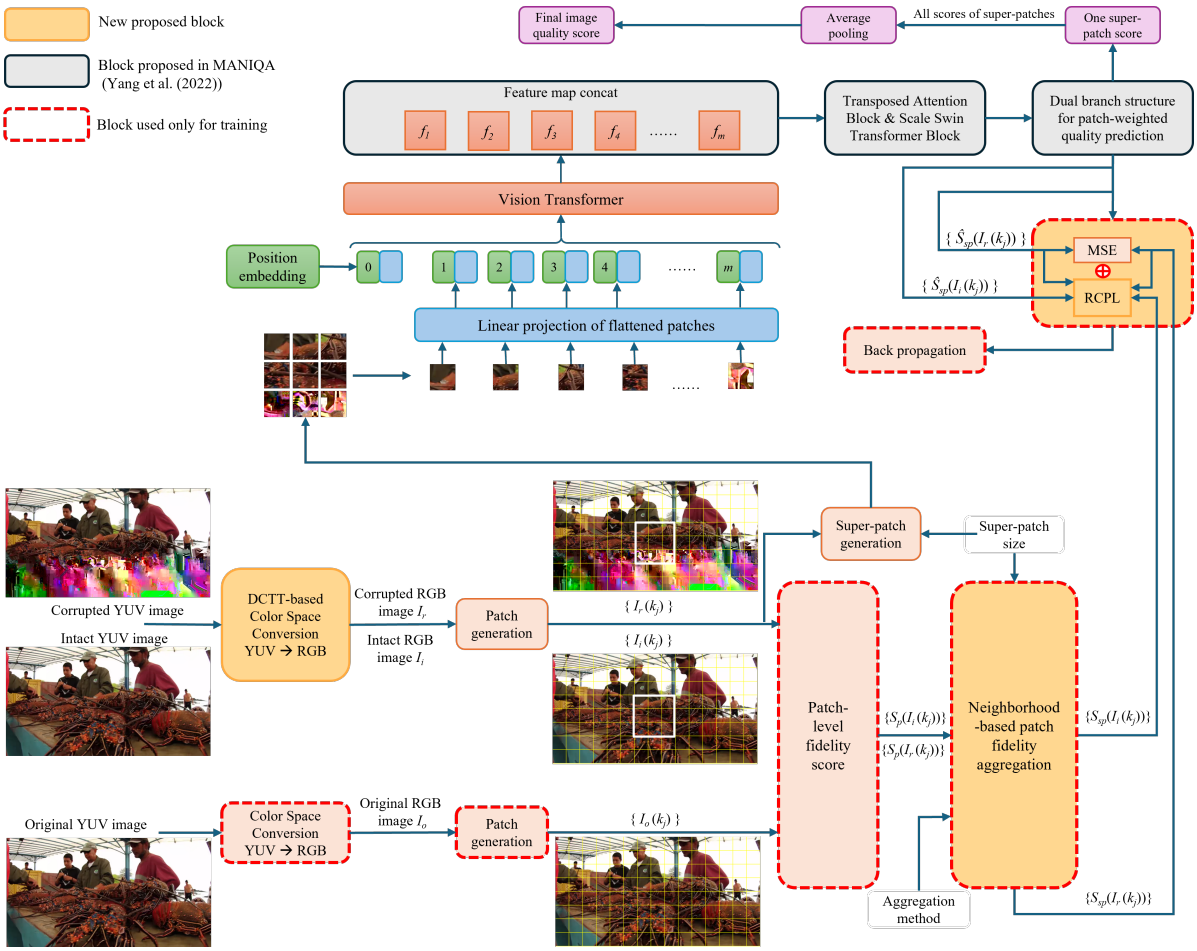


Figure 3.1 The proposed Transformer-assisted image quality estimation metric

As shown in Figure 3.1, the architecture of our metric consists of several blocks, including image pre-processing, network training and final image score calculation. For each original YUV video sequence, a list of candidate videos is generated and the first corrupted frames from each candidate sequence are extracted to enter our network. The candidate frame is represented by I_r, and each list of candidate videos includes an intact version I_i, which is received without error. We also prepare the corresponding original frame version I_o. Before feeding the images to the training network, the image pre-processing is applied. It consists of several steps. Firstly, we apply the image color space conversion to change the image format from YUV420 to RGB 4:4:4, with a DCTT (see Section 3.3.3) component to well distinguish the uniform regions caused by errors from the actual flat areas. Secondly, we generate the patches $I(k_j)$ for each version of the image and calculate the patch-level full-reference fidelity scores $S_p(I(k_j))$. Then, we combine $p \times p$ normal patches to generate *super-patches* and use the proposed NPFA (see Section 3.3.2) to generate the reference scores for super-patches to better consider the local discontinuities at HEVC CTU (codec blocks) horizontal and vertical boundaries between neighbourhood patches. The size of the super-patches and the aggregation method are variable parameters which could be changed in the future.

We propose our method based on MANIQA (Yang *et al.*, 2022), aiming to estimate the IQA for our specific case. We use super-patches as the input data for the neural network and propose the RCPL (see Section 3.3.4) to ensure the predicted score of the corrupted super-patch to be lower than the corresponding intact version. The following section will introduce in more detail our proposed Transformer-assisted IQA metric framework with three new proposed components.

3.3 Proposed Transformer-based visual quality evaluation method

To address the limitations of the CNN-based metric, we propose using a more comprehensive deep learning architecture, the Vision Transformer (Dosovitskiy *et al.*, 2020). Transformer models can learn to focus on the local patches and evaluate image quality. By employing an attention mechanism to swiftly compute the significance and interrelations among patches, these

approaches enhance the efficacy of handling extensive image datasets and evaluating image quality.

Our proposed Transformer-based image quality estimation metric is sensitive to local distortions in the image, which are non-uniformly distributed, based on a self-attention mechanism. Our proposed Transformer-assisted framework relies on a dependable process to select the video candidate with the best visual quality.

3.3.1 The original Transformer architecture

After a number of comparisons, we chose to use the Multi-dimension Attention Network for no-reference Image Quality Assessment (MANIQA, (Yang *et al.*, 2022)) as the base model, and improved it to better fit our system. As shown in the Figure 3.2, the original method consists of four components: a feature extractor using ViT (Dosovitskiy *et al.*, 2020), a transposed attention block, a scale swin transformer block, and a two-branch structure for patch-weighted quality prediction. This method first extracts and connects 4 layers of features from ViT, and then computes the weights of different channels by the proposed Transposed Attention Block (TAB). The authors apply the Self-Attention algorithm across channels rather than spatial dimensions to compute the mutual covariance across channels to generate the attention graphs in this module. To enhance the local interactions between image blocks, Scale Swin Transformer Block (SSTB) is applied. To stabilize the training process, the scale factor α is applied to adjust the residuals. These two modules (TAB and SSTB) apply the attention mechanism in the channel and spatial dimensions, respectively. Through this multidimensional approach, the modules synergize to increase the interaction between different regions of the global and local image. Finally, a two-branch structure consisting of weighted and scored branches for the importance of each patch is proposed and quality prediction is presented to obtain the final score of the image. The authors hypothesized that salient themes located mainly in the center of an image are compelling to the human visual system, but are not always of high quality. Due to the inconsistency between noteworthy and high-quality regions, the final weighted map balances the difference between the two through the Hadamard Product.

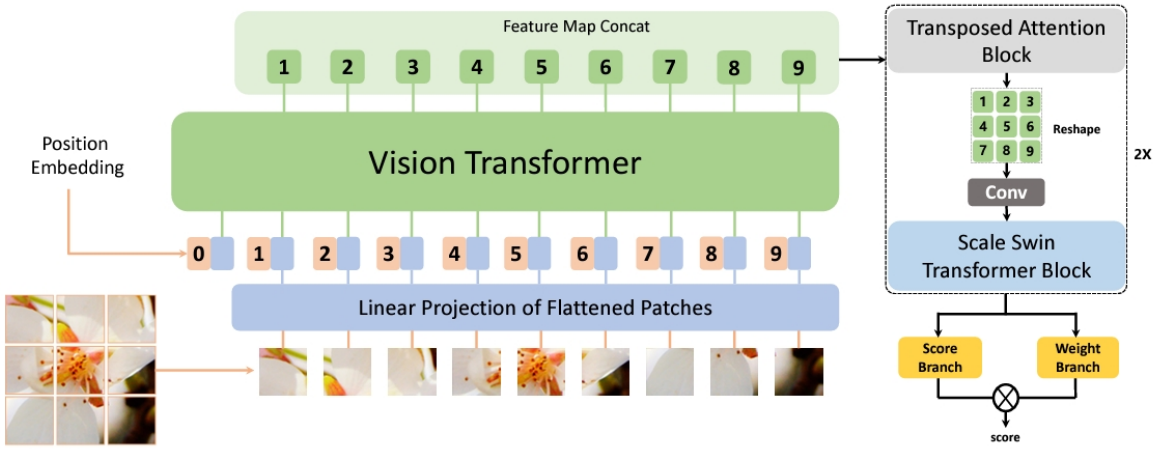


Figure 3.2 The architecture of MANIQA (Yang *et al.*, 2022)

As shown in Figure 3.2, the authors first proposed the overall pipeline of MANIQA structure. Given a distorted image $I \in R^{H \times W \times 3}$, where H and W denote the height and width of the image. Let f_ϕ present the vision transformer (ViT, (Dosovitskiy *et al.*, 2020)) with learnable parameters ϕ, and $F_i \in R^{b \times c_i \times H_i W_i}$ denotes the features from the i_{th} layer of ViT, where $i \in \{1, 2, ..., 12\}$, b denotes the batch size, and c_i, H_i, and W_i denote the channel size, width, and height of the i_{th} feature, respectively. The method uses 4 of the total 12 layers to extract features from different semantic degrees. Next, it concatenates $\hat{F}_i$, where $i \in \{7, 8, 9, 10\}$, and obtains the output denoted by $\hat{F} \in R^{b \times \sum_i c_i \times H_i W_i}$.

Next, the TAB is employed to boost the channel interaction among the extracted features. This block applies self-attention across channels rather than the spatial dimension to compute cross-covariance across channels to generate an attention map encoding the global context implicitly (see Figure 3.3). TAB first generates query (Q), key (K) and value (V) projections, which are achieved by 3 independent linear projections, to encode the pixel-wise cross-channel context. Then, it reshapes query and key projections such that their dot product interaction generates a transposed-attention map of size $R^{\tilde{C} \times \tilde{C}}$, where $\tilde{C}$ is numerically equal to c_i. As the dot product is conducted across channel dimensions, this TAB removes the layer normalization and multi-layer perceptron from the original Transformer (Vaswani *et al.*, 2017), which has advantages of rearranging channels' weight in terms of their importance to the perceptual

quality score and generate the attention map to encode the global context implicitly, which is complementary to downstream local interaction.

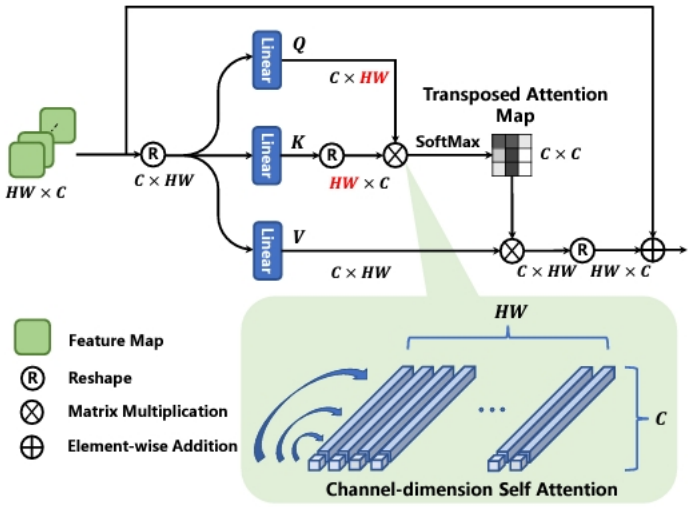


Figure 3.3 Transposed Attention Block of MANIQA (Yang *et al.*, 2022)

Then, the feature $\tilde{F}$ will be sent to the SSTB to strengthen the interaction of the local information. As shown in Figure 3.4, the SSTB consists of Swin Transformer Layers (STL) (Liu *et al.*, 2021) and a convolutional layer. The SSTB first encodes the feature through 2 layers of STL, then applies a convolutional layer before the residual connection. There are 2 advantages of this design: the convolutional layer with spatially invariant filters can enhance translational equivariance and the scale factor stabilizes the training through residual connection.

The final feature map is sent to the dual branch prediction module with the scoring branch and weighting branch. As shown in Figure 3.5, this module consists of a scoring and weighting branch which predicts each patch's score and weight, respectively. In this module, the final patch scores of the distorted image are generated by multiplying the scores of each patch with the weights, and then the final scores of the whole image are generated by summing up the final patch scores.

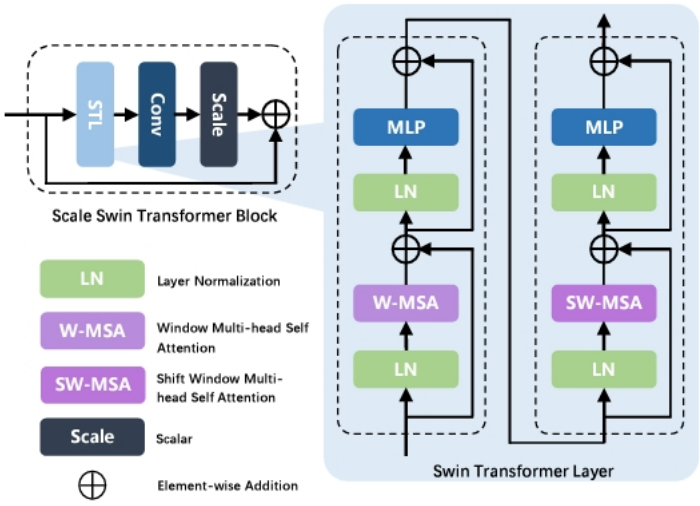


Figure 3.4 Scale Swin Transformer Block of MANIQA (Yang *et al.*, 2022)

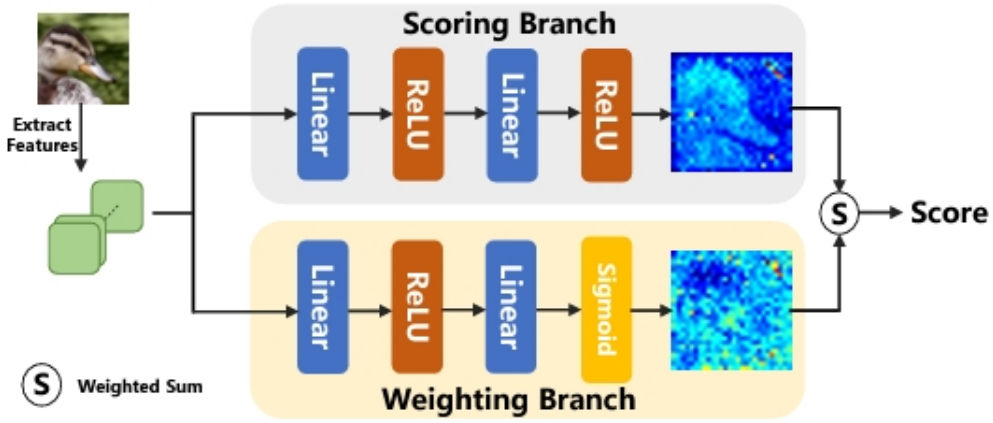


Figure 3.5 Dual branch structure for patch-weighted quality prediction of MANIQA (Yang *et al.*, 2022)

3.3.2 Application of Neighborhood-based patch fidelity aggregation

After reproducing the simulations with the CNN IQA metric in (Zhang *et al.*, 2023), we found that this simple system still has many shortcomings, such as the inability to detect discontinuities between neighbouring CTU blocks when the trained patch size is the same as the coding CTU size. Furthermore, replacing the CNN metric with a Transformer-based metric poses a challenge

due to the small size of the individual patches used in the CNN model, which is not compatible with the basic Transformer architecture (Yang *et al.*, 2022). Therefore, we introduce the use of super-patches instead of simple individual patches, where super-patches are the combination of $p \times p$ patches such that a complete image is divided into overlapping super-patches. The super-patches contain several neighbouring CTU blocks, enabling the model to more effectively analyze localized distortions resulting from error propagation in the neighbourhood of a given block. We demonstrate the necessity of super-patches for our research in the following.

3.3.2.1 Analysis of individual patches

We would like to have: $\hat{S}_p(I_r(k)) \leq \hat{S}_p(I_i(k))$ for all I_r, associated I_i and patch k, i.e. the estimated score of a patch from a tentatively repaired image should be smaller than (or equal to) that of the associated intact one. This is realized by adding a penalty term to the loss function during training, i.e. add a term $F2$ to the loss function as a penalty when $\hat{S}_p(I_r(k)) > \hat{S}_p(I_i(k))$.

Once trained, we can establish the performance per patch (based on $\hat{S}_p(I_r(k))$) before considering the performance for the whole image (based on $\hat{S}(I_r)$). For the analysis of patched, we calculate the difference d_r between $\hat{S}_p(I_r(k))$ and $\hat{S}_p(I_i(k))$, as shown in Eq.3.1.

$$d_r = \hat{S}_p(I_i(k)) - \hat{S}_p(I_r(k)) \tag{3.1}$$

We can also measure how often we have a higher score for a patch coming from a tentatively repaired image than from the intact one in the training set. As shown in Eq.3.2 and Eq.3.3, we calculate the average and barycenter for further analysis. K_p is the total number of patches, and $g(d_r)$ is the weight corresponding to each d_r.

$$\overline{d_r} = \frac{1}{K_p} \sum_{K_p}^{k=1} d_r = \frac{1}{K_p} \sum_{K_p}^{k=1} (\hat{S}_p(I_i(k)) - \hat{S}_p(I_r(k))) \tag{3.2}$$

$$d'_r = \frac{\sum_1^{K_p} d_r g(d_r)}{\sum_1^{K_p} g(d_r)} \tag{3.3}$$

We firstly use CNN predicted patch scores from intra-encoded RGB images, with patch size 64×64, and CNN model trained with improved local normalization algorithm. The histogram of the distribution of predicted scores is shown in Figure **??**, and the analysis results are presented in Table 3.1.

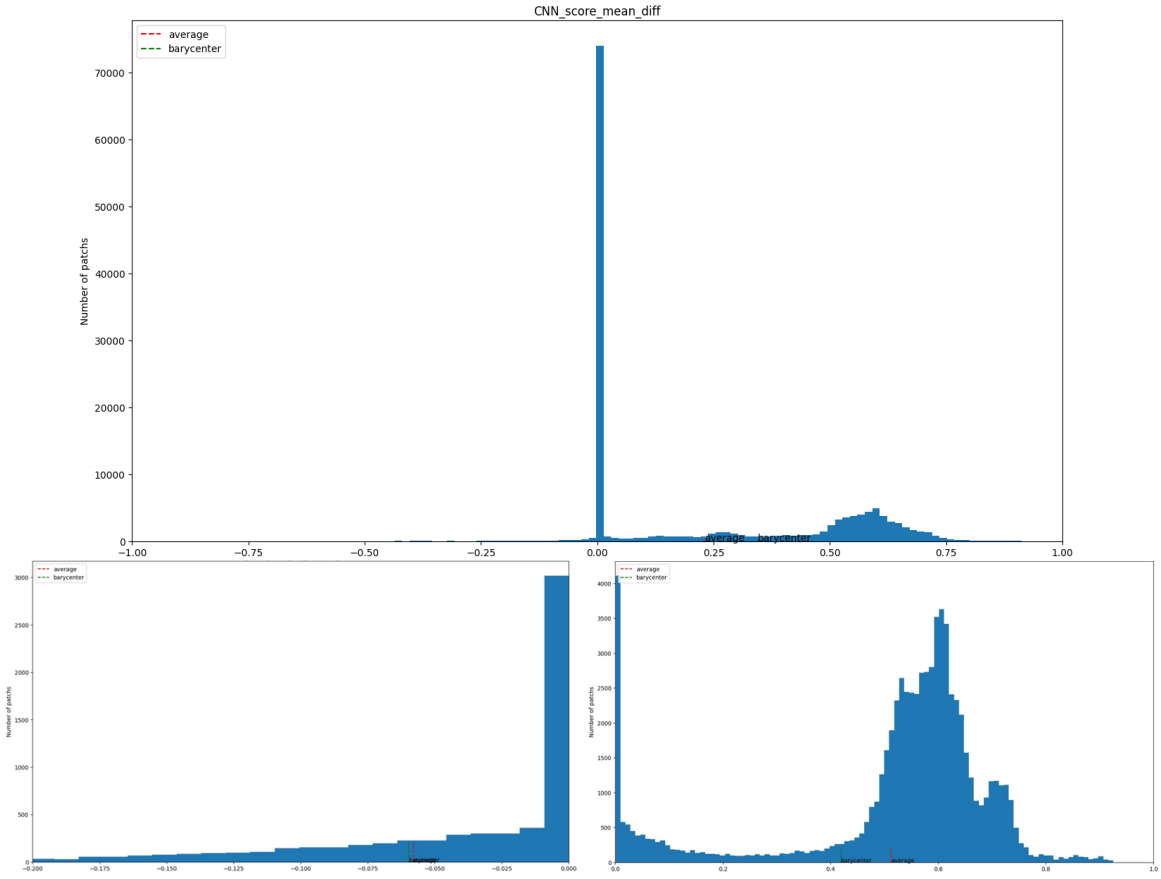


Figure 3.6 Analysis of patches in training set with CNN predicted scores (top: all scores, bottom left: negative scores, bottom right: positive scores)

We notice that the proposed CNN method leads to a huge number of patches with a difference equal or higher than 0 (zero or positive), and some patches with a difference smaller than 0

Image type	Negative	Positive	Zero	% Error	% Error_NZ	Barycenter	Average
CNN_RGB proposed (intra)	6600	72264	79536	4.17%	8.37%	0.3426	0.2315

Table 3.1 Analysis of patch results with CNN predicted scores.

(negative). The zero or positive differences indicate that our model makes the correct decision (good decision). And the negative differences show that our system made a bad decision and awarded a higher score to a lower quality patch. % Error represents the percentage of negative difference values from all difference values, while % Error_NZ represents the same percentage but excluding the zero values. The patch score predicted by the CNN approach leads to errors for about 4.17% patches.

Image type	% Error	% Error_NZ	Barycenter (negative)	Average (negative)	Barycenter (negative, PSNR/dB)	Average (negative, PSNR/dB)
CNN_RGB proposed (intra)	4.17%	8.37%	-0.0597	-0.0580	10.74	10.55

Table 3.2 Analysis of negative difference results.

With Table 3.2, we further analyze the bad decisions. We calculate the ground-truth difference $S_p(I_i(k)) - S_p(I_r(k))$ for the patches given higher scores than the correspond intact patches. This difference is based on PSNR scores, which are calculated between the repaired patches and the original patches. From the average and the barycenter of PSNR difference, we can see that there is a significant difference in PSNR scores between the distorted patches and the corresponding lossless versions, showing how much our model misjudges quality in some cases. This indicates that there is still room for further improvement in our model, and further proves the necessity of improving the loss function, as we will propose in this subsection.

3.3.2.2 Justification for the utilization of NPFA

Following the analysis of patches we did in the previous section, we also investigated combining the estimated scores of several patches in a manner different from simple averaging, i.e. using super-patch rather than simply patch-wise. Therefore, after completing the training using the CNN model, we performed an analysis of the patched as in Section 3.3.2.1, which allowed us to

also predict the performance that can be expected by using super-patches (based on $\hat{S}_{sp}(I_r(k))$) before applying the idea to the whole image (based on $\hat{S}(I_r)$).

We propose combining each $p \times p$ patch together as a "super-patch" to perform the analysis. Eq.3.4 presents an example of a "super-patch", x is an original patch. We also show two real examples of super-patches with $p = 2$ in Figure 3.7.

$$\begin{bmatrix} x_{a,b} & \dots & x_{a,b+p-1} \\ \dots & \dots & \dots \\ x_{a+p-1,b} & \dots & x_{a+p-1,b+p-1} \end{bmatrix} \tag{3.4}$$



Figure 3.7 Example of super-patches (left: from intact frame, right: from corrupted frame)

For the analysis of super-patches, we propose to use neighborhood-based patch fidelity aggregation for the analysis of super-patches, which constitutes the first originality of our transformer-based method. $\hat{S}_{sp}(I_r(k))$ presents the combination of the patch-level fidelity score $\hat{S}_p(I_r(k))$ associated with several neighbouring patches, which is calculated in Eq. (3.5). $f(k,i)$ returns patch number i in the neighbourhood of patch k and COMB is a function to aggregate the score of multiple neighbouring patches. It presents the proposed NPFA component, which can be selected among various aggregation functions, such as average, minimum or power pooling to obtain the aggregation score for each super-patch. This allows us to give greater importance to local errors instead of computing simple averages, which are considered to be too smooth. This is in accordance with the fact that quality assessment is not a global process but a local process based on a number of regions of interest that are more degraded.

$$\hat{S}_{sp}(I_r(k)) = \text{COMB}(\hat{S}_p(I_r(f(k,1))), \hat{S}_p(I_r(f(k,2))), ..., \hat{S}_p(I_r(f(k,n))))$$

where

$$\begin{aligned}
&\text{COMB}(\hat{S}_p(I_r(f(k,1))), \hat{S}_p(I_r(f(k,2))), ..., \hat{S}_p(I_r(f(k,n)))) \\
&= \begin{cases}
\min\limits_{i\in[1,n]} \hat{S}_p(I_r(f(k,i))), & \text{if minimum} \\
1 - \frac{1}{n}\sum_{i=1}^{n}[1 - \hat{S}_p(I_r(f(k,i)))]^2, & \text{if squared error} \\
\frac{1}{n}\sum_{i=1}^{n} \hat{S}_p(I_r(f(k,i))), & \text{if average}
\end{cases}
\end{aligned} \tag{3.5}$$

Ideally, we want all patches of the intact frame to receive from the IQA system a score higher than or equal to the corresponding patches in any candidate frame. This applies whether we are dealing with individual patches or with super-patches. Formally, we would like:

$$d_{r,sp} = \hat{S}_{sp}(I_i(k)) - \hat{S}_{sp}(I_r(k)) \geq 0 \tag{3.6}$$

As shown in Eq.3.7, we calculate the average and barycenter of d_r for further analysis. K is the total number of patches, and $g(d_r)$ is the weight corresponding to each d_r.

$$\begin{aligned}
\overline{d_r} &= \frac{1}{K_{sp}} \sum_{K_{sp}}^{k=1} d_r = \frac{1}{K_{sp}} \sum_{K_{sp}}^{k=1} (\hat{S}_{sp}(I_i(k)) - \hat{S}_{sp}(I_r(k))) \\
d'_r &= \frac{\sum_1^{K_{sp}} d_r g(d_r)}{\sum_1^{K_{sp}} g(d_r)}
\end{aligned} \tag{3.7}$$

Once the system is trained to estimate the quality of each patch, we can evaluate the performance per patch and per super-patch with different NPFA methods, before establishing the performance for the whole image. Evaluating the performance at the super-patch level using various aggregation functions will indicate which to select. The histogram of the distribution of predicted scores in average, minimum and squared error combination is shown in Figure 3.8, 3.9 and 3.10. Table 3.3 shows the results by reproducing the CNN predicted patch scores from

intra-coded RGB images in (Zhang *et al.*, 2023), with patch size 64×64, and CNN model trained with improved local normalization algorithm. We used $p = 2$ to combine the super-patches from the individual patches in the analysis.

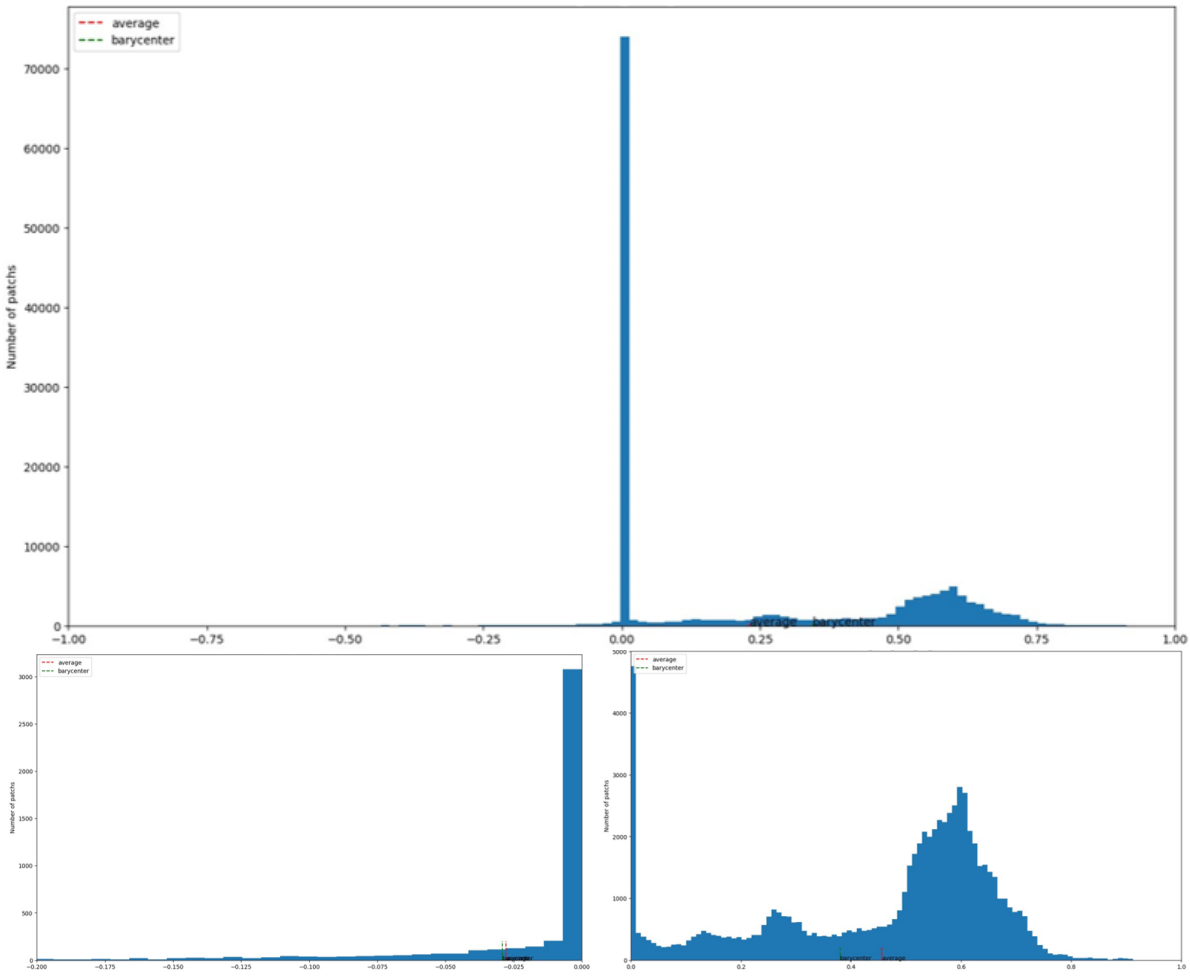


Figure 3.8 Analysis of super-patches in training set with CNN predicted scores and average combinations (top: all scores, bottom left: negative scores, bottom right: positive scores)

Combination type	Negative	Positive	Zero	% Error	% Error_NZ	Barycenter	Average
Individual patch	6600	72264	79536	4.17%	8.37%	0.3426	0.2315
Average	4623	72772	66155	3.22%	5.97%	0.3447	0.2302
Minimum	2645	71178	69727	1.84%	3.58%	0.3910	0.2467
Squared error	4541	72854	66155	3.16%	5.87%	0.3635	0.3146

Table 3.3 Analysis of super-patch results with CNN predicted scores.

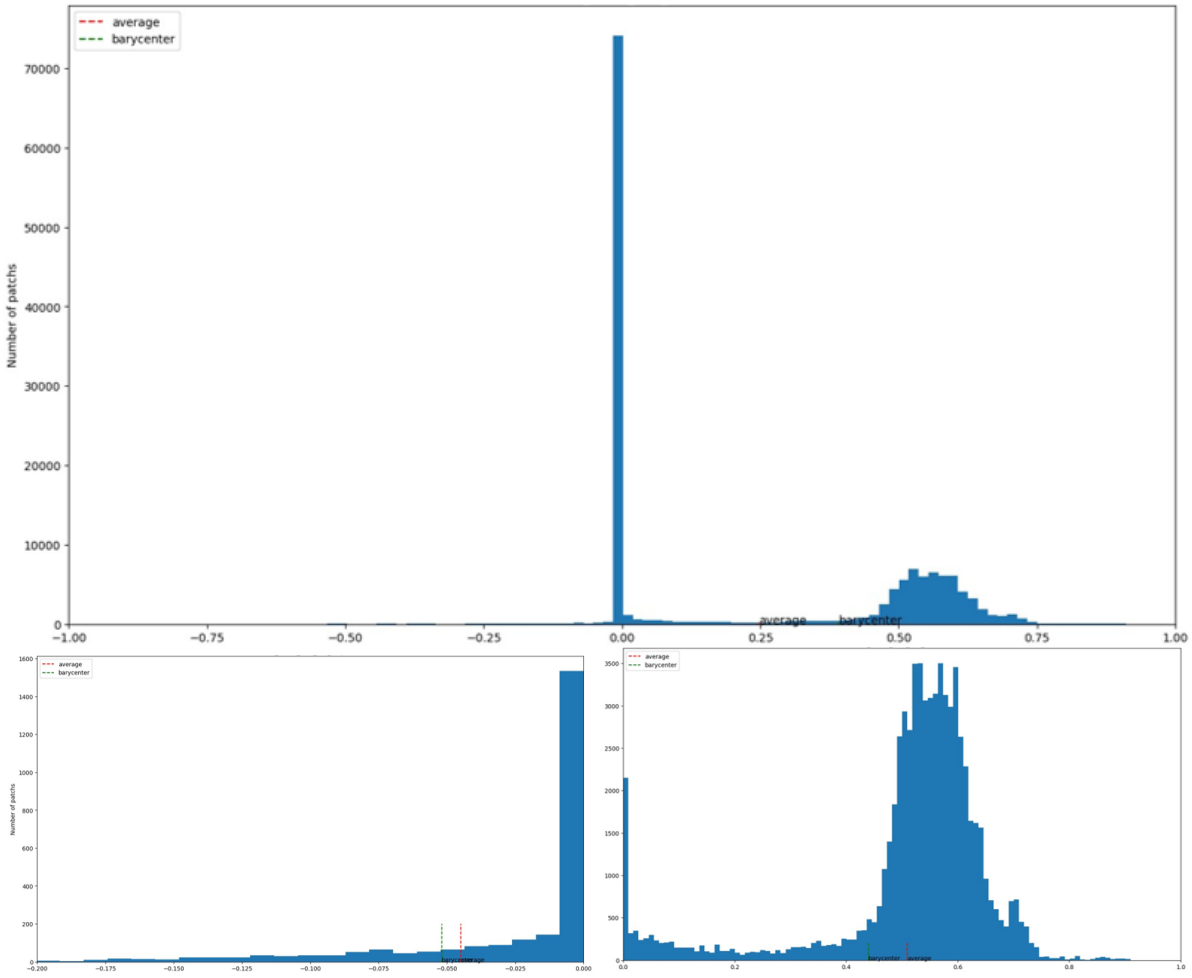


Figure 3.9 Analysis of super-patches in training set with CNN predicted scores and min combinations (top: all scores, bottom left: negative scores, bottom right: positive scores)

According to the table and the figures, we find that super-patches can help further enhance the system's performance compared to regular individual patches. We notice that using individual patch scores predicted by CNN leads to more errors than using an aggregation method to calculate the super-patch score. In other words, compared to the individual patch result, using the super-patch helps reduce the percentage of time when we make a mistake between the corrupted patch score and the intact patch score.

Each combination got a significant number of super-patches with a difference equal or higher than 0 (zero or positive), and some super-patches with a difference smaller than 0 (negative).

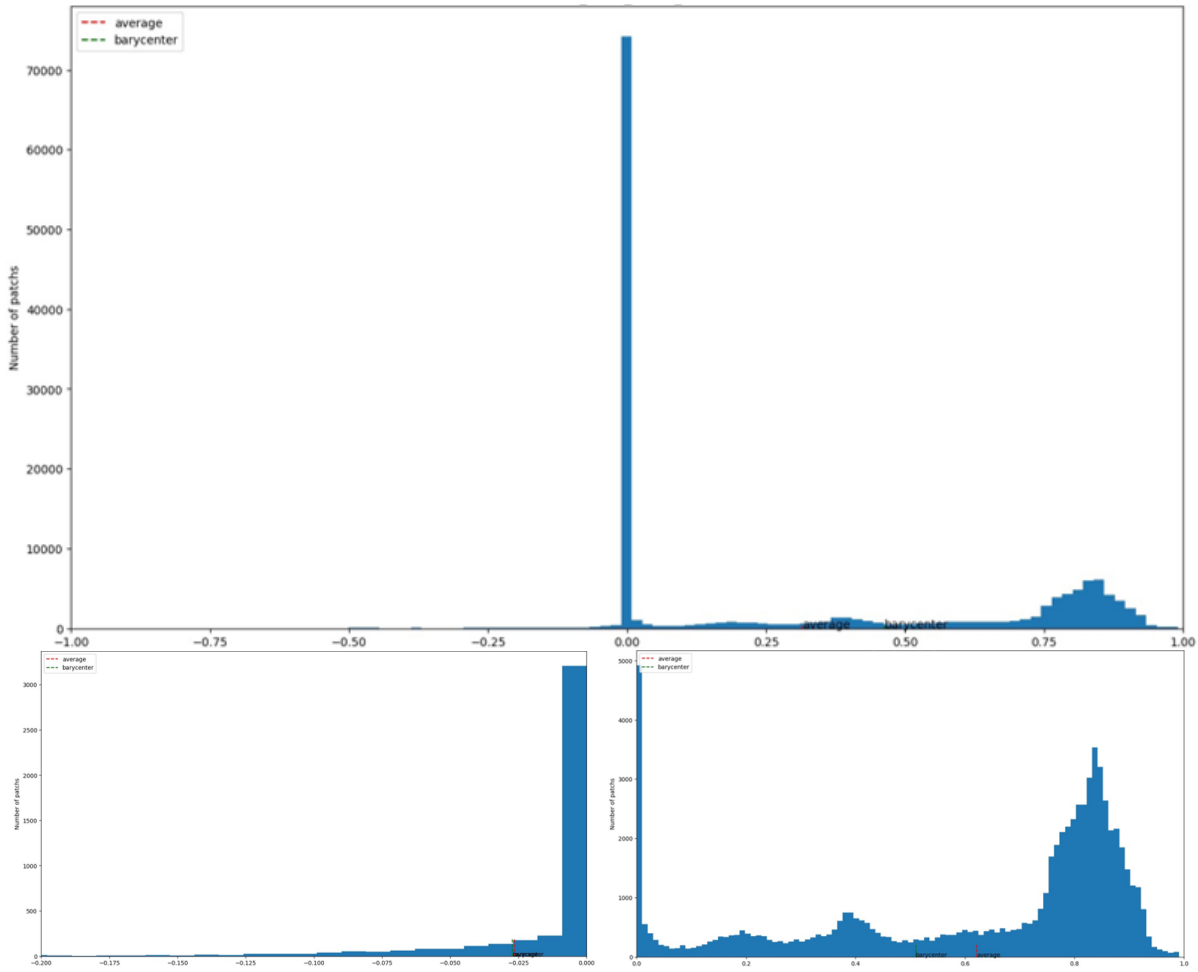


Figure 3.10 Analysis of super-patches in training set with CNN predicted scores and squared error combinations (top: all scores, bottom left: negative scores, bottom right: positive scores)

From these results, the 'minimum' aggregation leads to fewer errors. The distributions of negative differences for the three (3) combinations are very close to 0 (mostly between -0.2 and 0). But the barycenter of the negative points in Table 3.4 shows that the 'squared error' method has the barycenter closest to 0.

Furthermore, if we use the full image as input to the model, due to the high resolution of the image we are using (1920×1024), randomly cropping the image to 224×224 size from the training dataset, as proposed in the original MANIQA article (Yang *et al.*, 2022) is not sufficient to guarantee that the model learns the local distortions in different regions of the

Combination type	% Error	% Error_NZ	Barycenter (negative)	Average (negative)	Barycenter (negative PSNR/dB)	Average (negative PSNR/dB)
Individual patch	4.17%	8.37%	-0.0597	-0.0580	10.7392	10.5473
Average	3.22%	5.97%	-0.0292	-0.0279	15.7277	15.9148
Minimum	1.84%	3.58%	-0.0519	-0.0450	15.9450	15.8663
Squared error	3.16%	5.87%	-0.02716	-0.0265	15.7074	15.8609

Table 3.4 Analysis of super-patch results with ground-truth PSNR scores.

image completely. Therefore, we propose to use overlapping super-patches instead of randomly cropping the image. This not only ensures that the model fully learns the different types of local distortions in different images, but also increases the amount of data substantially and reduces the possibility of overfitting the model in training due to insufficient data.

We prefer to use super-patches with reliable metric scores to supervise the training of the Transformer system. We define our super-patches by combining the original patches together, where the original patch size is 32×32, and we choose $p = 7$ to create the super-patch in size 224×224 to adapt the input size of ViT (Dosovitskiy *et al.*, 2020) in the basic model. Considering the local discontinuities at horizontal and vertical boundaries between neighbourhood patches, we can also change the size and shape of the super-patch in the future.

For testing purposes, we use the average of the predicted super-patch scores for each image to obtain the image-level quality score $\hat{S}(I_r)$:

$$\hat{S}(I_r) = \frac{1}{K_{sp}} \sum_{k=1}^{K_{sp}} \hat{S}_{sp}(I_r(k)) \tag{3.8}$$

where $\hat{S}_{sp}(I(k))$ denotes the quality score predicted for super-patch $I_r(k)$ by our Transformer metric, and K_{sp} is the total number of super-patches in the image.

3.3.3 Discriminant Color Texture Transformation

In our proposed method presented in Section 2.3, and illustrated in Figure 2.3, we separated an image into several smaller patches and extracted features from each patch to assess their quality. We proposed to use local scores so that each patch has its own quality score. This was expected to help the neural network to learn local distortions more efficiently. We also proposed a solution to distinguish between a well-received uniform patch whose value is normalized to 0 and an erroneous patch that is uniform because it has been initialized to 0 by the decoder, which is to improve local normalization during training.

However, we found that simply changing local normalization still does not completely differentiate between the two. Instead, we propose a new discriminant color texture transformation (DCTT) based color space conversion to solve this problem during the conversion of the original YUV image to RGB. This transformation creates a totally different pattern for each channel of an RGB image instead of simply forcing the value to (0,0,0) when a spatial area is lost due to transmission errors. Since the decoder initializes each YUV pixel of an image to (0,0,0) prior to decoding, the value of a lost pixel remains zero when an error occurs. Therefore lost regions are easy to identify. For each pixel (i, j) where YUV values are all detected as 0 in patch k of image I_r, we apply Eq. (3.9) :

$$\begin{aligned} I_{r,R}(k,i,j) &= 255((-1)^{i+j}+1)/2 \\ I_{r,G}(k,i,j) &= 255((-1)^{i}+1)/2 \\ I_{r,B}(k,i,j) &= 255((-1)^{j}+1)/2 \end{aligned} \tag{3.9}$$

where $I_{r,R}(k,i,j)$ represents the value after the conversion of red channel for pixel (i, j) in patch k of image I_r, and, similarly, $I_{r,G}(k,i,j), I_{r,B}(k,i,j)$ represent the values of the green and blue channels, respectively. These patterns do not exist in natural RGB images; they are high-energy high-frequency patterns which do not exhibit the strong inter-channel correlation observed in natural images. We expect that patches having these patterns, being very different from other

patches, will be assigned an extremely low quality score through the learning process. Figure 3.11 shows an example of DCTT applied to two uniform patches. The candidate image on the top has its bottom region initialized to zero in all channels Y,U,V since it was received erroneously. A patch within this region normally appears green. After applying the proposed DCTT-based color space conversion from YUV to RGB, the transformed patch exhibits a texture in all color planes that is visually distinct from the transformed uniform patch shown at the bottom of Figure 3.11. With these new patch patterns, the neural network will be able to differentiate between uniform patches from erroneous regions, which receive low scores, and those from intact regions, which receive high scores. After normalization, a uniform patch and an erroneous patch become dissimilar, allowing the network to train with different reference scores.

3.3.4 Ranking-constrained penalty loss function

In the proposed method of Section 2.3, we utilized the mean absolute error (L1) function to calculate the loss during training, while the MANIQA model (Yang *et al.*, 2022) employed the MSE loss. These single loss functions performed well on intra-coded images, but there remains room for improvement when applied to inter-coded images. Therefore, we consider improving the loss function for our improved system, to further penalize cases where a damaged patch obtains a higher score than an intact one, which is particularly important for inter-coded images where distortions resulting from a transmission error are not as severe as for intra-coded images.

As mentioned in Section 3.3.2, we would like to make sure that the estimated score of a super-patch from a tentatively repaired image be smaller than (or equal to) that of the associated intact one. Once trained, we can establish the performance per super-patch with NPFA method 'minimum' as shown in Table 3.3. We noticed that the proper selection of the aggregation function is important but that we should also push the system to avoid such negative differences in the first place by adding a penalty term in the loss function, i.e. add a term F_2 to the loss function as a penalty when $\hat{S}_{sp}(I_r(k)) > \hat{S}_{sp}(I_i(k))$.

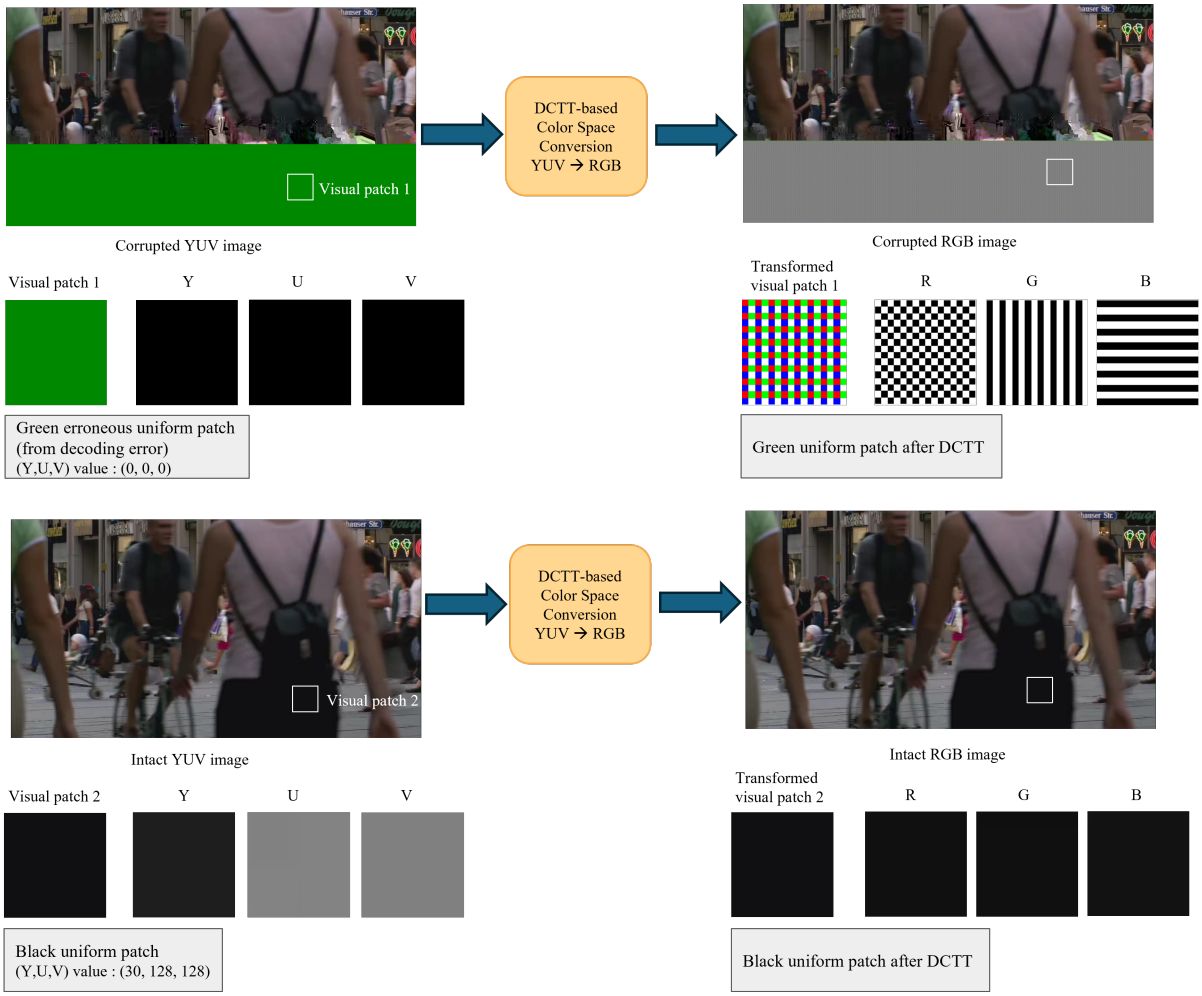


Figure 3.11 Example of the application of the proposed Discriminant Color Texture Transformation (DCTT), as part of the YUV to RGB conversion, to an erroneous green patch (top) and to an intact uniform patch (bottom)

As shown in the Eq.(3.10), F_1 is the original loss function (mean square error) used in the MANIQA (Yang *et al.*, 2022) system. F_2 is the new loss term we add to the loss function, F, during training. By adding F_2, we impose a penalty when the predicted score of the corrupted super-patch exceeds that of the corresponding intact super-patch. I represents an image that was coded, transmitted, and decoded. The image with transmission errors may not be decodable. I_i is defined as the intact image associated with image I, i.e. just coded and decoded without error. I_o is the original image associated with image I, i.e. without any compression. I_r is the repair tentative version r of image I, where $r \in [1, R]$. $\hat{S}_{sp}(I_r(k))$ presents the predicted score

of the super-patch k by our proposed Transformer-based model, $S_{sp}(I_r(k), I_o(k))$ is the ground truth score of each super-patch k and $\hat{S}_{sp}(I_i(k))$ is the predicted score by the Transformer-based model of the corresponding intact super-patch. $S_{sp}(I_r(k), I_i(k))$ is the actual score between the corrupted super-patches and the super-patches transmitted without errors. K_{sp} is the number of super-patches in each candidate image.

$$
\begin{aligned}
F_1 &= \frac{1}{K_{sp}} \sum_{k=1}^{K_{sp}} ||\hat{S}_{sp}(I_r(k)) - S_{sp}(I_r(k), I_o(k))||^2 \\
F_2 &= \frac{1}{K_{sp}} \sum_{k=1}^{K_{sp}} \max(0, \hat{S}_{sp}(I_r(k)) - \hat{S}_{sp}(I_i(k)) + \delta) \\
F &= \begin{cases} \min F_1, \text{ if } S_{sp}(I_r(k), I_i(k)) = 1 \\ \min(\alpha F_1 + (1-\alpha) F_2), \quad \text{if not} \end{cases}
\end{aligned}
\tag{3.10}
$$

By adding F_2, we try to ensure that $\hat{S}_{sp}(I_r(k))$ is smaller than $\hat{S}_{sp}(I_i(k))$. At the start of training, we will surely have cases where $\hat{S}_{sp}(I_r(k)) - \hat{S}_{sp}(I_i(k)) > 0$ so the max value between 0 and the difference will take the value of $\hat{S}_{sp}(I_r(k)) - \hat{S}_{sp}(I_i(k))$. From that the predicted corrupted super-patch has a score lower than the score of the predicted intact super-patch, then we have 0. The purpose of adding the variable δ is to guarantee that $\hat{S}_{sp}(I_r(k)) < \hat{S}_{sp}(I_i(k))$ to avoid equality between corrupted and intact super-patches, so one should apply the constraint function only if $\hat{S}_{sp}(I_r(k)) - \hat{S}_{sp}(I_i(k)) \neq 0$. If the super-patches at the start between the intact and corrupted images are the same, we do not want to penalize the network. So we add a condition of F_2 : when we test $S_{sp}(I_r(k), I_i(k)) = 1$, that is to say $I_r(k)$ and $I_i(k)$ have the same reference scores, we only consider F_1. Otherwise, we consider the two loss functions together, where we use $\alpha \in [0, 1]$ to represent the weight coefficient of the F_1 component of the loss function. We hope to train the system to find the parameters allowing the new loss functions to be minimized with suitable weight coefficient α.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

In this chapter, we present the experimental results of the proposed DL-assisted video list decoding system, utilizing either CNN-based or Transformer-based IQA components. The results include performance evaluations, ablation studies, and parameter sensitivity analyses.

4.2 Experimental results for the CNN-assisted IQA system component

In this section, we evaluate the performance and effectiveness of the proposed CNN-assisted video list decoding framework under various conditions. We begin by describing the training and testing methodologies, including the datasets, evaluation metrics, and baseline methods used for comparison. Following this, we provide a detailed analysis of the results, highlighting the strengths and weaknesses of our approach. We also perform an ablation study and parameter sensitivity analysis to show the positive impact of our newly proposed system components. The results are interpreted in the context of the research objectives, and we conclude with a summary of the insights gained from the experiments.

4.2.1 Training and testing methodologies

Our experiments are conducted using an NVIDIA Quadro RTX 5000 GPU with PyTorch 2.3.0 and CUDA 12.1 for training and testing. We trained and tested our model on our proposed patch based datasets. Based on the database created in Section 2.4, we use the same original sequences from xip; Wang *et al.* (2016); Pinson (2013) with the proposed DCTT color space conversion. The collected videos are in YUV format with a resolution of 1920×1024. We extract the first 10 frames from each video to encode them with the HEVC standard (Sze & al, 2014) using the HM version 16.20 reference encoder with default low-delay P configuration. Among the different possible QP values, we chose 37 which corresponds to a frequently used value in low bitrate

real-time video applications, and that will also create more severe degradations compared to low QP. We assume that each encoded frame is contained in a single video packet. The first frame of the encoded video is an intra (I) frame, and the next 9 frames are inter (P) frames. To simulate the combination of a transmission error followed by list decoding where bits are inverted at different locations. Inverted bit positions are chosen based on the equation $p = \beta \times M$, where $\beta = \{0.1, 0.2, ..., 0.9, 0.99\}$ and M is the size of each packet. We add the transmission error pattern for intra frame and inter frame separately, which means when the error pattern is added to inter frame, it directly hits the corrupted inter frame, and does not correspond to an error propagation occurring in the previous intra or inter frame. Therefore, for each sequence and each frame type, we have 11 candidates each time for the list-decoding system, including an error-free candidate.

Our database has about 475 200 non-overlapping patches for each frame type in our simulation, with a patch size of 64×64 pixels. Each patch is associated with a PSNR (Sara *et al.*, 2019) score between the reconstructed version and the original version in the interval [0, 50] dB, which is normalized to the interval [0, 1] during training. Following the standard training strategy outlined in existing IQA algorithms (Kang *et al.*, 2014), we randomly split each dataset into a 60:40 ratio, with 60% allocated for training, 20% for validation and the remaining 20% for testing. During training, we set the learning rate l to 0.001 and the batch size B to 128. We utilized the ADAM optimizer. The training loss used is the proposed RCPL, where F_1 is the L1 loss, and $\alpha = 0.5$. The final score is generated by averaging the scores predicted for all patches predicted in each image.

4.2.2 Application of DCTT and RCPL to CNN-based method

We enhance the framework proposed in Section 2.2 by adding the following components proposed in Sections 3.3.3 and 3.3.4:

(a) Discriminant Color Texture Transformation is proposed to distinguish between a well-received uniform patch and an error patch initialized to be uniform by the decoder.

(b) Ranking-constrained penalty loss function is proposed to further penalize cases where a damaged patch obtains a higher score than an intact one, which is particularly important for inter images with subtle errors.

The proposed advanced framework is designed to evaluate image quality with local distortions. In other words, it is sensitive to detect the non-uniform distortions caused by transmission errors. By adding two new components, our CNN-assisted framework performs significantly better both in intra-coded and inter-coded images.

4.2.3 Simulation results and analysis

As discussed in Section 2.5, we train and test the original CNN model and our improved version on the new dedicated database. We use the same metrics (Eq.2.4) to evaluate the performance of the models with different configurations. As shown in Section 2.5.2, $\overline{S}_{\text{intact}}$ indicates the average PSNR, relative to the original versions, of all intact frames, which are compressed but received without transmission errors. Here S is the PSNR score calculated on the RGB color space. $\overline{S}_{\text{system}}$ represents the average PSNR, relative to the original versions, of all images selected by a given method. $\overline{S}_{\text{diff}}$ is, for a method, the absolute difference between the average quality of the selected images and that of the intact images.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
PIQE (Venkatanath *et al.* (2015))	30.4%	34.76	19.84	14.92
NIQE (Mittal *et al.* (2013))	5.4%		16.30	18.46
BRISQUE (Mittal *et al.* (2012))	19.6%		18.29	16.47
CNN_NR_IQA (Kang *et al.* (2014))	46.4%		24.78	9.98
CNN_NR_IQA_RE	96.4%		34.44	0.32
CNN_NR_IQA_NL	**100%**		34.76	**0.00**
CNN_RGB_DCTT_RCPL	**100%**		34.76	**0.00**

Table 4.1 Performance on intra-coded images with CNN models applied newly proposed components DCTT and RCPL.

Tables 4.1 and 4.2 present the experimental results comparing our approach with state-of-the-art methods, using images encoded in *intra* and *inter* modes, respectively. The best results are

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
PIQE (Venkatanath *et al.* (2015))	28.6%	34.22	24.38	9.84
NIQE (Mittal *et al.* (2013))	17.9%		26.48	7.74
BRISQUE (Mittal *et al.* (2012))	28.6%		22.75	11.47
CNN_NR_IQA (Kang *et al.* (2014))	33.9%		28.78	5.44
CNN_NR_IQA_RE	67.9%		32.12	2.10
CNN_NR_IQA_NL	75.0%		32.20	2.02
CNN_RGB_DCTT_RCPL	**92.9%**		33.60	**0.62**

Table 4.2 Performance on inter-coded images with CNN models applied newly proposed components DCTT and RCPL.

highlighted in bold. PIQE (Venkatanath *et al.*, 2015), NIQE (Mittal *et al.*, 2013) and BRISQUE (Mittal *et al.*, 2012) use packaged functions directly within Matlab for inference simulations. CNN_NR_IQA denotes our use of the pre-trained model from the study (Kang *et al.*, 2014), which assigns the same score to each patch of the Y component of the image (global score) and is tested with our database containing non-uniform distortions. CNN_NR_IQA_RE refers to our use of the retrained model from the study (Kang *et al.*, 2014). This model assigns a patch-level fidelity score (local score) to each patch of the R, G, and B components of the image. The CNN method with the suffix _NL indicates a configuration that incorporates the enhanced local normalization method described in Section 2.3. The methods suffixed with _DCTT and _RCPL denote the application of our new components proposed in this paper to CNN-assisted systems.

For intra-encoded images, using local normalization achieves better accuracy and less difference in quality when the CNN uses the RGB channel. When we apply the proposed DCTT color space conversion and the improved RCPL components, the performances on intra frames always return perfect results. For inter images, the use of local normalization makes it possible to significantly improve performance for RGB with a precision which goes from 67.9% to 75%. We can see great improvements after applying the newly proposed components to the simulation with inter frames, with a precision which goes from 75% to 92.9%.

We can see that with the addition of the proposed two components to the dataset we created earlier, the simulation results are somewhat improved for both intra frames and inter frames. This

proves that our proposed DCTT color space conversion also performs well for distinguishing between two cases: a well-received uniform patch whose value is normalized to 0, and an erroneous patch that is uniform because it has been initialized to 0 by the decoder. The great improvements to the performance on inter frames also shows that our proposed new loss function is effective for improving the performance of CNN. It allows our model to better learn the quality degradations caused by local distortion on transmission error images compared to lossless images, while penalizing the network when the score of a patch from non intact frame has higher score than that of the intact one, allowing it to better select high-quality candidate images. This provides us with a significant reduction in the difficulty of selecting the "best candidate" in the candidate list of inter frames.

4.2.4 Ablation study

We provide ablation experiments to illustrate the effect of different parameters and each component of our proposed method by comparing the results on our proposed dataset.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_RETRAINED (Kang *et al.* (2014))	96.4%	34.76	34.44	0.32
CNN_DCTT	**100%**		34.76	**0.00**
CNN_RCPL	98.2%		34.39	0.37
CNN_DCTT_RCPL	**100%**		34.76	**0.00**

Table 4.3 Performance on intra-coded images with CNN model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_RETRAINED (Kang *et al.* (2014))	67.9%	34.22	32.12	2.02
CNN_DCTT	73.2%		33.34	0.88
CNN_RCPL	41.1%		21.02	13.20
CNN_DCTT_RCPL	**92.9%**		33.60	**0.62**

Table 4.4 Performance on inter-coded images with CNN model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Proposed new components: Tables 4.3 and 4.4 compare the CNN-assisted method from Section 2.3 to different simulation configurations incorporating the proposed DCTT and RCPL components into the CNN-based system. The first row of each table, labeled CNN_RETRAINED, indicates that we retrained the original CNN model from (Kang *et al.*, 2014) using our created database, where images initially used in YUV format are converted to RGB format for training and inference, without applying the proposed DCTT color space conversion or using RCPL during training. A method with the suffix _DCTT indicates a configuration that applies our DCTT color space conversion to our database (Eq. (3.9)). The suffix _RCPL indicates a configuration that applies our improved RCPL component as part of the loss function F with α initially set to 0.5 in Eq. (3.10).

For intra-coded images, applying the proposed DCTT color space conversion and RCPL consistently yields perfect results. For inter-coded images, the use of the proposed DCTT color space conversion significantly improves performance, with precision increasing from 67.9% to 73.2%. However, simply applying the RCPL component to the system does not result in any performance improvement; we observe the opposite. Notably, applying both new components to the simulation with inter-coded frames shows substantial improvements, with precision increasing from 67.9% to 92.9%. We notice that adding the proposed DCTT color space conversion to our datasets improves the simulation results for both intra-coded and inter-coded frames. This improvement aligns with the enhancement achieved by the local normalization algorithm proposed in Section 2.3.

Coefficients of RCPL: The coefficients α of F_1 and F_2 in Eq. (3.10) can be modified. We vary α to different values ($0, 0.2, 0.5, 0.8$ and 1) to observe the variations in performance with inter-coded frames on the CNN-assisted model, as shown in Table 4.5.

We can clearly see that a value near $\alpha = 0.2$ brings the best performance. According to Table 4.5, when $\alpha = 0.2$, the CNN-assisted system shows the highest accuracy of selecting the best candidate and the lowest difference between the average quality of the selected image and the average quality of the lossless image. This reflects the necessity of properly adjusting the weight

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_DCTT_RCPL ($\alpha = 0$)	91.1%		32.77	1.45
CNN_DCTT_RCPL ($\alpha = 0.2$)	**98.2%**		**33.69**	**0.53**
CNN_DCTT_RCPL ($\alpha = 0.5$)	92.9%	34.22	33.60	0.62
CNN_DCTT_RCPL ($\alpha = 0.8$)	76.8%		33.41	0.81
CNN_DCTT_RCPL ($\alpha = 1.0$)	73.2%		33.34	0.88

Table 4.5 Performance on inter-coded images with different coefficients in loss function on CNN model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

of F_2 compared to F_1, to optimize the model's performance brought by the new proposed loss function.

4.2.4.1 Parameters sensitivity analysis

We also conducted parameter sensitivity experiments to evaluate how sensitive the output of our proposed model is to changes in database variables.

Patch size: We explored changing the patch size of our database to observe its impact on our model. Table 4.6 and 4.7 present the performance with variable patch sizes for intra-coded and inter-coded frames, respectively.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_DCTT_RCPL (patch_size=32)	100%		34.76	0.00
CNN_DCTT_RCPL (patch_size=64)	100%	34.76	34.76	0.00
CNN_DCTT_RCPL (patch_size=128)	100%		34.76	0.00

Table 4.6 Performance on intra-coded images with CNN model by changing the patch size. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

For intra-coded images, changing the patch size does not affect the final classification results, and our model consistently maintains excellent accuracy. For inter-frame coded images, varying the patch size leads to small fluctuations in the final classification results, but these changes are

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_DCTT_RCPL (patch_size=32)	94.6%	34.22	33.60	0.62
CNN_DCTT_RCPL (patch_size=64)	92.9%		33.60	0.62
CNN_DCTT_RCPL (patch_size=128)	85.7%		33.33	0.89

Table 4.7 Performance on inter-coded images with CNN model by changing the patch size. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

not dramatic overall. The stability in accuracy is enabled by our proposed DCTT and RCPL components.

4.3 Experimental results for the Transformer-assisted IQA system component

This section details the experimental results that demonstrate the performance of our proposed Transformer-assisted video list decoding system component. We start by outlining the training and testing methodologies, including the datasets, evaluation metrics, and baseline methods used for comparison. We then present the results of our experiments, providing a comprehensive analysis of the data. The outcomes are discussed in terms of accuracy, efficiency, and robustness, with visual aids such as tables to illustrate key points. We also compare our results with existing methods to highlight the improvements and innovations introduced by our approach. Finally, we summarize the findings and discuss their implications for future research and practical applications.

4.3.1 Training and testing methodologies

Our experiments are conducted using two NVIDIA RTX A6000 GPUs with PyTorch 2.3.0 and CUDA 12.1 for training and testing. We trained and tested our model on our proposed super-patch-based datasets. Based on the backbone model MANIQA (Yang *et al.*, 2022), we also use ViT-B/8 (Dosovitskiy *et al.*, 2020) as our pre-trained model, which is trained on ImageNet-21k (Ridnik *et al.*, 2021) and fine-tuned on ImageNet1k (Russakovsky *et al.*, 2015) with the patch size P set to 8.

Our database has about 830 000 overlapping super-patches for each frame type in our simulation, with a super-patch size of 224×224 pixels. Each super-patch is associated with a PSNR (Sara *et al.*, 2019) score between the reconstructed version and the original version in the interval [0, 50] dB, which is normalized to the interval [0, 1] during training. PSNR and SSIM scores are both adequate choices for patch-level reference scores. For simplicity, we choose PSNR as the full-reference patch-level fidelity score, as it is a low-complexity widely recognized distortion metric. The PSNR scores are aggregated using the proposed NPFA with *minimum* aggregation function to calculate the super-patches scores from the individual patches of size 32×32 (see Eq. (3.5)). Following the training strategy outlined in existing IQA algorithms (Yang *et al.*, 2022), we randomly split each dataset into 80:20 ratio, with 80% allocated for training and the remaining 20% for validation. During training, we set the learning rate l to 10^{-5} and the batch size B to 8. We utilized the ADAM optimizer with weight decay 10^{-5}. The training loss used is the proposed RCPL, where F_1 is the MSE loss, and $\alpha = 0.5$. The final score is generated by averaging the scores predicted for all super-patches predicted in each image.

Note that we train our system on QP=37 because it represents a higher quantization parameter, which introduces more compression artifacts and a higher level of distortion. This challenging scenario allows the model to learn how to handle significant visual degradation and error propagation, making it robust and effective in improving visual quality under difficult conditions. By testing on both QP=37 and QP=22, we can evaluate the model's performance across a range of compression levels, ensuring it is versatile and performs well not only in high-distortion scenarios (QP=37) but also in lower-distortion, higher-quality scenarios (QP=22). This comprehensive testing demonstrates the model's ability to generalize and maintain high accuracy across different levels of video compression.

4.3.2 Simulation results and analysis

As discussed in Section 2.5, we train and test the original Transformer model and our improved version on our new dedicated database. We use the same metrics (Eq.2.4) to evaluate the performance of the models with different configurations. As shown below, $\overline{S}_{\text{intact}}$ indicates the

average PSNR, relative to the original versions, of all intact frames, which are compressed but received without transmission errors. Here S is the PSNR score calculated on the RGB color space. $\overline{S}_{\text{system}}$ represents the average PSNR, relative to the original versions, of all images selected by a given method. $\overline{S}_{\text{diff}}$ is, for a method, the absolute difference between the average quality of the selected images and that of the intact images.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
PIQE (Venkatanath *et al.* (2015))	30.4%		19.84	14.92
NIQE (Mittal *et al.* (2013))	5.4%		16.30	18.46
BRISQUE (Mittal *et al.* (2012))	19.6%		18.29	16.47
CNN_NR_IQA (Kang *et al.* (2014))	46.4%		24.78	9.98
CNN_NR_IQA_RE	96.4%	34.76	34.44	0.32
CNN_NR_IQA_NL	**100%**		34.76	**0.00**
MANIQA (Yang *et al.* (2022))	75.0%		28.84	5.92
CNN_DCTT_RCPL	**100%**		34.76	**0.00**
MANIQA_DCTT_RCPL	**100%**		34.76	**0.00**

Table 4.8 Performance on intra-coded images with Transformer model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
PIQE (Venkatanath *et al.* (2015))	28.6%		24.38	9.84
NIQE (Mittal *et al.* (2013))	17.9%		26.48	7.74
BRISQUE (Mittal *et al.* (2012))	28.6%		22.75	11.47
CNN_NR_IQA (Kang *et al.* (2014))	33.9%		28.78	5.44
CNN_NR_IQA_RE	67.9%	34.22	32.12	2.10
CNN_NR_IQA_NL	75.0%		32.20	2.02
MANIQA (Yang *et al.* (2022))	55.4%		27.67	6.55
CNN_DCTT_RCPL	92.9%		33.60	0.62
MANIQA_DCTT_RCPL	**96.4%**		34.22	**0.0007**

Table 4.9 Performance on inter-coded images with Transformer model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Tables 4.8 and 4.9 present the experimental results comparing our approach with state-of-the-art methods, using images encoded in *intra* and *inter* modes, respectively. The best results are highlighted in bold. PIQE (Venkatanath *et al.*, 2015), NIQE (Mittal *et al.*, 2013) and BRISQUE

(Mittal *et al.*, 2012) use packaged functions directly within Matlab for inference simulations. CNN_NR_IQA denotes our use of the pre-trained model from the study (Kang *et al.*, 2014), which assigns the same score to each patch of the Y component of the image (global score) and is tested with our database containing non-uniform distortions. CNN_NR_IQA_RE refers to our use of the retrained model from the study (Kang *et al.*, 2014). This model assigns a patch-level fidelity score (local score) to each patch of the R, G, and B components of the image. The CNN method with the suffix _NL indicates a configuration that incorporates the enhanced local normalization method described in Section 2.3. The MANIQA (Yang *et al.*, 2022) method indicates our use of the pre-trained Transformer model to test with our proposed databases. The methods suffixed with _DCTT and _RCPL denote the application of our new components proposed in this paper to Transformer-assisted and CNN-assisted systems, respectively.

Applying the proposed DCTT and RCPL components to both Transformer-assisted and CNN-assisted systems for intra-coded and inter-coded images results in improved accuracy and reduced quality difference compared to other pre-trained models. The benefits of using these newly proposed components in our Transformer-assisted system and re-training on our proposed databases are evident, with precision increasing from 75% to 100% for intra-coded images and from 55.4% to 96.4% for inter-coded images. Similarly, the precision of the CNN-assisted system improves from 46.4% to 100% for intra-coded images and from 33.9% to 92.9% for inter-coded images.

Compared to the CNN-assisted model, the Transformer-assisted model shows greater sensitivity in detecting small distortions in inter-coded images when using the newly proposed components. These components enable our model to better learn the quality degradations caused by local distortions due to transmission errors in corrupted images.

The experimental results demonstrate that integrating the DCTT and RCPL components significantly enhances model performance, validating their effectiveness in improving the quality assessment framework. The findings indicate that our proposed deep-learning assisted video list

decoding framework is robust across various types of encoded images, effectively handling both intra- and inter-coded scenarios.

Our framework helps to efficiently select the "best" candidate video from the list generated by list decoding. In subsequent ablation studies, we will further detail the impact and enhancements brought by each new module in our proposed model.

4.3.3 Ablation studies

We provide ablation experiments to illustrate the effect of different parameters and each component of our proposed method by comparing the results on our proposed dataset.

Proposed new components: Tables 4.10 and 4.11 compare the Transformer-assisted method from Section 3.3 to different simulation configurations incorporating the proposed DCTT and RCPL components. The first row of each table, labelled MANIQA_RETRAINED, indicates that we retrained the original Transformer model from (Yang *et al.*, 2022) using our created database, where images initially used in YUV format are converted to RGB format for training and inference, without applying the proposed DCTT color space conversion or using RCPL during training. A method with the suffix _DCTT indicates a configuration that applies our DCTT color space conversion to our database (Eq. (3.9)). The suffix _RCPL indicates a configuration that applies our improved RCPL component as part of the loss function F with α initially set to 0.5 in Eq. (3.10).

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
MANIQA_RETRAINED (Yang *et al.*, 2022)	100%	34.76	34.76	0.00
MANIQA_DCTT	**100%**		34.76	**0.00**
MANIQA_RCPL	67.9%		29.78	4.98
MANIQA_DCTT_RCPL	**100%**		34.76	**0.00**

Table 4.10 Performance on intra-coded images with Transformer model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
MANIQA_RETRAINED Yang *et al.* (2022)	92.9%	34.22	33.79	0.43
MANIQA_DCTT	78.6%		30.23	3.99
MANIQA_RCPL	**96.4%**		34.22	0.004
MANIQA_DCTT_RCPL	**96.4%**		34.22	**0.0007**

Table 4.11 Performance on inter-coded images with Transformer model (QP=37). $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

For intra-coded images, the proposed DCTT color space conversion consistently yields perfect results. For inter-coded images, incorporating the RCPL component significantly enhances performance, boosting precision from 93% to 96%. However, applying only the DCTT color space conversion does not lead to any performance improvement. In fact, it results in a decline. Notably, when both new components are applied to the simulation with inter-coded frames, there are obvious improvements, with precision rising from 93% to 96% and smaller differences between the average quality of the selected images and that of the intact versions. We also observe that adding the RCPL component to the Transformer-assisted method enhances the simulation results for both intra-coded and inter-coded frames.

Coefficients of RCPL: The coefficients α of F_1 and F_2 in Eq. (3.10) can be modified. We vary α to different values $(0.2, 0.5,$ and $1)$ to observe the variations in performance with inter-coded frames on the Transformer-assisted model, as shown in Table 4.12.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
MANIQA_DCTT_RCPL ($\alpha = 0.2$)	91.1%	34.22	34.21	0.01
MANIQA_DCTT_RCPL ($\alpha = 0.5$)	**96.4%**		34.22	**0.0007**
MANIQA_DCTT_RCPL ($\alpha = 1.0$)	78.6%		30.23	3.99

Table 4.12 Performance on inter-coded images with different coefficients in loss function on Transformer model. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB

For the Transformer-assisted model, an α value near 0.5 provides the best overall performance, achieving the highest accuracy in selecting the best candidate. However, an α value of 0.2 results in the smallest difference between the average quality of the selected image and the

average quality of the lossless image, indicating a trade-off between selection accuracy and quality consistency.

4.3.4 Parameter sensitivity analysis

We also conducted parameter sensitivity experiments to evaluate how sensitive the output of our proposed model is to changes in database variables.

Quantization Parameter (QP): The results reported in Table 4.13 and 4.14 demonstrate the performance on intra-coded frames and inter-coded frames, respectively, when evaluating systems trained with QP=37 on video encoded with QP settings of 37 and 22 in our database.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_DCTT_RCPL (QP=37)	100%	34.76	34.76	0.00
CNN_DCTT_RCPL (QP=22)	96.4%	43.62	43.62	0.002
MANIQA_DCTT_RCPL (QP=37)	100%	34.76	34.76	0.00
MANIQA_DCTT_RCPL (QP=22)	96.4%	43.62	43.62	0.002

Table 4.13 Performance on intra-coded images with our model by changing QP. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Methods	Accuracy	$\overline{S}_{\text{intact}}$	$\overline{S}_{\text{system}}$	$\overline{S}_{\text{diff}}$
CNN_DCTT_RCPL (QP=37)	92.9%	34.22	33.60	0.62
CNN_DCTT_RCPL (QP=22)	96.4%	42.61	42.49	0.12
MANIQA_DCTT_RCPL (QP=37)	96.4%	34.22	34.22	0.0007
MANIQA_DCTT_RCPL (QP=22)	100%	42.61	42.61	0.00

Table 4.14 Performance on inter-coded images with our model by changing QP. $\overline{S}_{\text{intact}}$, $\overline{S}_{\text{system}}$, and $\overline{S}_{\text{diff}}$ are expressed in dB.

Encoding with a low QP value, such as 22, results in more residual information being transmitted and present at the decoder. This means that bit errors have less chance of damaging critical information such as motion vectors or coding modes. Consequently, when an error is introduced to the video packet, it has a lesser impact on the decoded picture quality. In contrast, a higher QP transmits less residual information, making errors more likely to affect crucial elements of

the compressed video. For intra images, the system is not trained on this QP, so performance is worse, since it did not learn its subtle artifacts generated at QP=22 on residuals. For inter images, it is important to note that although we use the same formula to introduce errors by flipping a single bit in video streams with different QP values, lower QP retains redundant information and thus the test material is different for each QP. As a result, the received video candidates and bits flipped at QP=22 differ from those flipped at higher QP, introducing a degree of randomness in the distortion at different QP and limiting direct comparison between QPs. There is a possibility that flipped bits at QP=22 cause more noticeable local distortions, making it easier for our model to select the error-free version. The opposite is also true. However, overall, our model maintains good overall stability. For our small dataset, changing the QP does not significantly impact the model's performance.

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this thesis, we presented a deep-learning assisted video list decoding framework to identify the best candidate from a list in the context of unreliable video communications. As part of the framework, we introduced several IQA metrics based on deep-learning architecture to evaluate the quality of candidate videos. This framework selects the candidate video with the highest assessed visual quality from the candidate list.

We proposed two different NR IQA metrics to evaluate the quality of video content damaged by transmission errors:

- A CNN-based improved approach, which uses the same structure proposed in (Kang *et al.*, 2014). The original network was designed to evaluate image quality with global (uniform) distortions, and we created a more advanced metric based on CNN to evaluate image quality with local (non-uniform) distortions caused by transmission errors. We improve the local normalization and the loss function in the architecture to better perform in the presence of local (non-uniform) distortions.
- The second approach is based on the Transformer model to improve the performance of our IQA system. The basic model is from Yang *et al.* (2022), and we re-train the Transformer with "super-patches" supervised by reference metric scores, which is better in the presence of local discontinuities at horizontal and vertical boundaries of coded blocks between neighbouring patches. We also apply the improved loss functions in the Transformer model, which improves the model's performance.

We constructed a new database containing corrupted video using original videos sourced from public datasets. Our set of video sequences was encoded using the HEVC standard, and transmission errors were intentionally introduced to create non-uniformly corrupted images.

Simple error patterns were applied to the encoded video packets to simulate the effect of transmission over error-prone networks. We gathered 90 original videos with a resolution of 1920×1080 from public datasets. All videos underwent encoding and decoding without error concealment to showcase the various distortions.

Through comprehensive simulations and experiments, we demonstrated that our deep-learning-assisted framework performed very well, especially in scenarios involving inter-frame encoded images. We showed that the two proposed DL-based IQA metrics outperformed state-of-the-art methods in that context. Furthermore, by employing the new DCTT color space conversion and the improved ranking-constrained penalty loss function (RCPL), our system is better equipped to distinguish between well-received uniform patches and erroneous patches, thereby reducing the impact of transmission errors on the final video quality.

Our approach achieves remarkable decision accuracy, reaching 100% for intra-frame errors and 96% for inter-frame errors, which is a significant improvement over other evaluated methods. This is particularly important for applications in error-prone network environments where maintaining high visual quality is critical.

5.2 Further works and perspectives

The advancements presented in our research highlight the potential of deep-learning-based models in the field of visual quality assessment and list decoding. Our framework not only enhances the accuracy of candidate selection but also ensures a more reliable and efficient decoding process, thereby offering a substantial performance gain over conventional video receivers.

While our proposed framework represents a significant step forward in video list decoding, there are still areas that warrant further exploration. One of the key future research directions is to incorporate temporal information into the metrics. Currently, our approach primarily focuses on

spatial quality assessments. However, considering the temporal coherence and continuity of video frames could further enhance the accuracy and robustness of our method, particularly for inter-frame errors where temporal artifacts may be more pronounced.

By integrating temporal information, we can develop a more comprehensive assessment that accounts for the dynamic nature of video content. This is expected to not only improve the visual quality of the reconstructed video but also enhance the overall user experience. Future work will involve developing new temporal-aware deep learning models and expanding our dataset to include temporal distortions, allowing us to refine our framework and achieve even greater performance in video error correction. In summary, our research paves the way for more advanced video list decoding techniques, and by incorporating temporal information in the future, we aim to further elevate the standards of visual quality assessment and error correction in video transmission systems.

In future work, a more extensive database will be developed, and a true wireless simulator will be implemented to ensure that the proposed system functions adequately under real-world error scenarios. Additionally, we aim to combine and simulate the CRC-based error correction approach with our proposed system.

Future work will also focus on further refining our model and exploring its applicability to a broader range of video compression standards and error scenarios. The promising results obtained from this study lay the foundation for developing more robust and intelligent video transmission systems capable of delivering high-quality visual experiences even in challenging network conditions.

BIBLIOGRAPHY

Xiph.org Video Test Media [derf's collection]. Consulted at https://media.xiph.org/video/derf/.

Balatsoukas-Stimming, A., Parizi, M. B. & Burg, A. (2015). LLR-Based Successive Cancellation List Decoding of Polar Codes. *IEEE Transactions on Signal Processing*, 63(19), 5165-5179.

Bosse, S., Maniry, D., Wiegand, T. & Samek, W. (2016). A deep neural network for image quality assessment. *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3773-3777. doi: 10.1109/ICIP.2016.7533065.

Bossen, F. (2013). Common test conditions and software reference configurations. *JCTVC-L1100*, 12(7).

Boussard, V., Coulombe, S., Coudoux, F.-X. & Corlay, P. (2020a). Table-Free Multiple Bit-Error Correction Using the CRC Syndrome. *IEEE Access*, 8, 102357-102372. doi: 10.1109/ACCESS.2020.2998950.

Boussard, V., Golaghazadeh, F., Coulombe, S., Coudoux, F.-X. & Corlay, P. (2020b). Robust H.264 Video Decoding Using CRC-Based Single Error Correction And Non-Desynchronizing Bits Validation. *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1098-1102. doi: 10.1109/ICIP40778.2020.9190650.

Boussard, V., Coulombe, S., Coudoux, F.-X. & Corlay, P. (2021a). Enhanced CRC-based correction of multiple errors with candidate validation. *Signal Processing: Image Communication*, 99, 116475. doi: https://doi.org/10.1016/j.image.2021.116475.

Boussard, V., Coulombe, S., Coudoux, F.-X., Corlay, P. & Trioux, A. (2021b). CRC-Based Multi-Error Correction of H.265 Encoded Videos in Wireless Communications. *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pp. 1-5. doi: 10.1109/VCIP53242.2021.9675400.

Boussard, V., Coulombe, S., Coudoux, F.-X. & Corlay, P. (2022). CRC-Based Correction of Multiple Errors Using an Optimized Lookup Table. *IEEE Access*, 10, 23931-23947. doi: 10.1109/ACCESS.2022.3155457.

Brooks, A. C., Zhao, X. & Pappas, T. N. (2008). Structural similarity quality metrics in a coding context: exploring the space of realistic distortions. *IEEE Transactions on image processing*, 17(8), 1261–1273.

BT, R. et al. (2011). Studio encoding parameters of digital television for standard 4: 3 and wide-screen 16: 9 aspect ratios. *International Radio Consultative Committee International Telecommunication Union, Switzerland, CCIR Rep*.

Buades, A., Coll, B. & Morel, J.-M. (2011). Non-Local Means Denoising. *Image Processing On Line*, 1, 208–212. https://doi.org/10.5201/ipol.2011.bcm_nlm.

Caron, F. & Coulombe, S. (2015). Video Error Correction Using Soft-Output and Hard-Output Maximum Likelihood Decoding Applied to an H.264 Baseline Profile. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(7), 1161-1174. doi: 10.1109/TCSVT.2013.2291353.

Chang, C.-C. & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3), 1–27.

Chen, D., Wang, Y. & Gao, W. (2020a). No-reference image quality assessment: An attention driven approach. *IEEE Transactions on Image Processing*, 29, 6496–6506.

Chen, P., Li, L., Ma, L., Wu, J. & Shi, G. (2020b). RIRNet: Recurrent-in-recurrent network for video quality assessment. *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 834–842.

Chen, T., Zhang, X. & Shi, Y.-Q. (2003). Error concealment using refined boundary matching algorithm. *International Conference on Information Technology: Research and Education, 2003. Proceedings. ITRE2003.*, pp. 55–59.

Cheon, M., Yoon, S.-J., Kang, B. & Lee, J. (2021). Perceptual image quality assessment with transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 433–442.

Chikkerur, S., Sundaram, V., Reisslein, M. & Karam, L. J. (2011). Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison. *IEEE Transactions on Broadcasting*, 57(2), 165-182. doi: 10.1109/TBC.2011.2104671.

Chua, T.-K. & Pheanis, D. C. (2006). QoS evaluation of sender-based loss-recovery techniques for VoIP. *IEEE Network*, 20(6), 14–22.

Chubarau, A. & Clark, J. (2021). VTAMIQ: Transformers for Attention Modulated Image Quality Assessment. *arXiv preprint arXiv:2110.01655*.

Chung, B. & Yim, C. (2019). Bi-sequential video error concealment method using adaptive homography-based registration. *IEEE Transactions on circuits and systems for video technology*, 30(6), 1535–1549.

Collotta, M., Pau, G., Talty, T. & Tonguz, O. K. (2018). Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Communications Magazine*, 56(7), 125-131.

Committee, I. C. S. L. S. et al. (2007). IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11*.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.

Coudoux, F.-X., Gazalet, M. G., Corlay, P. & Rouvaen, J.-M. (1997). A perceptual approach to the reduction of blocking effect in DCT-coded images. *Journal of Visual Communication and Image Representation*, 8(4), 327–337.

De Simone, F., Naccari, M., Tagliasacchi, M., Dufaux, F., Tubaro, S. & Ebrahimi, T. (2009). Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel. *2009 International Workshop on Quality of Multimedia Experience*, pp. 204–209.

De Simone, F., Tagliasacchi, M., Naccari, M., Tubaro, S. & Ebrahimi, T. (2010). A H.264/AVC video database for the evaluation of quality metrics. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2430–2433.

Deshpande, R., Ragha, L. & Sharma, S. (2018). Video Quality Assessment through PSNR Estimation for Different Compression Standards. *Indonesian Journal of Electrical Engineering and Computer Science*, 11, 918-924. doi: 10.11591/ijeecs.v11.i3.pp918-924.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dosselmann, R. & Yang, X. D. (2011). A comprehensive assessment of the structural similarity index. *Signal, Image and Video Processing*, 5(1), 81–91.

Giannopoulos, M., Tsagkatakis, G., Blasi, S., Toutounchi, F., Mouchtaris, A., Tsakalides, P., Mrak, M. & Izquierdo, E. (2018). Convolutional neural networks for video quality assessment. *arXiv preprint arXiv:1809.10117*.

Golaghazadeh, F. (2019). *Enhanced quality reconstruction of erroneous video streams using packet filtering based on non-desynchronizing bits and UDP checksum-filtered list decoding*. (Ph.D. thesis, École de technologie supérieure).

Golaghazadeh, F., Coulombe, S., Coudoux, F.-X. & Corlay, P. (2017). Low complexity H.264 list decoder for enhanced quality real-time video over IP. *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1-6. doi: 10.1109/CCECE.2017.7946732.

Golaghazadeh, F., Coulombe, S., Coudoux, F.-X. & Corlay, P. (2018). Checksum-Filtered List Decoding Applied to H.264 and H.265 Video Error Correction. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8), 1993-2006. doi: 10.1109/TCSVT.2017.2686647.

Golestaneh, S. A., Dadsetan, S. & Kitani, K. M. (2021). No-Reference Image Quality Assessment via Transformers, Relative Ranking, and Self-Consistency. *arXiv preprint arXiv:2108.06858*.

Gomes, P., Olaverri-Monreal, C. & Ferreira, M. (2012). Making Vehicles Transparent Through V2V Video Streaming. *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 930-938.

Göring, S., Skowronek, J. & Raake, A. (2018). DeViQ–A deep no reference video quality model. *Electronic Imaging*, 2018(14), 1–6.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527–1554.

Hosu, V., Hahn, F., Jenadeleh, M., Lin, H., Men, H., Szirányi, T., Li, S. & Saupe, D. (2017). The Konstanz natural video database (KoNViD-1k). *2017 Ninth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6.

IEEE. (2016). IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, 1-3534.

Ioannis, K., Sr. Research, S. & Algorithm, V. (2018, March, 5). Dynamic optimizer - a perceptual video encoding optimization framework [tech blog describing how VMAF is used in an codec-agnostic encoding optimization framework]. Consulted at https://netflixtechblog.com/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f.

ITU-T RECOMMENDATION, P. (1999). Subjective video quality assessment methods for multimedia applications. *International telecommunication union*.

Jokela, T. & Lehtonen, E. (2007). Reed-Solomon Decoding Algorithms and Their Complexities at the DVB-H Link-Layer. *2007 4th International Symposium on Wireless Communication Systems*, pp. 752-756.

Kang, L., Ye, P., Li, Y. & Doermann, D. (2014). Convolutional Neural Networks for No-Reference Image Quality Assessment. *2014 IEEE Conference on Computer Vision and*

Pattern Recognition, pp. 1733-1740. doi: 10.1109/CVPR.2014.224.

Kazemi, M., Ghanbari, M. & Shirmohammadi, S. (2020). The performance of quality metrics in assessing error-concealed video quality. *IEEE Transactions on Image Processing*, 29, 5937–5952.

Koloda, J., Østergaard, J., Jensen, S. H., Sánchez, V. & Peinado, A. M. (2013a). Sequential error concealment for video/images by sparse linear prediction. *IEEE Transactions on Multimedia*, 15(4), 957–969.

Koloda, J., Sánchez, V. & Peinado, A. M. (2013b). Spatial error concealment based on edge visual clearness for image/video communication. *Circuits, Systems, and Signal Processing*, 32(2), 815–824.

Kossi, K., Coulombe, S., Desrosiers, C. & Gagnon, G. (2022). No-Reference Video Quality Assessment Using Distortion Learning And Temporal Attention. *IEEE Access*, 10, 1-1.

Krasula, L., Fliegel, K., Le Callet, P. & Klíma, M. (2016). On the accuracy of objective image and video quality models: New methodology for performance evaluation. *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6.

Krasula, L., Baveye, Y. & Le Callet, P. (2019). Training objective image and video quality estimators using multiple databases. *IEEE Transactions on Multimedia*, 22(4), 961–969.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097–1105.

Kung, W.-Y., Kim, C.-S. & Kuo, C.-C. (2006). Spatial and temporal error concealment techniques for video transmission over noisy channels. *IEEE transactions on circuits and systems for video technology*, 16(7), 789–803.

Kwok, W. & Sun, H. (1993). Multi-directional interpolation for spatial error concealment. *IEEE Transactions on consumer electronics*, 39(3), 455–460.

Laghari, A. A., Shahid, S., Yadav, R., Karim, S., Khan, A., Li, H. & Shoulin, Y. (2023). The state of art and review on video streaming. *Journal of High Speed Networks*, 29(3), 211–236.

Lam, W.-M., Reibman, A. R. & Liu, B. (1993). Recovery of lost or erroneously received motion vectors. *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5, 417–420.

Lasmar, N.-E., Stitou, Y. & Berthoumieu, Y. (2009). Multiscale skewed heavy tailed model for texture analysis. *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 2281–2284.

LeCun, Y., Bengio, Y. et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.

Lee, C., Cho, S., Choe, J., Jeong, T., Ahn, W. & Lee, E. (2006). Objective video quality assessment. *Optical engineering*, 45(1), 017004.

Lee, P.-J., Chen, H. H. & Chen, L.-G. (2004). A new error concealment algorithm for H.264 video transmission. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pp. 619–622.

Lei, X. (2013, September, 14) [Blog]. Consulted at http://blog.csdn.NET/leixiaohua1020/article/details/11694369.

Leijnen, S. & Veen, F. v. (2020). The neural network zoo. *Multidisciplinary Digital Publishing Institute Proceedings*, 47(1), 9.

Li, D., Jiang, T. & Jiang, M. (2019). Quality assessment of in-the-wild videos. *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2351–2359.

Li, S., Zhang, F., Ma, L. & Ngan, K. N. (2011). Image quality assessment by separately evaluating detail losses and additive impairments. *IEEE Transactions on Multimedia*, 13(5), 935–949.

Li, Y., Feng, L., Xu, J., Zhang, T., Liao, Y. & Li, J. (2021). Full-Reference And No-Reference Quality Assessment For Compressed User-Generated Content Videos. *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6.

Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A. & Manohara, M. (2016, June, 6). Toward A Practical Perceptual Video Quality Metric [tech blog with VMAF's open sourcing on Github]. Consulted at https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

Lin, H., Hosu, V. & Saupe, D. (2019). KADID-10k: A Large-scale Artificially Distorted IQA Database. *2019 Tenth International Conference on Quality of Multimedia Experience*

(QoMEX), pp. 1–3.

Lin, H., Hosu, V. & Saupe, D. (2020). DeepFL-IQA: Weak Supervision for Deep IQA Feature Learning. *arXiv preprint arXiv:2001.08113*.

Lin, J., Zhang, Y., Li, N. & Jiang, H. (2022). Joint Source-Channel Decoding of Polar Codes for HEVC-Based Video Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.*, 18(4). doi: 10.1145/3502208.

Liu, C., Ma, R. & Zhang, Z. (2012). Error concealment for whole frame loss in HEVC. *Advances on Digital Television and Wireless Multimedia Communications: 9th International Forum on Digital TV and Wireless Multimedia Communication, IFTC 2012, Shanghai, China, November 9-10, 2012. Proceedings*, pp. 271–277.

Liu, W., Duanmu, Z. & Wang, Z. (2018). End-to-End Blind Quality Assessment of Compressed Videos Using Deep Neural Networks. *ACM Multimedia*, pp. 546–554.

Liu, X., Wu, S., Wang, Y., Zhang, N., Jiao, J. & Zhang, Q. (2020). Exploiting Error-Correction-CRC for Polar SCL Decoding: A Deep Learning-Based Approach. *IEEE Transactions on Cognitive Communications and Networking*, 6(2), 817-828. doi: 10.1109/TCCN.2019.2946358.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022.

Ma, K., Liu, W., Zhang, K., Duanmu, Z., Wang, Z. & Zuo, W. (2018). End-to-End Blind Image Quality Assessment Using Deep Neural Networks. *IEEE Transactions on Image Processing*, 27(3), 1202-1213. doi: 10.1109/TIP.2017.2774045.

Mercat, A., Viitanen, M. & Vanne, J. (2020). UVG Dataset: 50/120fps 4K Sequences for Video Codec Analysis and Development. *Proceedings of the 11th ACM Multimedia Systems Conference*, (MMSys '20), 297–302. doi: 10.1145/3339825.3394937.

Mittal, A., Moorthy, A. K. & Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12), 4695–4708.

Mittal, A., Soundararajan, R. & Bovik, A. C. (2013). Making a "Completely Blind" Image Quality Analyzer. *IEEE Signal Processing Letters*, 20(3), 209-212. doi: 10.1109/LSP.2012.2227726.

Mittal, A., Saad, M. A. & Bovik, A. C. (2016). A Completely Blind Video Integrity Oracle. *IEEE Transactions on Image Processing*, 25(1), 289-300. doi: 10.1109/TIP.2015.2502725.

Moorthy, A. K. & Bovik, A. C. (2011). Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality. *IEEE Transactions on Image Processing*, 20(12), 3350-3364. doi: 10.1109/TIP.2011.2147325.

Narwaria, M. & Lin, W. (2010). Objective image quality assessment based on support vector regression. *IEEE Transactions on Neural Networks*, 21(3), 515–519.

Ni, H. & Li, Y. (2017). Spatial error concealment algorithm based on adaptive edge threshold and directional weight. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(08), 1754014.

O'Shea, K. & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Peng, Q., Yang, T. & Zhu, C. (2002). Block-based temporal error concealment for video packet using motion vector extrapolation. *IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions*, 1, 10–14.

Pinson, M. H. (2013). The Consumer Digital Video Library [Best of the Web]. *IEEE Signal Processing Magazine*, 30(4), 172-174. doi: 10.1109/MSP.2013.2258265.

Ponomarenko, N., Ieremeiev, O., Lukin, V., Egiazarian, K., Jin, L., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F. et al. (2013). Color image database TID2013: Peculiarities and preliminary results. *European workshop on visual information processing (EUVIP)*, pp. 106–111.

Poynton, C. A. (1996). *A technical introduction to digital video*. John Wiley & Sons, Inc.

Qian, L., Pan, T., Zheng, Y., Zhang, J., Li, M., Yu, B. & Wang, B. (2021). No-Reference Nonuniform Distorted Video Quality Assessment Based on Deep Multiple Instance Learning. *IEEE MultiMedia*, 28(1), 28-37. doi: 10.1109/MMUL.2020.3034338.

Rassool, R. (2017). VMAF reproducibility: Validating a perceptual practical video quality metric. *2017 IEEE international symposium on broadband multimedia systems and broadcasting (BMSB)*, pp. 1–2.

Rehman, A. & Wang, Z. (2012). Reduced-Reference Image Quality Assessment by Structural Similarity Estimation. *IEEE Transactions on Image Processing*, 21(8), 3378-3389. doi: 10.1109/TIP.2012.2197011.

Ridnik, T., Ben-Baruch, E., Noy, A. & Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*.

Rongfu, Z., Yuanhua, Z. & Xiaodong, H. (2004). Content-adaptive spatial error concealment for video communication. *IEEE Transactions on Consumer Electronics*, 50(1), 335–341.

Ruderman, D. L. (1994). The statistics of natural images. *Network: Computation in Neural Systems*, 5(4), 517-548. doi: 10.1088/0954-898X_5_4_006.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 211–252.

Saad, M. A., Bovik, A. C. & Charrier, C. (2012). Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain. *IEEE Transactions on Image Processing*, 21(8), 3339-3352. doi: 10.1109/TIP.2012.2191563.

Saad, M. A., Bovik, A. C. & Charrier, C. (2014). Blind prediction of natural video quality. *IEEE Transactions on Image Processing*, 23(3), 1352–1365.

Sabeva, G. & al. (2006). Robust Decoding of H.264 Encoded Video Transmitted over Wireless Channels. *2006 IEEE Workshop on Multimedia Signal Processing*, pp. 9-13. doi: 10.1109/MMSP.2006.285258.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E. & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.

Sara, U., Akter, M. & Uddin, M. S. (2019). Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, 7(3), 8–18.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.

Schölkopf, B., Smola, A. J., Williamson, R. C. & Bartlett, P. L. (2000). New support vector algorithms. *Neural computation*, 12(5), 1207–1245.

Seiler, J., Schoberi, M. & Kaup, A. (2013). Spatio-temporal error concealment in video by denoised temporal extrapolation refinement. *2013 IEEE International Conference on Image Processing*, pp. 1613–1616.

Seshadrinathan, K. & Bovik, A. C. (2011). Temporal hysteresis model of time varying subjective video quality. *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1153–1156.

Seshadrinathan, K., Soundararajan, R., Bovik, A. C. & Cormack, L. K. (2010). Study of Subjective and Objective Quality Assessment of Video. *IEEE Transactions on Image Processing*, 19(6), 1427-1441. doi: 10.1109/TIP.2010.2042111.

Sheikh, H. (2005). LIVE image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*.

Sheikh, H. R. & Bovik, A. C. (2006). Image information and visual quality. *IEEE Transactions on image processing*, 15(2), 430–444.

Shih, H.-C., Wang, C.-T. & Huang, C.-L. (2018). Spiral-like pixel reconstruction algorithm for spatiotemporal video error concealment. *IEEE Access*, 6, 6370–6381.

Shirani, S., Kossentini, F. & Ward, R. (1999). Error concealment methods, a comparative study. *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 99TH8411)*, 2, 835–840.

Shrestha, A. & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, 7, 53040–53065.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sinno, Z. & Bovik, A. C. (2018a). Large scale subjective video quality study. *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 276–280.

Sinno, Z. & Bovik, A. C. (2018b). Large-scale study of perceptual video quality. *IEEE Transactions on Image Processing*, 28(2), 612–627.

Sobolewski, J. S. (2003). Cyclic Redundancy Check. In *Encyclopedia of Computer Science* (pp. 476–479). GBR: John Wiley and Sons Ltd.

Søgaard, J., Krasula, L., Shahid, M., Temel, D., Brunnström, K. & Razaak, M. (2016). Applicability of existing objective metrics of perceptual quality for adaptive video streaming. *Electronic Imaging*, 2016(13), 1–7.

Sullivan, G. J. & Wiegand, T. (2005). Video Compression - From Concepts to the H.264/AVC Standard. *Proceedings of the IEEE*, 93(1), 18-31.

Sullivan, G. J., Ohm, J., Han, W. & Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1649-1668.

Systems, C. (2016, June, 1). Cisco Visual Networking Index: Forecast and Methodology, 2015-2020, Document ID:1465272001663118 [White paper].

Sze, V. & al. (2014). *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. Springer Publishing.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tang, J., Dong, Y., Xie, R., Gu, X., Song, L., Li, L. & Zhou, B. (2020). Deep Blind Video Quality Assessment for User Generated Videos. *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 156–159.

Then, X. (2019, January). Overview of Video/Image Quality Evaluation (1). Consulted at https://zhuanlan.zhihu.com/p/54539091.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D. & Li, L.-J. (2016). YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2), 64–73.

Tu, Z., Wang, Y., Birkbeck, N., Adsumilli, B. & Bovik, A. C. (2021). UGC-VQA: Benchmarking blind video quality assessment for user generated content. *IEEE Transactions on Image Processing*, 30, 4449–4464.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, pp. 5998–6008.

Vega, M. T., Mocanu, D. C., Famaey, J., Stavrou, S. & Liotta, A. (2017). Deep learning for quality assessment in live video streaming. *IEEE signal processing letters*, 24(6), 736–740.

Venkatanath, N., Praneeth, D., Bh, M. C., Channappayya, S. S. & Medasani, S. S. (2015). Blind image quality evaluation using perception based features. *2015 twenty first national conference on communications (NCC)*, pp. 1–6.

Vranješ, M., Rimac-Drlje, S. & Grgić, K. (2013). Review of objective video quality metrics and performance comparison using different databases. *Signal Processing: Image Communication*, 28(1), 1-19. doi: https://doi.org/10.1016/j.image.2012.10.003.

Wang, H., Gan, W., Hu, S., Lin, J. Y., Jin, L., Song, L., Wang, P., Katsavounidis, I., Aaron, A. & Kuo, C.-C. J. (2016). MCL-JCV: A JND-based H.264/AVC video quality assessment dataset. *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1509-1513.

doi: 10.1109/ICIP.2016.7532610.

Wang, Y. & Zhu, Q.-F. (1998). Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5), 974–997.

Wang, Y.-K., Hannuksela, M. M., Varsa, V., Hourunranta, A. & Gabbouj, M. (2002). The error concealment feature in the H.26L test model. *Proceedings. International Conference on Image Processing*, 2, II–II.

Wang, Y., Inguva, S. & Adsumilli, B. (2019). YouTube UGC dataset for video compression research. *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–5.

Wang, Z., Simoncelli, E. P. & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2, 1398–1402.

Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. (2004a). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600–612.

Wang, Z., Lu, L. & Bovik, A. C. (2004b). Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2), 121–132.

Wiegand, T., Sullivan, G. J., Bjontegaard, G. & Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), 560-576.

Wu, J., Liu, X. & Yoo, K.-Y. (2008). A temporal error concealment method for H.264/AVC using motion vector recovery. *IEEE Transactions on Consumer Electronics*, 54(4), 1880–1885.

Xing, F., Wang, Y.-G., Wang, H., Li, L. & Zhu, G. (2021). StarVQA: Space-Time Attention for Video Quality Assessment. *arXiv preprint arXiv:2108.09635*.

Xinghao, T. Overview of Video/Image Quality Evaluation (2). Consulted at https://zhuanlan.zhihu.com/p/54950132.

Xu, K., Liao, L., Xiao, J., Chen, C., Wu, H., Yan, Q. & Lin, W. (2023). Local Distortion Aware Efficient Transformer Adaptation for Image Quality Assessment. *arXiv preprint arXiv:2308.12001*.

Yang, S., Wu, T., Shi, S., Lao, S., Gong, Y., Cao, M., Wang, J. & Yang, Y. (2022). Maniqa: Multi-dimension attention network for no-reference image quality assessment. *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1191–1200.

Ying, Z., Mandal, M., Ghadiyaram, D. & Bovik, A. (2021). Patch-VQ:'Patching Up'the Video Quality Problem. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14019–14029.

You, J. (2021). Long Short-term Convolutional Transformer for No-Reference Video Quality Assessment. *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2112–2120.

You, J. & Korhonen, J. (2021). Transformer for image quality assessment. *2021 IEEE international conference on image processing (ICIP)*, pp. 1389–1393.

Zabihi, S. M., Ghanei-Yakhdan, H. & Mehrshad, N. (2021). Content-based hybrid error concealment approach for packet video communication over the noisy channels. *Multimedia Tools and Applications*, 80(8), 12335–12365.

Zhang, W., Ma, K., Yan, J., Deng, D. & Wang, Z. (2020). Blind Image Quality Assessment Using a Deep Bilinear Convolutional Neural Network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1), 36-47. doi: 10.1109/TCSVT.2018.2886771.

Zhang, Y., Gao, X., He, L., Lu, W. & He, R. (2019). Blind Video Quality Assessment With Weakly Supervised Learning and Resampling Strategy. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8), 2244-2255. doi: 10.1109/TCSVT.2018.2868063.

Zhang, Y., Coulombe, S., Coudoux, F.-X., Trioux, A. & Corlay, P. (2023). Optimisation du décodage par liste de vidéos corrompues basée sur une architecture CNN. *22ème édition de la conférence COmpression et REprésentation des Signaux Audiovisuels*. Consulted at https://hal.science/hal-04246635.

Zhang, Y., Coulombe, S., Coudoux, F.-X., Guichemerre, A. & Corlay, P. (2024). Deep learning-assisted video list decoding in error-prone video transmission systems. *Submitted to IEEE Access*.

Zhao, Y., Wang, G., Tang, C., Luo, C., Zeng, W. & Zha, Z.-J. (2021). A Battle of Network Structures: An Empirical Study of CNN, Transformer, and MLP. *arXiv preprint arXiv:2108.13002*.

Zhou, W. & Chen, Z. (2020). Deep local and global spatiotemporal feature aggregation for blind video quality assessment. *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 338–341.