

Étude et implémentation d'algorithmes de quantification basés sur des modèles psycho-visuels en vue de valider la qualité en temps réel de codecs vidéo

Ing. T. JOYE
HELMo-Gramme – Liège

Cet article concerne l'étude et l'implémentation d'algorithmes basés sur des modèles psycho-visuels pour évaluer la qualité des codecs vidéo en temps réel. Il examine les mécanismes de perception humaine, les formats vidéo et d'images, ainsi que les algorithmes de compression. Il analyse aussi plusieurs méthodes de comparaison d'images, telles que le PSNR, le SSIM et le VMAF. Enfin, il propose des résultats produits par mon programme pour mettre en évidence la théorie.

Mots-clés : Algorithmes, Modèles psycho-visuels, Qualité vidéo, Codecs vidéo, Perception humaine, Formats vidéo, Compression d'images.

This article focuses on the study and implementation of algorithms based on psycho-visual models to evaluate the quality of real-time video codecs. It examines the mechanisms of human perception, video and image formats, as well as compression algorithms. It also analyses several image comparison methods, such as PSNR, SSIM, and VMAF. Finally, it presents results produced by my program to highlight the theory.

Keywords: Algorithms, Psycho-visual models, Video quality, Video codecs, Human perception, Video formats, Image compression.

1. Contexte et objectifs du travail

Dans le domaine du streaming vidéo, l'objectif primordial est de fournir aux utilisateurs la meilleure expérience visuelle possible. Un des facteurs clés pour atteindre cet objectif est de s'assurer que la qualité des vidéos soit optimisée en temps réel, tout en utilisant un minimum de bande passante et de ressources.

Cependant, atteindre cet équilibre est un véritable défi pour EVS Broadcast Equipment. D'un côté, EVS veut que chaque image soit parfaite pour le spectateur, mais d'un autre côté, l'entreprise doit minimiser l'utilisation de la bande passante et des ressources. C'est là que mon travail de fin d'études [1] entre en jeu. Ce travail est centré sur le développement d'algorithmes de quantification basés sur des modèles psycho-visuels pour résoudre cette problématique.

Fondamentalement, ce travail a pour but de répondre à la question suivante : comment détecter les défauts d'image en temps réel ? Ici, le terme "temps réel" signifie que cette analyse doit être effectuée pendant que la vidéo est en cours de visionnage, exactement comme lorsqu'une personne regarde un flux vidéo à la télévision. Pour rappel, une vidéo est composée d'une succession d'images. Il fallait donc inspecter chaque image pour détecter les éventuelles erreurs et le défi était de réaliser cette analyse sans ajouter de latence ou de saccade à la vidéo.

Il s'agit d'une tâche complexe qui nécessite de comprendre non seulement comment une vidéo est encodée et transmise, mais aussi comment nos yeux et notre cerveau perçoivent les images.

Les modèles psycho-visuels reproduisent la manière dont le système visuel humain traite les informations visuelles. Ils sont déployés dans ces algorithmes pour détecter toute distorsion ou anomalie qui pourrait affecter la perception visuelle de l'utilisateur.

L'application de ces algorithmes sur des flux vidéo en temps réel permet à EVS d'optimiser la qualité visuelle et l'efficacité du streaming. En quantifiant en temps réel et en minimisant les erreurs potentielles, ces algorithmes favorisent une utilisation plus efficace de la bande passante et une qualité optimale au consommateur.

2. D'où viennent les erreurs

2.1. Les enjeux de la compression d'images et de vidéos

La compression d'images est une étape essentielle pour optimiser le stockage et la transmission de données visuelles, en réduisant leur taille tout en maintenant une qualité acceptable. Les enjeux majeurs de la compression d'images sont les suivants:

- **Taille du fichier** : Il est primordial de minimiser la taille du fichier image afin de diminuer les coûts liés au stockage et à la transmission, sans dégrader significativement sa qualité.
- **Qualité d'images** : La conservation de la qualité d'images est un facteur déterminant pour garantir une expérience visuelle agréable et une utilisation adaptée des données compressées dans diverses applications.
- **Complexité de l'algorithme** : Un algorithme de compression doit être suffisamment simple pour être implémenté efficacement sur différents dispositifs et plateformes, tout en étant assez sophistiqué pour fournir un bon compromis entre la taille du fichier et la qualité d'images.
- **Adaptabilité** : La méthode de compression doit pouvoir traiter divers types d'images et tenir compte des particularités propres à chaque format.
- **Résistance aux erreurs et à la corruption** : Une méthode de compression robuste doit être capable de détecter et corriger des erreurs ou corruptions éventuelles, afin d'assurer un affichage correct de l'image compressée. Néanmoins, il n'est pas toujours possible de supprimer totalement toutes les erreurs. C'est d'ailleurs pour cette raison que les artefacts de compression existent.

2.2. Compression sans perte

La compression sans perte préserve l'intégralité des informations contenues dans l'image d'origine. Voici quelques-uns des formats les plus courants qui utilisent des algorithmes de compression sans perte :

- **PNG (Portable Network Graphics)** : PNG est un format de compression sans perte souvent utilisé pour les images en ligne. Il utilise l'algorithme de compression DEFLATE et est particulièrement efficace pour les images avec des zones de couleurs uniformes.
- **TIFF (Tagged Image File Format)** : TIFF est un format d'images flexible capable de gérer plusieurs types de compressions, y compris certaines sans perte. Par exemple, la compression LZW peut être utilisée pour compresser les images TIFF de manière non destructive.

2.3. Compression avec pertes

La compression avec perte réduit la taille des fichiers en supprimant certaines informations de l'image d'origine, ce qui peut entraîner une perte de qualité.

- **JPEG (Joint Photographic Experts Group) :** JPEG est le format de compression avec perte le plus couramment utilisé pour les images numériques. Il utilise la transformation discrète en cosinus (DCT) pour compresser les données. La qualité et la taille du fichier peuvent être ajustées en utilisant un facteur de qualité, où une valeur inférieure entraîne une plus grande compression et une qualité d'image réduite.
- **WebP :** WebP est un format d'images conçu par Google, offrant une compression avec perte et sans perte. Pour la compression avec perte, il est connu pour offrir une qualité d'images similaire à JPEG avec une taille de fichier inférieure.

3. Les défauts classiques d'une image

La compression d'images peut entraîner des défauts visuels en raison des techniques utilisées pour réduire la taille du fichier. Parmi les défauts courants provoqués par la compression [2], on compte :

- **Le blocage :** Figure 3-1. La division de l'image en blocs pour faciliter la compression peut provoquer des discontinuités entre les blocs adjacents, donnant un aspect de mosaïque.



Figure 3-1 : Exemple d'effet de blocage.[3]

- **Le flou :** Figure 3-2. Lors de la quantification, certaines informations de hautes fréquences sont perdues, ce qui peut entraîner une perte de netteté et un aspect flou sur l'image compressée.



Figure 3-2 : Exemple de flou.[4]

- **Les contours irréguliers :** Figure 3-3. Les artefacts dus à la quantification peuvent également causer des contours irréguliers ou anguleux dans les zones contrastées de l'image.

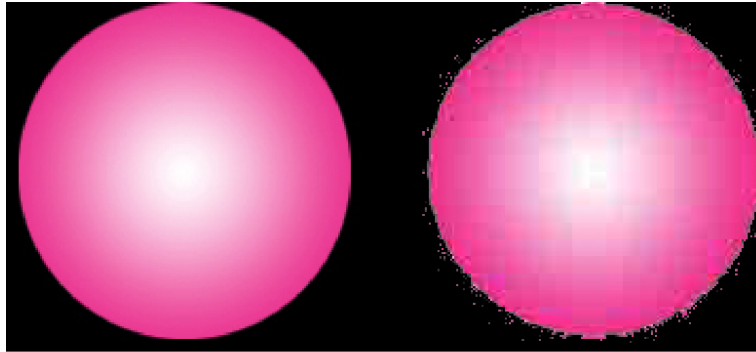


Figure 3-3 : Exemple de contours irréguliers.[2]

- **Le ringing** : Figure 3-4. Les erreurs d'arrondi dues à la quantification peuvent générer des motifs circulaires autour des contours ou autour des objets lumineux sur un fond sombre, connus sous le nom d'anneaux (ringing) ou d'effet Gibbs.



Figure 3-4 : Exemple de ringing appelé aussi effet de Gibbs. [5]

- **Le bruit numérique et la propagation d'erreurs** : Figure 3-5. Dans certains algorithmes de compression avec perte, les erreurs se propagent et s'amplifient au fil du temps, surtout en cas d'utilisation de la prédiction inter-trame lors de compression vidéo.

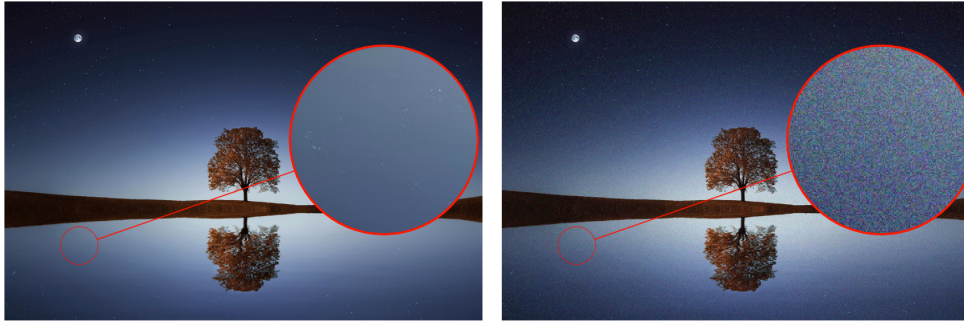


Figure 3-5 : Exemple de bruit numérique et de propagation d'erreur. [6]

4. Composition d'une image

Dans cette section, nous allons explorer les différents espaces colorimétriques courants, pour pouvoir par la suite, analyser une séquence vidéo complète dans la section 6.5.

Dans le contexte du traitement numérique des images et des vidéos, un espace colorimétrique est un outil essentiel. En effet, chaque image est composée d'un ensemble de pixels, et chaque pixel a une valeur colorimétrique. Cette valeur est définie dans un espace colorimétrique spécifique.

Pour imaginer un espace colorimétrique, imaginez un modèle mathématique qui sert à décrire les couleurs. A l'intérieur de cet espace, chaque couleur est représentée par trois ou quatre composantes. En résumé, un espace colorimétrique est une manière de représenter les couleurs à l'aide de nombres, ce qui est très pratique pour le traitement numérique des images.

4.1. Espaces colorimétriques

RGB

Le format RGB (Red, Green, Blue) [7] est basé sur l'espace colorimétrique qui utilise les trois couleurs primaires pour représenter les images. Chaque pixel est constitué d'une combinaison des trois couleurs, Figure 4-1, avec des valeurs (pour des couleurs sur 8 bits) comprises entre 0 et 255 pour chaque canal (R, G, B). Une valeur de (0, 0, 0) représente le noir tandis que (255, 255, 255) donne du blanc.

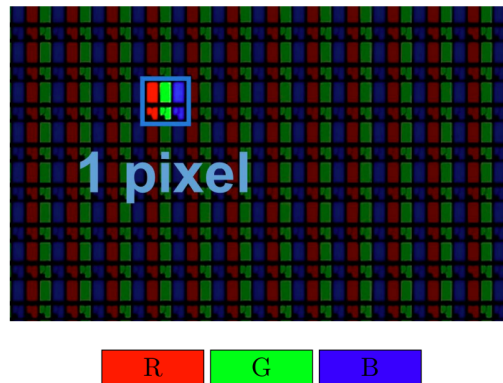


Figure 4-1 : Structure d'un pixel en RGB [8]

Le principal avantage du format RGB est qu'il est intuitif et facile à comprendre. Cependant, il n'est pas très efficace pour compresser les vidéos car il stocke toute l'information de couleur. La Figure 4-2 nous montre un exemple d'image en RGB.

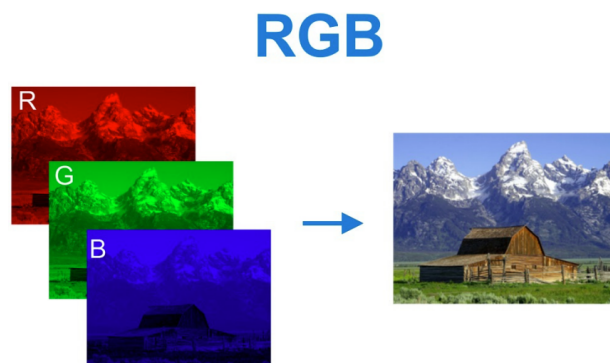


Figure 4-2 : Exemple d'une image sous forme RGB [9]

YUV

Le format YUV (ou YCbCr) [10] représente les images de manière différente que le format RGB en se basant sur l'espace colorimétrique YUV. Il sépare les informations en trois composantes : la luminance (Y) et deux composantes de chrominance (U et V). Cette manière de représenter les données est plus proche des caractéristiques perceptuelles de l'œil humain, qui est plus sensible aux variations de luminance qu'aux variations de chrominance.

Luminance: La luminance représente la partie claire ou sombre d'une image. Elle indique la quantité totale de lumière émise ou réfléchiée par chaque pixel sans prendre en compte sa couleur. Autrement dit, une image en niveaux de gris est constituée

uniquement d'informations concernant la luminance. La composante Y du format YUV correspond à cette information. Voici la Figure 4-3 qui montre à quoi ressemble la décomposition d'une image en ses niveaux de luminances.

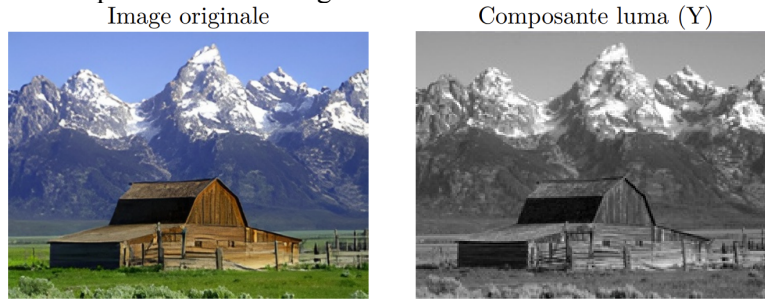


Figure 4-3 : Image décomposée avec sa luminance [9]

Chrominance: La chrominance, quant à elle, mesure la différence entre chaque couleur dans une image et une référence blanche. Elle renseigne sur la saturation et pureté d'une couleur sans prendre en compte son niveau de clarté ou d'obscurité. Les champs U et V du format YUV sont dédiés au stockage des données relatives à la chrominance. Dans l'exemple d'une image grise, les valeurs de U et V qui représentent la chrominance seront inutilisées car l'image ne contient pas d'information de couleur. Au contraire, une image en couleur sera décomposée suivant la Figure 4-4.

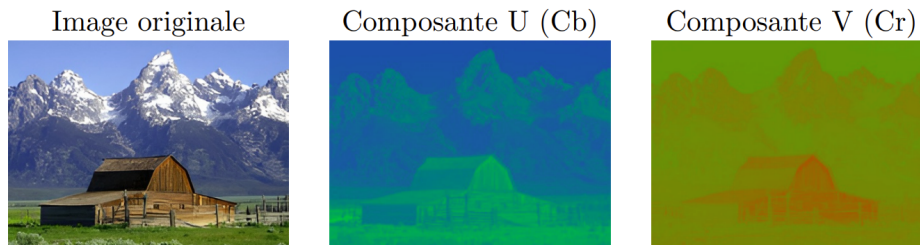


Figure 4-4 : Image décomposée avec ses composantes de chrominances [9]

En combinant ces deux informations - luminance et chrominance - on peut reconstituer l'image complète avec ses informations de couleurs tout en exploitant plus efficacement l'espace mémoire pour le stockage. Voici un exemple Figure 4-5.

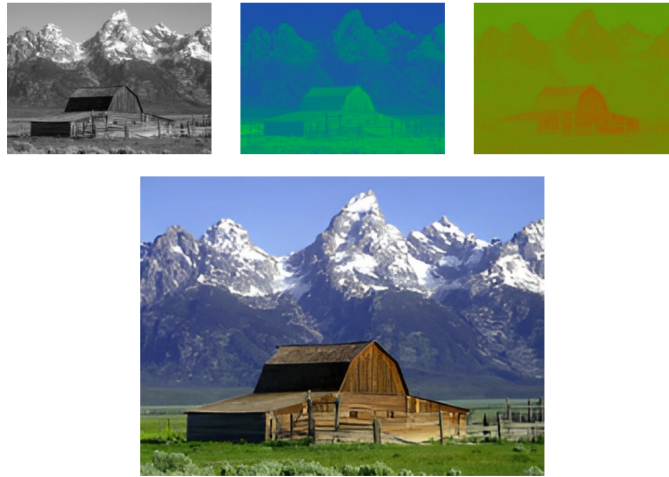


Figure 4-5 : Image sous forme YUV [9]

Récapitulatif des espaces colorimétriques

La compréhension des différences entre les formats vidéo RGB, YUV est essentielle pour choisir efficacement le format approprié en fonction des contraintes techniques (qualité d'image) et pratiques (stockage).

5. Différents algorithmes retenus

5.1. Choix final des algorithmes

Dans ce travail, plusieurs algorithmes ont été exploités en vue d'évaluer la qualité des images, en prenant en compte leurs avantages et inconvénients respectifs.

Le choix final des algorithmes utilisés a été basé sur leur complémentarité afin de couvrir un large spectre de types d'images à tester. Les avantages et inconvénients de chaque algorithme seront détaillés dans les sections correspondantes.

Les algorithmes choisis pour réaliser l'évaluation de la qualité des images sont les suivants :

- PSNR (Peak Signal-to-Noise Ratio)
- SSIM (Structural Similarity Index Measure)
- VMAF (Video Multi-Method Assessment Fusion)

En combinant ces trois méthodes, il est possible d'obtenir une évaluation plus précise et complète de la qualité d'une image. De plus, en les combinant, nous pouvons tester et vérifier davantage de types de distorsions possibles, telles que celles mentionnées dans la section 3: le blocage, le flou, les contours irréguliers, le ringing, le bruit, etc... Par conséquent, parmi les algorithmes sélectionnés, nous devrions disposer d'au moins un qui puisse identifier efficacement chacune de ces imperfections.

5.2. PSNR : Peak Signal to Noise Ratio

Présentation de l'algorithme

Le PSNR (Peak Signal-to-Noise Ratio) [11] est une mesure de qualité couramment employée pour évaluer la performance de reconstruction d'une image ou d'une vidéo en la comparant avec l'original. Cette méthode quantifie la qualité en se basant sur le rapport entre l'amplitude maximale du signal et le bruit introduit à la suite d'un processus de compression ou d'autres types de traitement.

Dans le contexte des images et vidéos au format YUV, l'algorithme PSNR se concentre principalement sur les composantes Y (luma). En effet, les valeurs des composantes U et V sont souvent négligées dans ce calcul car elles ont moins d'impact sur notre perception visuelle. Néanmoins, il est également possible d'étendre la mesure PSNR aux autres plans des données si cela s'avère nécessaire.

Pour calculer le PSNR, on utilise la formule suivante :

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad \text{Équation 5-1}$$

où MAX représente l'amplitude maximale du signal (par exemple, 255 pour les images 8 bits), et MSE correspond à l'erreur quadratique moyenne (Mean Squared Error) entre l'image originale et celle traitée. Le résultat sera exprimé en décibels (dB).

L'erreur quadratique moyenne est déterminée par cette expression :

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i,j) - g(i,j)]^2 \quad \text{Équation 5-2}$$

Ici, $f(i,j)$ fait référence au pixel de l'image originale, $g(i,j)$ au pixel de l'image traitée, et M et N représentent les dimensions de l'image.

Généralement, un PSNR élevé indique une meilleure qualité d'image. L'exemple illustré dans la Figure 5-1, montre comment le calcul du PSNR peut s'appliquer à une image dégradée de différentes manières. La figure présente la même image avec différents niveaux de dégradation, ainsi que la valeur du PSNR correspondante pour chaque cas.

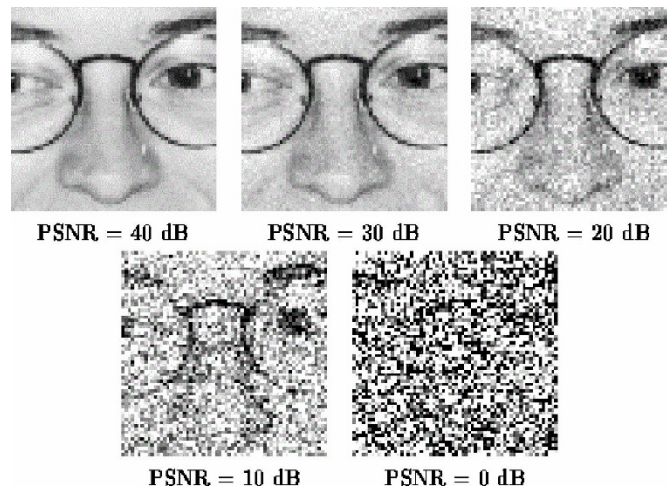


Figure 5-1 : Exemple de bruit numérique et de propagation d'erreur. [12]

Il est aisé de constater que lorsque le PSNR diminue, la qualité de l'image se détériore également.

Cependant, il est important de souligner que cette mesure présente des limites en matière de perception humaine de la qualité d'une image ou d'une vidéo. Le PSNR repose sur une approche mathématique simple qui peut parfois ne pas concorder avec l'appréciation subjective des observateurs. En d'autres termes, une valeur PSNR élevée ne garantit pas nécessairement une qualité perçue optimale par l'être humain. Nous verrons plus en détail les limites de cette méthode dans la section Inconvénients.

Avantages

L'algorithme PSNR (Peak Signal to Noise Ratio) est toujours une méthode de mesure largement utilisée pour estimer la qualité d'une image ou d'une vidéo. Voici quelques avantages de cet algorithme :

- **La facilité de compréhension:** Le PSNR est basé sur le ratio entre la puissance du signal et la puissance du bruit, ce qui en fait une mesure facile à comprendre et à interpréter.

- **La facilité de calcul:** Le calcul du PSNR ne nécessite que des opérations arithmétiques simples, ce qui le rend rapide et peu coûteux en termes de ressources informatiques.
- **Un usage répandu:** Le PSNR est largement utilisé dans les applications liées au traitement numérique des images et des vidéos pour évaluer la qualité des images ou vidéos compressées ou restaurées, telles que les codecs vidéo et les algorithmes de restauration d'images.
- **La sensibilité aux erreurs:** Le PSNR est sensible aux erreurs dans l'image reconstruite, en particulier les erreurs dues à la compression ou au bruit. Cette sensibilité permet une évaluation précise du niveau de dégradation de la qualité.

Inconvénients

Cependant, le PSNR présente également certains inconvénients :

- **La perception humaine :** Le PSNR ne constitue pas toujours un indicateur fiable de la perception visuelle humaine.

Par exemple, considérons deux images qui ont subi une compression avec des méthodes différentes ayant toutes deux un PSNR de 29,98 dB (Figure 5-2). Cela signifie que ces deux images ont le même niveau global d'erreurs en comparaison avec l'image d'origine.

Pour rappel, ce niveau d'erreurs est mesuré en fonction de leurs erreurs quadratiques moyennes (MSE) par rapport à l'amplitude maximale du signal d'image (MAX), comme indiqué à l'Équation 5-1.



Figure 5-2 : Exemple d'images ayant, toutes deux, un PSNR = 29,98 dB [13]

La valeur PSNR de 29,98 dB indique que les erreurs entre les images reconstruites et l'original sont relativement faibles, car plus la valeur PSNR est élevée, moins il y a d'erreurs.

En effet, bien que ces deux images aient le même PSNR, elles peuvent présenter des différences visuelles dans leur contenu ou leur apparence pour un observateur humain. Par conséquent, malgré une mesure objective identique (PSNR), les qualités perceptives subjectives des deux images peuvent varier en fonction de facteurs tels que le contexte spatial et temporel ou encore la nature des artefacts produits par le traitement appliqué.

Dans certains cas plus extrêmes, des images ayant un faible PSNR peuvent sembler assez similaires à l'original, tandis que celles avec un PSNR élevé peuvent présenter des artefacts visibles. Ce constat va à l'encontre de ce que l'on attendrait du PSNR.

À titre d'illustration, la Figure 5-3 dévoile des images¹ qui affichent une corrélation inverse. En se dirigeant de la droite vers la gauche, le PSNR augmente², indiquant ainsi une moindre fidélité de l'image ; en revanche, la qualité visuelle diminue.

¹ Ces images ont subi un traitement faisant appel à des algorithmes de deep-learning pour améliorer leur qualité et leur résolution en considérant divers paramètres de qualité.

² Les valeurs de PSNR sont données à titre indicatif.

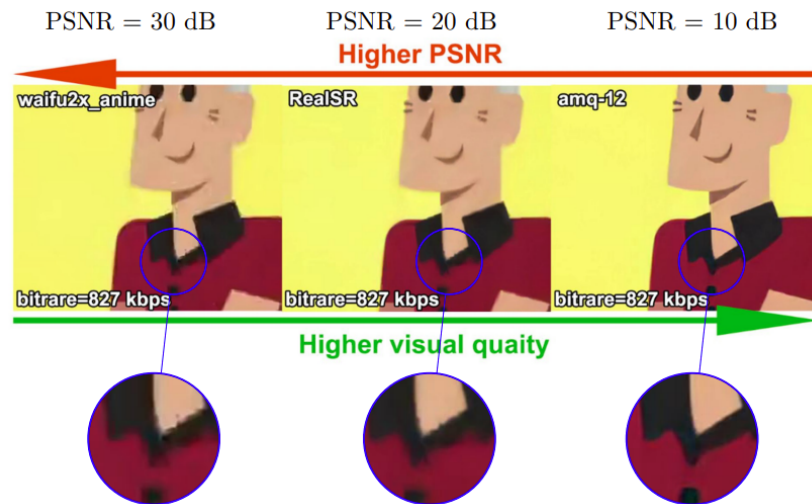


Figure 5-3 : Exemple d'images ayant une corrélation inversée [14]

- **La sensibilité aux erreurs spatiales :** Le PSNR est moins sensible aux erreurs spatiales, telles que les distorsions géométriques, qui peuvent affecter la qualité perçue de l'image.
- **Un manque d'adaptabilité :** Le PSNR est un indicateur unique pour toutes les images et toutes les conditions. Par conséquent, il peut ne pas être adapté pour évaluer les images avec des caractéristiques spécifiques ou dans des scénarios particuliers.

Conclusion

En somme, l'algorithme PSNR est un outil de mesure largement utilisé dans le domaine du traitement numérique des images et des vidéos. Il présente plusieurs avantages. Cependant, bien que le PSNR soit utile dans un certain nombre de cas, il présente également d'importants inconvénients en lien avec la perception humaine. La corrélation imparfaite entre les valeurs de PSNR et la qualité perceptuelle subjective peut être problématique lorsqu'il s'agit d'évaluer précisément la qualité d'une image en tenant compte de la manière dont les observateurs humains perçoivent réellement cette image.

5.3. SSIM : Structural Similarity Index

Présentation de l'algorithme

L'indice de similarité structurelle (Structural Similarity Index, SSIM) est une méthode d'évaluation de la qualité d'une image qui se base sur les perturbations au niveau de la structure, de la luminance, et du contraste entre deux images. Cet algorithme a été développé par Wang Zhou [15], avec pour motivation la prise en compte des erreurs de perceptions humaines dans le processus d'évaluation de la qualité des images.

L'indice SSIM est calculé en utilisant une approche basée sur trois composantes : une mesure de la similitude des *luminances* $l(x, y)$, une mesure de la similitude des *contrastes* $c(x, y)$ et une mesure de la similitude *structurale* $s(x, y)$. Les images comparées sont notées respectivement x et y .

L'indice SSIM pour ces deux images est défini comme suit :

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad \text{Équation 5-3}$$

où α , β et γ sont les paramètres qui déterminent l'importance relative des trois composantes.

Les SSIM locaux

L'algorithme SSIM (Structural Similarity Index) repose sur le concept de fenêtres glissantes. Il s'agit d'évaluer la qualité d'images en divisant celles-ci en petites sous-régions, non chevauchantes et de taille fixe³, plutôt que de les comparer entièrement en une seule fois.

Les valeurs des SSIM locaux correspondent aux SSIM calculés pour chacune des sous-régions déterminées par la fenêtre glissante. En d'autres termes, l'algorithme SSIM compare la qualité des images de manière locale sur des zones spécifiques, définies en fonction de la taille de la fenêtre. Ainsi, les deux images sont séparées en blocs grâce à la fenêtre glissante, et un SSIM local est généré pour chaque paire de blocs correspondants.

³ Le choix de la taille de la fenêtre glissante doit respecter ce critère : la taille de la fenêtre glissante doit être un diviseur commun (entier) de la largeur et de la longueur de l'image.

Cela permet d'obtenir une évaluation plus détaillée et nuancée des variations locales de qualité. On évalue ainsi chaque fenêtre individuellement pour déterminer les différences locales entre les deux images.

La Figure 5-4 montre un exemple de fenêtre glissante qui se déplace dans chaque bloc de l'image à analyser.

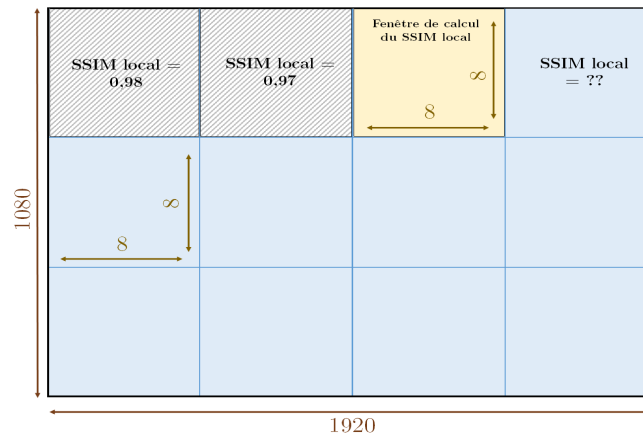


Figure 5-4 : Exemple de fenêtre glissante

La méthode mise en œuvre permet au SSIM d'être moins sensible aux erreurs globales qui ne sont pas facilement perçues par le SVH⁴ et donne une estimation plus précise des différences entre les images. Plus la taille de la fenêtre est grande, plus la valeur du SSIM reflétera fidèlement la perception humaine.

Pour le calcul local du SSIM, nous allons reprendre l'Équation 5-3 et en étudier chaque terme. Les facteurs pondérateurs par défaut étant tous égaux à 1, l'équation devient :

$$\text{SSIM}(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \quad \text{Équation 5-4}$$

En outre, les formules pour les mesures de la similitude des *luminances* $l(x, y)$, de la similitude des *contrastes* $c(x, y)$ et de la similitude *structurale* $s(x, y)$ sont définies comme suit :

⁴ Système Visuel Humain.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad \text{Équation 5-5}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad \text{Équation 5-6}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad \text{Équation 5-7}$$

où μ_x et μ_y représentent les valeurs moyennes des intensités dans les fenêtres glissantes pour les images x et y , σ_x et σ_y représentent l'écart-type dans ces mêmes fenêtres, σ_{xy} représente la covariance entre les intensités d'images de ces deux fenêtres, et enfin, C_1 , C_2 , et C_3 sont des constantes.

En substituant les équations « Équation 5-5 », « Équation 5-6 » et « Équation 5-7 » dans l'équation « Équation 5-4 », on obtient une formule qui regroupe les trois mesures de similitude :

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_3)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad \text{Équation 5-8}$$

Grace à cette formule, nous pouvons calculer les SSIM locaux pour chaque bloc de l'image. Si nous regroupons toutes les valeurs SSIM locales dans une matrice, nous obtenons une carte SSIM (SSIM Map), illustrée à la Figure 5-5. Cette carte représente la qualité de l'image en fonction de la position de la fenêtre glissante.

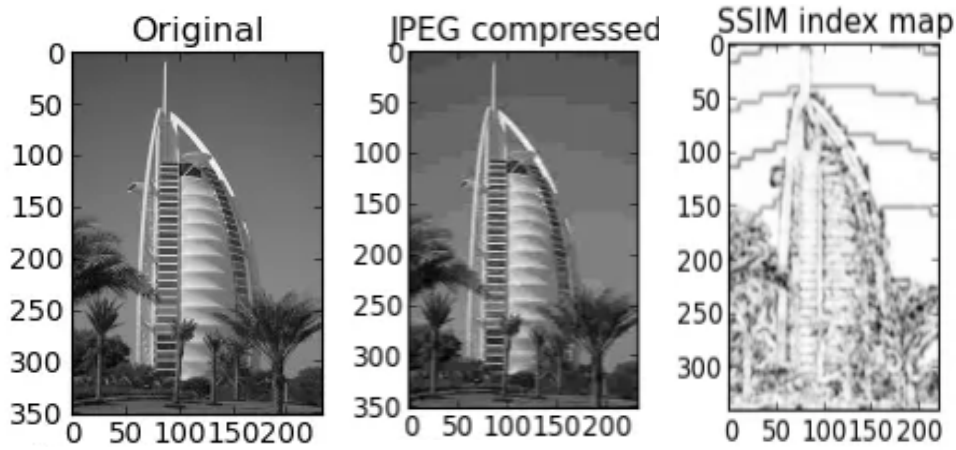


Figure 5-5 : Exemple du SSIM map [16]

Le SSIM global

Afin d'obtenir une mesure globale du SSIM sur toute l'image, il est nécessaire de combiner les résultats issus des calculs locaux en une unique valeur résumant l'évaluation de la qualité d'image perçue. La méthode habituelle consiste à calculer la moyenne arithmétique des valeurs SSIM locales pour chacune des positions de la fenêtre, ce qui donne lieu à une valeur appelée mean SSIM (mSSIM), définie comme suit :

$$\text{mSSIM}(x, y) = \frac{1}{N} \sum_{i=1}^N \text{SSIM}(x_i, y_i) \quad \text{Équation 5-9}$$

où x_i et y_i représentent chacune un bloc correspondant dans les images x et y , et N est le nombre total de blocs obtenus par le balayage complet des deux images avec la fenêtre glissante.

Le mSSIM⁵ fournit ainsi un score global compris entre -1 et 1, permettant d'évaluer objectivement la qualité perçue d'une image par rapport à sa référence. Comme le montre la Figure 5-6 une valeur proche de 1 indique que les deux images sont très similaires, tandis qu'une valeur proche de -1 signifie qu'elles sont très différentes.

⁵ Il est courant de nommer mSSIM comme SSIM et de spécifier seulement "SSIM local" lorsqu'il s'agit de la valeur SSIM d'une fenêtre glissante.

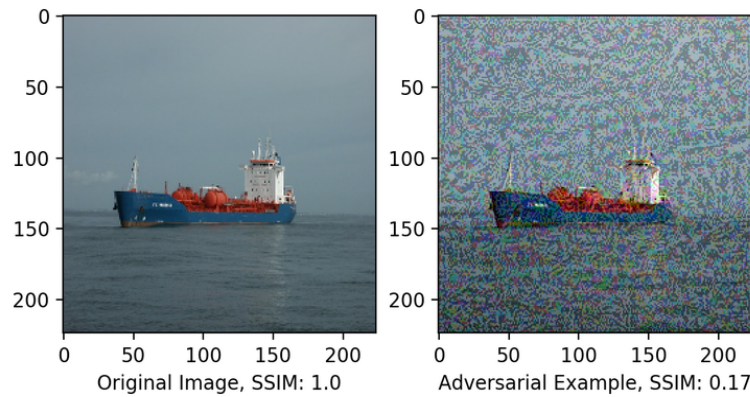


Figure 5-6 : Exemple de la qualité d'images et du mSSIM attendu [17]

Comme nous le verrons dans les avantages, la méthode SSIM présente l'avantage d'être plus corrélée avec la perception humaine que les méthodes traditionnelles basées sur les erreurs moyennes quadratiques (MSE) ou le rapport signal-bruit de pointe (PSNR). De plus, plus la taille de la fenêtre glissante sera grande, plus le résultat du SSIM sera proche de la perception humaine⁶. À l'inverse, une fenêtre plus petite comparera l'image par rapport à ses composantes intrinsèques.

Si nous appliquons l'algorithme SSIM sur des images YUV⁷, tout comme pour le PSNR, seule la composante de luminance Y est en pratique utilisée pour calculer l'indice SSIM, car elle joue un rôle plus significatif dans la perception humaine que les composantes chrominances U et V. Cependant, il est également possible de calculer le SSIM sur les autres composantes.

Avantages

L'algorithme SSIM (Structural Similarity Index Measure) est une méthode de mesure avancée introduite pour prendre en compte la perception humaine et les aspects structurels de l'image. Voici quelques avantages du SSIM:

⁶ Des exemples de ce phénomène seront présentés dans la section 6.

⁷ Pour rappel, Y représente la luminance, et U et V représentent les composantes de chrominance de l'espace colorimétrique.

- **Une meilleure correspondance avec la perception humaine:** Contrairement au PSNR, le SSIM tient plus compte des propriétés visuelles importantes pour la perception humaine. Ainsi, le SSIM est généralement plus cohérent avec l'évaluation subjective de la qualité d'image.
- **Une prise en compte de la structure:** L'algorithme SSIM considère les informations structurelles et textuelles présentes dans l'image, ce qui lui permet de mieux identifier que le PSNR, les erreurs liées aux déformations structurelles des images. Par exemple, il est généralement plus sensible aux distorsions d'intensité, de contraste et de corrélation aux structures perçues.
- **La flexibilité:** Le SSIM est basé sur un modèle modifiable qui peut être adapté à des applications spécifiques ou à des besoins particuliers en matière de mesures de qualité.
- **La détection des différences locales et globales:** Le SSIM est capable de détecter non seulement les différences locales entre les images, mais aussi leurs distorsions globales. Cela permet d'obtenir une évaluation plus précise de la qualité des images en comparant les différences à différentes échelles et en tenant compte des variations locales et globales dans l'image.

Inconvénients

Cependant, le SSIM présente également certaines limites:

- **La complexité de calcul:** Le SSIM nécessite des calculs plus complexes que le PSNR, ce qui peut entraîner une augmentation de la charge de travail et du temps de traitement.
- **La difficulté à détecter le bruit :** Le SSIM peut parfois avoir du mal à détecter le bruit dans une image, ce qui entraîne des valeurs moins précises par rapport au PSNR dans certains cas.
- **La sensibilité aux paramètres :** L'évaluation du SSIM est sensible aux paramètres sélectionnés pour le modèle, ce qui peut nécessiter un réglage minutieux pour obtenir des résultats cohérents et fiables.

Considérons la taille des fenêtres glissantes. Pour rappel, le choix de la taille des fenêtres glissantes ne se fait pas aléatoirement. En effet, il faut trouver les diviseurs communs (entiers) qui divisent la hauteur et la largeur de l'image. Ensuite, la taille de la fenêtre glissante influence grandement les valeurs obtenues.



Figure 5-7 : Exemple d'étalement des valeurs en fonction de la taille de la fenêtre glissante [18]

Dans cet exemple, l'image de gauche (originale) à été comparé à l'image de droite (distordue) de la Figure 5-7 à l'aide de l'algorithme SSIM⁸. Avec une fenêtre glissante de 2 sur 2 pixels, nous obtenons une valeur de SSIM (égale à mSSIM) égale à 0.779892. Néanmoins, pour la même analyse de la même image avec une fenêtre glissante de 10 sur 10 pixels, nous obtenons une valeur de SSIM (égale à mSSIM) égale à 0.832494., ce qui représente une différence de 6.75 % (arrondi à deux décimales près).

Amélioration de l'algorithme SSIM

Le SSIM offre déjà une mesure de qualité d'images PLUS rapprochée de la perception humaine que la méthode précédente. Cependant, il existe des moyens pour améliorer encore cette méthode en la complétant avec des techniques supplémentaires. L'une des pistes pour améliorer la méthode SSIM consiste à utiliser une pondération des pixels basée sur une fonction gaussienne lors du calcul de la fenêtre glissante. En exploitant cette approche, on peut attribuer un poids plus important aux pixels situés au centre de la zone étudiée, et accorder moins d'importance aux pixels situés en périphérie.

La pondération gaussienne est définie par une matrice qui contient les poids correspondants aux positions des pixels dans la fenêtre glissante. Cette matrice peut être adaptée en fonction des besoins spécifiques d'une application, en changeant par exemple le niveau de flou que l'on souhaite appliquer ou l'importance relative accordée aux pixels centraux par rapport à ceux situés en bordure.

Concrètement, une matrice gaussienne est une matrice contenant des valeurs obtenues à partir d'une distribution gaussienne (ou normale) bidimensionnelle. Les coefficients de la matrice représentent les poids attribués aux pixels en fonction de leur distance par rapport au centre de la fenêtre glissante.

⁸ C'est mon implémentation personnelle de l'algorithme SSIM qui a été utilisée pour effectuer les calculs étant donné que c'est le seul qui accepte une taille variable de fenêtre glissante.

Prenons l'exemple d'une matrice gaussienne 3x3 avec une moyenne de 0 et un écart-type de 1 :

$$G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Chaque élément de cette matrice représente le poids accordé aux pixels voisins lors du calcul sur la fenêtre glissante. On peut observer que les pixels du centre ont un poids plus important (4) que ceux situés immédiatement à côté (2) et ceux situés aux coins (1).

Cet exemple est assez simplifié, car dans la pratique, on utilise généralement des matrices plus grandes et avec des valeurs plus précises pour obtenir une pondération plus fine et mieux adaptée à la perception humaine.

Conclusion

En somme, le SSIM apparaît comme une méthode de mesure de la qualité d'images plus avancée et adaptée à la perception humaine que le traditionnel PSNR. Sa prise en compte des aspects structurels, de l'intensité et du contraste des images lui confère une meilleure précision dans l'évaluation de la qualité.

Toutefois, malgré ces avantages notables, il est important de souligner que le SSIM n'est pas exempt d'inconvénients, tels que sa complexité de calcul accrue et sa sensibilité aux paramètres.

Bien qu'il ne soit pas un algorithme universellement optimal pour toutes les applications ou contextes, le SSIM représente généralement un bon compromis entre performance et robustesse en comparaison avec d'autres méthodes telles que le PSNR. Il apporte une évaluation plus cohérente et fiable pour la majorité des situations, sans toutefois prétendre être une solution parfaite à tous les problèmes liés à la mesure de la qualité d'images.

5.4. VMAF : Video Multimethod Assessment Fusion

Présentation de l'algorithme

Le Video Multimethod Assessment Fusion (VMAF) est un outil développé par Netflix [19] pour évaluer la qualité des vidéos en se basant sur un algorithme spécifique. Cet algorithme permet de calculer une métrique de qualité perceptuelle qui cherche à modéliser les préférences humaines en matière de visionnage vidéo.

L'algorithme VMAF combine plusieurs méthodes d'évaluation objective de la qualité vidéo, notamment le Structural Similarity Index (SSIM), le Visual Information Fidelity (VIF) et l'erreur quadratique moyenne pondérée en fonction du contraste, en utilisant un modèle de deep learning. Ces méthodes sont fusionnées pour former une seule mesure qui reflète plus précisément l'expérience visuelle perçue par les utilisateurs lorsqu'ils regardent des vidéos compressées.

VMAF utilise une technique appelée Support Vector Regression (SVR) pour apprendre à estimer la qualité vidéo en se basant sur les opinions des spectateurs. Pour faire simple, l'algorithme prend en compte les spécificités locales de l'image et du mouvement dans la vidéo, ainsi que les préférences générales des personnes qui ont évalué la qualité.

Afin d'apprendre à prédire la qualité, VMAF utilise des données issues d'évaluations subjectives de la qualité vidéo réalisées par un groupe de personnes. Ces individus visionnent différentes séquences vidéo correspondant à diverses conditions expérimentales, telles que différents niveaux de compression ou dégradations artificielles, et donnent leur avis sur la qualité visuelle perçue.

Ces opinions sont ensuite normalisées et servent d'étiquettes pour entraîner l'algorithme à estimer la qualité vidéo de manière similaire à la perception humaine.

L'algorithme VMAF tient également compte des dépendances spatiales et temporelles dans la séquence vidéo pour améliorer la précision de l'évaluation qualitative. Cela signifie que VMAF prend en considération non seulement l'image courante, mais également les images précédentes et suivantes dans la séquence vidéo. En plaçant les images dans un contexte plus large, l'algorithme est mieux en mesure d'appréhender les pertes de qualité induites par des artefacts spécifiques à la compression vidéo ou des erreurs de transmission.

De plus, VMAF considère également la fréquence de rafraîchissement des images dans son modèle d'évaluation. Puisque la perception humaine varie avec le taux de rafraîchissement, il est important d'en tenir compte pour modéliser fidèlement l'expérience visuelle des utilisateurs. En général, un taux de rafraîchissement plus élevé améliore la qualité perçue, car les mouvements sont plus fluides et les artefacts visuels temporaires sont moins perceptibles.

L'algorithme VMAF ajuste donc ses prédictions en fonction de cette information afin d'obtenir une estimation qui soit représentative du ressenti humain face aux différents taux de rafraîchissement rencontrés lors du visionnage des vidéos. Ainsi, les scores VMAF associés aux vidéos testées deviennent plus précis et représentatifs de l'évaluation subjective des participants.

Avantages

- **La précision:** VMAF est conçu en tenant compte des caractéristiques du système visuel humain et s'appuie sur l'apprentissage automatique pour mieux comprendre et prédire la qualité vidéo telle qu'elle serait perçue par un observateur humain moyen. Cela rend VMAF plus précis que les méthodes traditionnelles basées sur des métriques objectives telles que le PSNR (Peak Signal-to-Noise Ratio) et le SSIM (Structural Similarity Index).
- **La fusion multiméthode:** Comme son nom l'indique, VMAF combine plusieurs méthodes d'évaluation pour calculer une note globale de qualité vidéo. Cette approche permet d'améliorer la fiabilité et la robustesse du score obtenu en prenant en compte divers aspects de la qualité vidéo tels que la texture, les artefacts de compression, les erreurs spatiales/temporelles, etc...
- **L'entraînement de modèle sur mesure:** VMAF permet d'entraîner un modèle personnalisé pour une application ou un service vidéo spécifique. Cela permet d'obtenir des scores encore plus précis et plus représentatifs de la qualité vidéo perçue par les utilisateurs finaux.

En résumé, les avantages de VMAF résident dans sa précision améliorée, sa capacité à prendre en compte divers aspects de la qualité vidéo grâce à la fusion multiméthode, sa facilité d'utilisation et son adaptabilité aux divers scénarios d'évaluation vidéo.

Inconvénients

Malgré ses nombreux avantages, l'algorithme VMAF présente également certains inconvénients. Certains des plus notables sont les suivants :

- **Le coût computationnel :** Le calcul du VMAF nécessite une analyse exhaustive de la séquence vidéo et de sa version distordue, ce qui peut exiger d'importantes ressources matérielles et logicielles.
- **La sensibilité aux changements d'échelle :** Bien que le VMAF prenne en compte les variations de résolution et de qualité dans son calcul, il peut ne pas être aussi sensible aux variations d'échelle que d'autres métriques. Cela signifie qu'il peut ne pas répondre correctement aux différences de qualité lorsque la taille des images varie considérablement.
- **Une non-applicabilité à toutes les formes de contenu vidéo:** Le succès du VMAF dépend en grande partie de la méthodologie d'apprentissage utilisée pour entraîner le modèle ; il se peut donc que l'algorithme ne fonctionne pas aussi bien pour des formes de contenu vidéo qui n'étaient pas présentes dans les données d'apprentissage. Par exemple, l'efficacité du VMAF pourrait être limitée pour évaluer des contenus tels que les vidéos de réalité virtuelle ou les vidéos générées par ordinateur.

Conclusion

En somme, l'algorithme VMAF offre une métrique d'évaluation de la qualité vidéo extrêmement précise et en adéquation avec la perception humaine. Néanmoins, il est important de souligner que son efficacité dépend étroitement du modèle utilisé et des conditions d'apprentissage.

Cela signifie que ses résultats hautement précis sont limités aux situations similaires à celles rencontrées lors de l'entraînement du modèle. Ainsi, il est crucial de considérer cet aspect en l'employant dans divers contextes et applications.

5.5. Comparaison synthétique des algorithmes

Le *Tableau 1 : Efficacité des algorithmes PSNR, SSIM et VMAF afin de détecter différents défauts et artefacts* synthétise l'efficacité des algorithmes PSNR, SSIM et VMAF⁹ pour détecter différents défauts et artefacts¹⁰ dans les images et vidéos compressées. La première colonne liste les types de défauts courants, tandis que les trois autres colonnes indiquent l'efficacité de chaque algorithme (Pas Trop, Moyen, Bon ou Très Bon) pour détecter ces défauts.

Notez que certains artefacts comme la distorsion géométrique et le vignettage ne sont pas pris en compte par ces mesures d'évaluation car ils ne sont généralement pas causés par la compression et donc ces artefacts seraient déjà présents dans l'image/vidéo de référence.

⁹ Dans un cas favorable où le modèle VMAF est entraîné pour ce défaut précis.

¹⁰ Ces défauts et artefacts sont repris dans la section 3.

Tableau 1 : Efficacité des algorithmes PSNR, SSIM et VMAF afin de détecter différents défauts et artefacts

Défauts/Artefacts	PSNR	SSIM	VMAF
Blocage	Moyen	Bon	Très Bon
Flou	Moyen	Bon	Très Bon
Contours irréguliers	Pas Trop	Moyen	Très Bon
Ringing	Pas Trop	Moyen	Très Bon
Propagation d'erreur	Moyen	Bon	Très Bon
Le bruit numérique	Moyen	Bon	Très Bon
Distorsion géométrique	Non	Non	Non
Vignettage	Non	Non	Non
Effet fantôme	Pas Trop	Pas Trop	Très Bon
Moirage	Pas Trop	Moyen	Très Bon

Conclusion

En somme, le PSNR et le SSIM présentent tous deux des avantages et des inconvénients. Le PSNR est facile à calculer et à comprendre, mais il ne tient pas compte de la perception humaine ni des aspects structurels de l'image. Le SSIM se concentre sur ces aspects, offrant une meilleure représentation de la qualité perçue. Cependant, il est plus complexe à calculer et nécessite un réglage approprié des paramètres.

Dans les domaines où la perception humaine est importante pour évaluer la qualité d'images ou vidéos (par exemple, les applications multimédias), le SSIM est souvent préférable au PSNR. Dans les scénarios où une mesure simplifiée suffit ou si les ressources informatiques sont limitées, le PSNR peut être utilisé avec succès.

6. Analyse des résultats

6.1. Comparaison des algorithmes sur des défauts classiques

Voici l'image qui illustre le bruit de la section 3. Toutefois, il y a en plus une autre image présentant un défaut de blocage, visible sur la Figure 6-1. Si la qualité des images devait être évaluée subjectivement, le classement serait le suivant : celle de référence est la plus agréable à regarder (l'image du dessus), suivie par celle de

gauche (l'image avec du bruit) en deuxième position et en dernière celle de droite (l'image avec le défaut de blocage), qui est nettement moins plaisante à regarder.

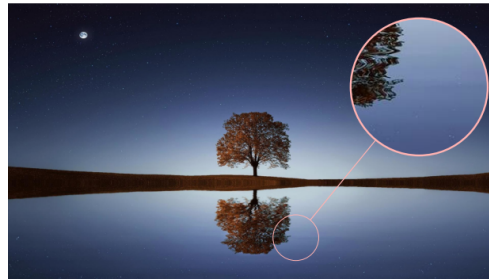


Image de référence



Image avec du bruit

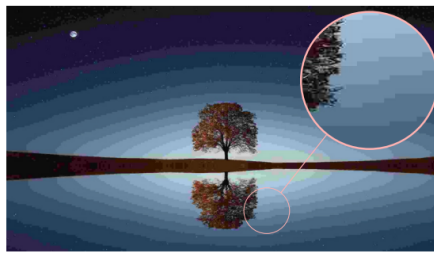


Image avec un défaut de blocage

Figure 6-1 : Voici trois images : la première (en haut) est l'image de référence. Tandis que les deux du bas présentent des artefacts de compression. L'image, en bas à gauche, montre un niveau de bruit élevé ; et celle, en bas à droite, affiche un défaut de blocage.

Les différents algorithmes de ce programme ont été appliqués et nous allons vérifier si cet aspect proche de la perception humaine est respecté. Les résultats sont affichés dans le Tableau 2. Les images ont une résolution classique de 1920x1080 pixels.

Tableau 2 : Tableau des valeurs de SSIM et PSNR pour la comparaison des deux images présentant des artefacts de compression.

Algorithme	Valeur (Image bruit)	Valeur (Image blocage)
$SSIM_{window=4}$	0,432186	0,790896
$SSIM_{window=8}$	0,43299	0,761506
$SSIM_{window=24}$	0,484686	0,59764
$SSIM_{window=60}$	0,588273	0,488254
$SSIM_{window=120}$	0,719698	0,372772
PSNR	27.681002 dB	30.470774 dB

Premièrement, en ce qui concerne mon implémentation du SSIM, on constate que lorsque l'on utilise des petites fenêtres, la valeur du SSIM semble également indiquer

une amélioration de l'image. Néanmoins, lorsque la taille des fenêtres augmente, la valeur du SSIM chute drastiquement et se rapproche davantage de l'impression humaine.

Nous commençons à présent à percevoir une tendance plus régulière concernant l'augmentation de la taille des fenêtres glissantes. Effectivement, accroître la taille des fenêtres ne signifie pas systématiquement une augmentation de la valeur du SSIM.

En réalité, augmenter la taille des fenêtres permet d'obtenir une mesure plus proche de la perception humaine. Les fenêtres les plus grandes sont plus proches du classement subjectif. Néanmoins, il faut garder à l'esprit que les deux images inférieures ont réellement été dégradées et comportent donc bel et bien des défauts; cela peut être constaté en zoomant sur ces images. Grâce aux fenêtres plus petites, nous sommes en mesure de détecter ces défauts intrinsèques à l'image; même si l'œil humain ne les avait pas nécessairement perçus comme visuellement impactant.

6.2. La preuve que le PSNR n'est pas toujours fiable et la vérification de la relation avec la taille des fenêtres glissantes

Examinons un autre exemple pour illustrer les limites du PSNR et valider la relation des fenêtres glissantes avec la perception humaine. La Figure 6-2 présente à gauche l'image originale et à droite l'image déformée, dont la résolution est de 1500x1000 pixels. Cette dernière présente un défaut de flou.

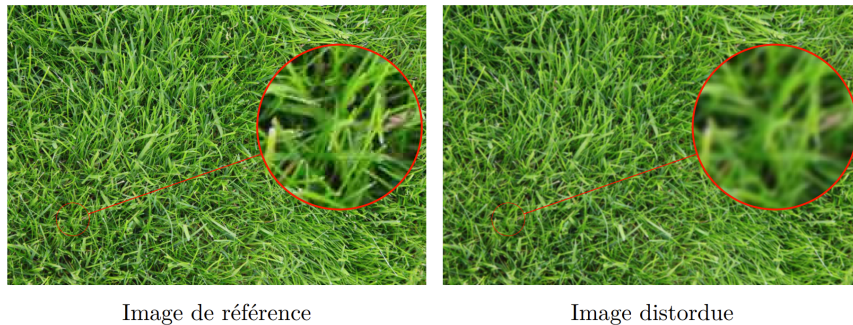


Figure 6-2 : Exemple de flou [4]

Cependant, à première vue, aucune de ces deux images ne semble significativement moins "belle" comparée à l'autre. Néanmoins, l'utilisation d'une loupe permet de constater que l'image de droite est intrinsèquement de moindre qualité.

Pourtant, selon la perception humaine, on pourrait dire qu'elles sont, pour ainsi dire, presque équivalentes, même si celle de droite est légèrement plus floue.

Ce qui est intéressant ici, c'est qu'en appliquant une analyse des algorithmes sur ces images, nous obtenons les résultats présentés dans le Tableau 3.

Tableau 3 : Tableau des valeurs de SSIM et PSNR avec différentes tailles de fenêtres

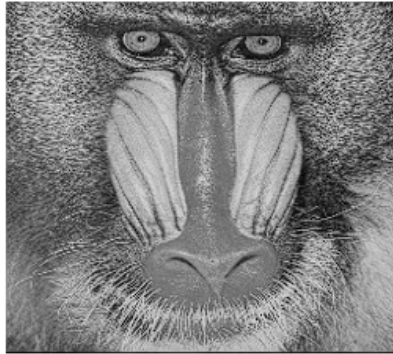
Mesure	Valeur
$SSIM_{windowssize=2}$	0,695134
$SSIM_{windowssize=5}$	0,682084
$SSIM_{windowssize=10}$	0,784714
$SSIM_{windowssize=25}$	0,85751
$SSIM_{windowssize=100}$	0.88515
PSNR	23.850395 dB

Il apparaît clairement que la taille des fenêtres glissantes influence l'adéquation de l'analyse avec le système visuel humain. En effet, dans cet exemple plus précisément, plus la taille de la fenêtre augmente, meilleure sera la valeur du SSIM. Or cette image contient bien des défauts. Ces défauts, nous les avons identifiés comme mineurs du point de vue humain, sachant que les deux images sont globalement similaires. Les valeurs du SSIM obtenues avec une fenêtre de 100x100 pixels nous confirme cela.

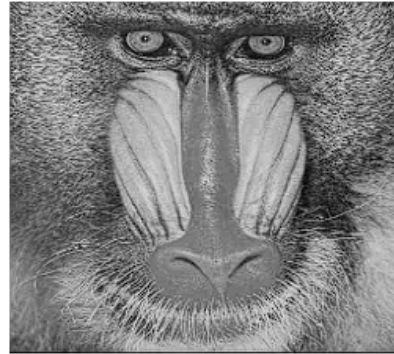
Toutefois, le PSNR n'est pas pertinent pour cet exemple. De fait, dans cet exemple-ci, il indique une dégradation plus importante de l'image ; comparée à celle de la Figure 6-1 avec l'image présentant un défaut de blocage. Ce qui ne correspond pas à la perception humaine.

6.3. Autre étude montrant les mêmes conclusions

Voici une étude qui obtient les mêmes conclusions que ce que le programme exporte comme conclusion.



(a1) image Mindrill



(a2) Mindrill compressée en JPEG



(b1) image Lake



(b2) image Lake compressée par JPEG



(c1) image Tiffany



(c2) Tiffany compressée par JPEG

Figure 3 : Couples d'images compressées par JPEG [20]

Couple d'image	Type de distortion	PSNR	ZQ ¹¹	SSIM	Evaluation subjective
(a1)-(a2)	Compression	25.9906	2.0770	0.8646	1
(b1)-(b2)	Compression	25.8677	26.225	0.7773	2
(c1)-(c2)	Compression	25.8854	40.114	0.7112	3

De ce tableau nous pouvons remarquer que le SSIM correspond bien à la perception visuelle et le ZQ a évalué efficacement les images compressées en JPEG, tandis que le PSNR demeure inchangé, ne reflétant pas correctement les différences de qualité visuelle entre les paires d'images.

6.4. Impact de la taille des fenêtres glissantes sur la valeur du SSIM

Dans cette section, nous analyserons l'impact de la taille des fenêtres glissantes¹² sur la valeur moyenne globale du SSIM¹³. En effet, dans la section 5.3, il a été mentionné comme point négatif, concernant le SSIM, que sa valeur varie considérablement en fonction de la taille des fenêtres. Ce constat est confirmé par le graphique présenté dans la Figure 6-4 : Voici le SSIM moyen global en fonction de la taille de la fenêtre glissante.

La conclusion tirée lors de l'analyse approfondie des différents algorithmes est la suivante : plus la taille de la fenêtre est grande, plus on se rapproche de la perception humaine.

Cela implique qu'en appliquant cette logique au graphique de la Figure 6-4, on peut déduire que le flux d'images est agréable pour un observateur humain.

Les valeurs obtenues avec une taille de fenêtre de 2x2 ou 3x3 sont trop petites pour être véritablement représentatives de la vidéo.

¹¹ ZQ signifie « Zipf Quality » qui est une métrique qui mesure la qualité des images JPEG.

¹² Se référer à la section 5.3.

¹³ Cela signifie que j'ai pris le SSIM global de chaque image de la vidéo et j'en ai fait une moyenne.

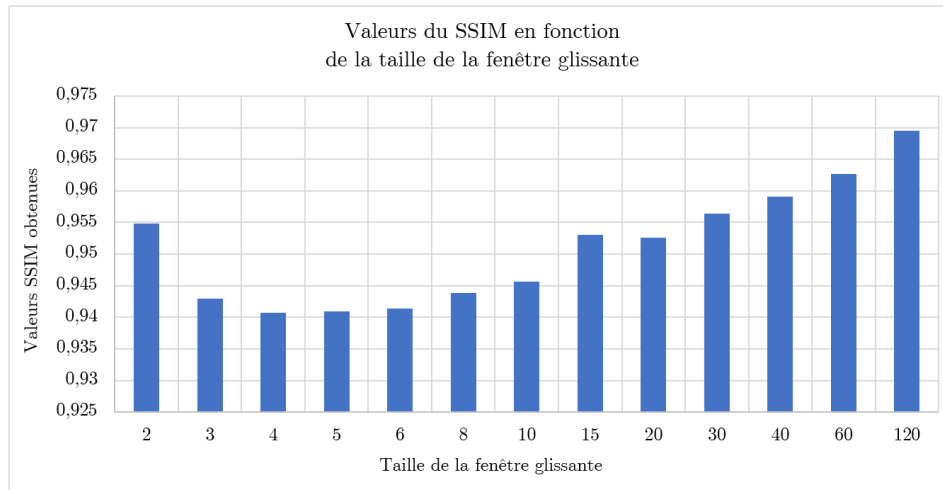


Figure 6-4 : Voici le SSIM moyen global en fonction de la taille de la fenêtre glissante.

6.5. Analyse d'une séquence vidéo

Analysons, maintenant, une séquence vidéo de 30 images 1080p (60 Hz). Pour ce faire, un SSIM¹⁴ est calculé pour chaque image. Les résultats des calculs sont présentés dans la Figure 6-5. Une précision est à donner : la fenêtre glissante avait une taille de 8x8 pixels.

¹⁴ Pour l'algorithme SSIM, c'est ma propre implémentation qui a été utilisée.

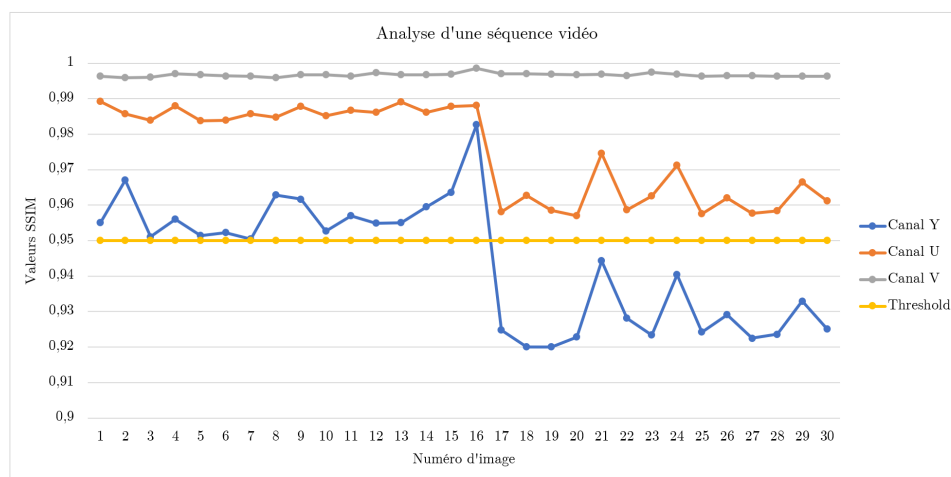


Figure 6-5 : Voici le graphique de la séquence vidéo qui a été analysée.

Le graphique obtenu est fort instructif. En effet, il reprend beaucoup d'éléments abordés dans cet article. Si nous examinons attentivement celui-ci, plusieurs questions se posent.

1. Pourquoi observe-t-on une chute importante après l'image 16 pour les courbes Y et U ?
2. Pourquoi y a-t-il un pic avant la chute de la courbe Y ?
3. Pourquoi le canal Y est le moins bon ?
4. Pourquoi le canal U suit le Y mais en ayant de meilleures valeurs ?
5. Pourquoi le canal V est très élevé et très constant ?

Nous allons maintenant expliquer chacun de ces points. Il est important de saisir ces différentes causes dans le but d'avoir une meilleure compréhension du processus de compression des flux vidéo.

1. Pourquoi observe-t-on une chute importante après l'image 16 ?

Nous pouvons constater une nette diminution de la valeur du SSIM à partir de la 17ème image. Cette chute peut être attribuée à un défi croissant pour le codec¹⁵ lorsqu'il doit compresser l'image ; en raison de changements dans la complexité des détails présents entre les deux sous-séquences d'images. Les différentes sous-séquences sont illustrées à la Figure 6-6. En effet, en comparant les images avant et après cette baisse, on remarque que la seconde sous-séquence est plus difficile à compresser sans perte en raison de l'apparition d'éléments complexes tels que l'herbe, qui présente des détails fins et riches en informations visuelles.

¹⁵ Le codec est l'algorithme qui compresse la vidéo.

À l'inverse, la première sous-séquence est plus aisée à compresser car elle possède moins de ces détails fins et se prête donc mieux à une réduction sans pertes majeures. Par conséquent, le codec parvient à maintenir une fidélité plus élevée par rapport à la séquence d'origine pour les premières images. Alors que, pour les suivantes, avec davantage de complexité visuelle, la qualité se dégrade encore plus lors de la compression; cela entraîne une diminution notable du score SSIM.

Partie initiale de
la vidéo (<16 images)



Partie suivante de
la vidéo (>=16 images)



Figure 6-6 : Voici les images des séquences vidéos.

2. Pourquoi y a-t-il un pic avant la chute ?

Nous pouvons observer un pic juste avant la chute. Cela vient du fait que le codec a moins transformé l'image numéro 16. En effet, l'image 16 est la première où il y a un changement de scène, c'est-à-dire la première image d'une nouvelle série d'images. Les codecs vidéo tirent parti des redondances d'informations visuelles entre les trames de la vidéo. Toutefois, étant donné qu'il s'agit d'une nouvelle scène, le codec n'a pas d'autres choix que de stocker davantage d'informations relatives à l'image de la nouvelle scène.

Nous constatons d'ailleurs que par la suite, le codec a recommencé à optimiser la compression grâce à la répétition des informations visuelles entre les trames. Cela étant dû au fait que les images qui suivent, ont des valeurs plus basses de SSIM ce qui traduit une dégradation de celles-ci.

3. Pourquoi le canal Y est le moins bon ?

Le canal Y présente des valeurs inférieures, ce qui s'explique par plusieurs raisons : premièrement, il représente la composante de luminance, qui occupe une part importante de l'image. Deuxièmement, l'image est en UYVY subsampling 4:2:2. Ce format d'échantillonnage des couleurs signifie qu'il y a deux fois plus d'informations pour le canal Y par rapport aux canaux U et V.

Étant donné que le canal Y contient davantage d'informations et que le codec doit compresser ces informations pour réduire la taille du fichier vidéo, il sera contraint de compresser plus fortement cette composante. Cette compression conduit inévitablement à une plus grande dégradation de la qualité du canal Y, par rapport aux canaux U et V. En conséquence, la fidélité à l'image de référence pour cette composante est moins bonne, ce qui se traduit par un score SSIM plus faible pour le canal Y.

4. Pourquoi le canal U suit le Y mais en ayant de meilleures valeurs ?

Le canal U suit le Y tout en ayant de meilleures valeurs. Cela s'explique par le fait que le canal U représente la composante chrominance (bleu / jaune). Dans cette séquence d'images, la composante U est plus sollicitée que la composante V en raison de la nature des couleurs présentes dans les images. Par conséquent, davantage d'informations sont conservées pour cette composante lors du processus de compression.

Toutefois, il est important de noter que la qualité du canal U reste inférieure à celle du canal V, car il subit tout de même une certaine dégradation due à la compression avec perte. Néanmoins, celle-ci est moins importante que celle subie par le canal Y (luminance). Cela étant dû au fait que notre vision est moins sensible aux variations de chrominance qu'à celles de luminance.

5. Pourquoi le canal V est très élevé et très constant ?

Le canal V est très élevé, et en examinant attentivement les images, on peut constater qu'il n'y a pas beaucoup de mélanges entre les couleurs rouge et bleu dans la scène. Cela indique que la composante V est peu sollicitée et que ses valeurs restent relativement stables tout au long de la séquence vidéo. Cette stabilité facilite grandement le travail du codec pour compresser et décompresser cette composante, car il doit gérer un nombre réduit d'informations variables.

En considérant que les informations de la composante V sont constantes sur une grande partie des images, le codec peut les compresser efficacement sans perdre amplement des informations lors de la décompression. Par conséquent, cette meilleure gestion des informations permet une plus grande fidélité à l'image de référence pour le canal V par rapport aux canaux U et Y, ce qui se traduit par un score SSIM plus élevé pour cette composante.

7. Conclusion

Malgré la complexité d'analyser un flux vidéo en temps réel, il est possible d'obtenir des résultats exploitables et en adéquation avec les attentes des utilisateurs pour des cas d'analyses spécifiques.

Bien que le PSNR soit meilleur que le SSIM dans certains cas exceptionnels, le SSIM permet une analyse plus approfondie [21], [22]. Il peut s'adapter à diverses situations, que ce soit pour analyser l'image dans sa totalité, des parties spécifiques ou même de très petits défauts. Le SSIM peut aussi être paramétré en fonction des besoins.

En outre, mon programme offre la possibilité d'exporter diverses données calculées ainsi que de sauver les images problématiques, pour une analyse plus poussée avec d'autres logiciels de traitement, détails non abordés dans cet article.

8. Sources

- [1] T. Joye, « Étude et implémentation d'algorithmes de quantification basés sur des modèles psycho-visuels en vue de valider la qualité en temps réel de codecs vidéo », 2023.
- [2] A. Robinson, « The difference between JPEG artifacts and image noise », Welcome to the Topaz Labs Learning Center. Consulté le: 26 mai 2023. [En ligne]. Disponible sur: <https://www.topazlabs.com/learn/the-difference-between-jpeg-artifacts-and-image-noise>
- [3] « Comment faire une belle photo noir et blanc ? », Quora. Consulté le: 8 février 2024. [En ligne]. Disponible sur: <https://fr.quora.com/Comment-faire-une-belle-photo-noir-et-blanc>
- [4] « 10 Best Types of Drought-Tolerant Lawn Grass », The Spruce. Consulté le: 8 février 2024. [En ligne]. Disponible sur: <https://www.thespruce.com/dought-tolerant-lawn-grass-2153119>
- [5] V. Lillo, « St. Marks National Wildlife Refuge in Florida », Wander With Wonder. Consulté le: 8 février 2024. [En ligne]. Disponible sur: <https://www.wanderwithwonder.com/st-marks-national-wildlife-refuge-florida/>
- [6] Futura, « Crépuscule : l'arbre au miroir », Futura. Consulté le: 8 février 2024. [En ligne]. Disponible sur: <https://www.futura-sciences.com/fonds-ecran/arbre-crepuscule-arbre-miroir-3046/>
- [7] PengoHome, « Qu'est-ce que l'espace couleur RGB ? Qu'est ce que le sous-échantillonnage de la chrominance ? 4:4:4 vs 4:2:2 vs 4:2:0 », PengoHome.com. Consulté le: 11 juin 2023. [En ligne]. Disponible sur: https://www.pengohome.com/FR/Learn_Detail.asp?LiD=3A96AC2BB3A41AD22A254205D8DCB795
- [8] « 8 bits / 10 bits, c'est quoi la profondeur des couleurs ? », L'Atelier du câble. Consulté le: 15 février 2024. [En ligne]. Disponible sur: <https://www.latelierducable.com/tv-tele-viseur/8-bits-10-bits-cest-quoi-la-profondeur-des-couleurs/>

- [9] « YUV 420, YCbCr 422, RGB 444, c'est quoi le chroma subsampling 2? », L'Atelier du câble. Consulté le: 15 février 2024. [En ligne]. Disponible sur: <https://www.latelierducable.com/tv-teliviseur/yuv-420-ycbcr-422-rgb-444-cest-quoi-le-chroma-sub-sampling/>
- [10] « YUV 420, YCbCr 422, RGB 444, c'est quoi le chroma subsampling 2? », L'Atelier du câble. Consulté le: 11 juin 2023. [En ligne]. Disponible sur: <https://www.latelierducable.com/tv-teliviseur/yuv-420-ycbcr-422-rgb-444-cest-quoi-le-chroma-sub-sampling/>
- [11] Q. Huynh-Thu et M. Ghanbari, « The accuracy of PSNR in predicting video quality for different video scenes and frame rates », *Telecommun Syst*, vol. 49, n° 1, p. 35-48, janv. 2012, doi: 10.1007/s11235-010-9351-x.
- [12] « Figure 5: Illustration of the PSNR measure », ResearchGate. Consulté le: 8 février 2024. [En ligne]. Disponible sur: https://www.researchgate.net/figure/Illustration-of-the-PSNR-measure_fig6_268807675
- [13] « Secret decrypting procedure in (3, 5)-SSSIS: (a) Result of three... », ResearchGate. Consulté le: 8 février 2024. [En ligne]. Disponible sur: https://www.researchgate.net/figure/Secret-decrypting-procedure-in-3-5-SSSIS-a-Result-of-three-shares-b-Result-of_fig2_368716099
- [14] « PSNR and SSIM: application areas and criticism ». Consulté le: 8 février 2024. [En ligne]. Disponible sur: <https://videoprocessing.ai/metrics/ways-of-cheating-on-popular-objective-metrics.html>
- [15] Z. Wang, A. C. Bovik, H. R. Sheikh, et E. P. Simoncelli, « Image quality assessment: from error visibility to structural similarity », *IEEE Transactions on Image Processing*, vol. 13, n° 4, Art. n° 4, avr. 2004, doi: 10.1109/TIP.2003.819861.
- [16] « Algorithm assesses image and video quality », Vision Systems Design. Consulté le: 8 février 2024. [En ligne]. Disponible sur: <https://www.vision-systems.com/boards-software/article/16736698/algorithm-assesses-image-and-video-quality>
- [17] « Fig. 1. Structural Similarity Index of an Adversarial Example compared... », ResearchGate. Consulté le: 8 février 2024. [En ligne]. Disponible sur: https://www.researchgate.net/figure/Structural-Similarity-Index-of-an-Adversarial-Example-compared-to-the-original-image_fig1_343850757
- [18] « Qatar Airways c. Airbus : La diffusion de l'information suffit-elle à dominer un rapport de force ? », Ecole de Guerre Economique. Consulté le: 15 février 2024. [En ligne]. Disponible sur: <https://www.ege.fr/infoguerre/qatar-airways-c-airbus-la-diffusion-de-linformation-suffit-elle-dominer-un-rapport-de-force>
- [19] S. Deng, J. Han, et Y. Xu, « VMAF Based Rate-Distortion Optimization for Video Coding », in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, sept. 2020, p. 1-6. doi: 10.1109/MMSP48831.2020.9287114.
- [20] T. Delleji et M. S. Bouhlel, « Evaluation de deux métriques pour la mesure de la qualité des images compressées par la norme JPEG ».
- [21] U. Sara, M. Akter, et M. S. Uddin, « Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study », *Journal of Computer and Communications*, vol. 7, n° 3, Art. n° 3, mars 2019, doi: 10.4236/jcc.2019.73002.
- [22] A. Horé et D. Ziou, « Image Quality Metrics: PSNR vs. SSIM », in *2010 20th International Conference on Pattern Recognition*, août 2010, p. 2366-2369. doi: 10.1109/ICPR.2010.579.