Compression de contenus visuels pour transmission mobile sur réseaux de très bas débit

Sebastien Hamis

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2020IPPAS020

Thèse de doctorat

Compression de contenus visuels pour transmission mobile sur réseaux de très bas débit

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 : École Doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Signal, Images, Automatique et Robotique

Thèse présentée et soutenue à Évry-Courcouronnes, le 6 novembre 2020, par

Sébastien Hamis

Composition du Jury :

Jenny Benois-Pineau Professeur, Université Bordeaux 1 (– LaBRI)	Présidente
Amel Benazza-Benyahia Professeur, Sup'Com Tunis, (– COSIM Lab)	Rapporteur
Azeddine Beghdadi Professeur, Université Sorbonne Paris Nord (– Institut Galilée)	Rapporteur
Didier Nicholson Docteur, Ektacom	Examinateur
Titus Zaharia Professeur, Télécom SudParis (– SAMOVAR)	Directeur de thèse
Oliver Rousseau Be-Bound	Co-Encadrant de thèse

INSTITUT POLYTECHNIQUE DE PARIS

Titre : Compression de contenus visuels pour transmission mobile sur réseaux de très bas débit

Mots clés : Contenus visuels à très bas débit, applications mobiles, apprentissage profond, compression d'image, réseaux antagonistes génératifs

Résumé : Le domaine de la compression de contenus visuels (image, vidéo, éléments graphiques 2D/3D) a connu, depuis maintenant plus de vingt ans, un essor considérable avec l'émergence notamment, au fil des années, de nombreuses normes internationales comme JPEG, JPEG2000 pour les images fixes ou les différentes versions de standards MPEG-1/2/4 pour les données vidéos et graphiques.

L'apparition des *smartphones* et l'explosion des applications qui leur sont dédiées a également bénéficié de ces avancées, l'image étant aujourd'hui omniprésente dans un contexte de mobilité/itinérance. Néanmoins, cela nécessite toujours des réseaux fiables et disponibles, offrant un débit suffisant pour la transmission effective de ces données visuelles qui sont intrinsèquement gourmandes en bande passante.

Si aujourd'hui les pays développés bénéficient de réseaux mobiles (3G, 4G…) hautement performantes, cela n'est pas le cas d'un certain nombre de régions du monde, en particulier dans les pays émergents, où les communications s'appuient encore sur des réseaux 2G SMS. Transmettre des contenus visuels dans un tel contexte devient un défi ambitieux, qui nécessite la mise en œuvre de nouveaux algorithmes de compression. Le défi à relever consiste à assurer une transmission des images sur une bande passante correspondant à un ensemble relativement réduit (10 à 20) de SMS (140 octets par SMS).

Pour répondre à ces contraintes, de multiples pistes de développement ont été envisagées. Après un état de l'art des techniques de compression traditionnelles et de leurs améliorations futures, nous avons finalement orienté nos travaux vers des méthodes de *deep learning*, visant à réaliser des post-traitements pour améliorer la qualité des contenus compressés.

Nos contributions s'articulent autour de la création d'un nouveau schéma de compression, incluant les codecs existants ainsi qu'un panel de post-traitements permettant une meilleure exploitation des contenus fortement compressés. Ces briques sont des réseaux de neurones profonds dédiés, qui réalisent des opérations de super-résolution et/ou de réduction d'artéfacts de compression, spécifiquement entraînés pour répondre aux objectifs ciblés. Ces opérations interviennent du côté du décodeur et peuvent être interprétées comme des algorithmes de reconstruction d'images à partir de versions fortement compressées. Cette approche présente l'avantage de pouvoir s'appuyer sur des codecs existants, particulièrement légers et peu coûteux en ressources. Dans nos travaux, nous avons retenu le format BPG, qui fait état de l'art dans le domaine, mais d'autres schémas de compression peuvent être également considérés.

Concernant le type de réseaux de neurones, nos recherches nous ont conduits vers les réseaux antagonistes génératifs (Generative Adversarials Nets–GAN), qui s'avèrent particulièrement adaptés pour des objectifs de reconstruction à partir de données incomplètes. Plus précisément, les deux architectures retenues et adaptées à nos objectifs sont les réseaux SRGAN et ESRGAN. L'impact des différents éléments et paramètres impliqués, comme notamment les facteurs de super-résolution utilisés et les fonctions de pertes, sont analysés en détail.

Enfin, une dernière contribution concerne l'évaluation expérimentale. Après avoir montré les limitations des métriques objectives, qui peinent à prendre en compte la qualité visuelle de l'image, nous avons mis en place un protocole d'évaluation subjective. Les résultats obtenus en termes de scores MOS (Mean Opinion Score) démontrent pleinement la pertinence des approches de reconstruction proposées.

Enfin, nous analysons une ouverture de nos travaux à des cas d'utilisation différents, de nature plus grand public. C'est notamment le cas pour le traitement de contenus de grande résolution plus ou moins compressés et même pour l'amélioration de la qualité de vidéos.

Institut Polytechnique de Paris
91120 Palaiseau, France

INSTITUT POLYTECHNIQUE DE PARIS

Title: Visual content compression for very low bitrate mobile transmission

Keywords: very low bitrate visual content, mobile applications, deep learning, image compression, generative adversarial networks

Abstract: The field of visual content compression (image, video, 2D/3D graphics elements) has known spectacular achievements for more than twenty years, with the emergence numerous international standards such as JPEG, JPEG2000 for still image compression, or MPEG-1/2/4 for video and 3D graphics content coding.

The apparition of smartphones and of their related applications have also benefited from these advances, the image being today ubiquitous in a context of mobility. Nevertheless, image transmission requires reliable and available networks, since such visual data that are inherently bandwidth-intensive. While developed countries benefit today from high-performance mobile networks (3G, 4G...), this is not the case in a certain number of regions of the world, particularly in emerging countries, where communications still rely on 2G SMS networks. Transmitting visual content in such a context becomes a highly ambitious challenge, requiring the elaboration of new very low bitrate compression algorithms. The challenge is to ensure images transmission over a narrow bandwidth corresponding to a relatively small set (10 to 20) of SMS (140 bytes per SMS).

To meet such constraints, multiple axes of development have been considered. After a state-of-the-art of traditional image compression techniques, we have oriented our research towards deep learning methods, aiming to achieve post-treatments over strongly compressed data in order to improve the quality of the decoded content.

Our contributions are structured around the creation of a new compression scheme, including existing codecs and a panel of post-processing bricks aiming at enhancing highly compressed content. Such bricks represent dedicated deep neural networks, which perform super-resolution and/or compression artifact reduction operations, specifically trained to meet the targeted objectives. These operations are carried out on the decoder side and can be interpreted as image reconstruction algorithms from heavily compressed versions. This approach offers the advantage of being able to rely on existing codecs, which are particularly light and resource-efficient. In our work, we have retained the BPG format, which represents the state of the art in the field, but other compression schemes can also be considered.

Regarding the type of neural networks, we have adopted Generative Adversarials Nets-GAN, which are particularly well-suited for objectives of reconstruction from incomplete data. Specifically, the two architectures retained and adapted to our objectives are the SRGAN and ESRGAN networks. The impact of the various elements and parameters involved, such as the super-resolution factors and the loss functions, are analyzed in detail.

A final contribution concerns experimental evaluation performed. After showing the limitations of objective metrics, which fail to take into account the visual quality of the image, we have put in place a subjective evaluation protocol. The results obtained in terms of MOS (Mean Opinion Score) fully demonstrate the relevance of the proposed reconstruction approaches.

Finally, we open our work to different use cases, of a more general nature. This is particularly the case for high-resolution image processing and for video compression.

Institut Polytechnique de Paris
91120 Palaiseau, France

Remerciements

Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Je voudrais tout d'abord remercier mon directeur de thèse, Titus Zaharia, pour toute son aide. Je suis plus que ravi d'avoir travaillé en sa compagnie car outre son appui scientifique, il a toujours été là pour me soutenir, me conseiller et surtout m'accompagner au cours de l'élaboration de cette thèse.

Je remercie également Olivier Rousseau, mon co-encadrant industriel, pour m'avoir accompagné tout au long de cette thèse, ainsi que pour l'accueil chaleureux qui m'a été fait à chaque fois que je suis allé travailler chez Be-Bound.

Je tiens à remercier particulièrement les Professeurs Amel Benazza-Benyahia et Azeddine Beghdadi m'ont fait l'honneur d'accepter la lourde tâche de rapporter ma thèse. Leurs rapports et leurs remarques m'ont permis d'avoir un regard nouveau sur plusieurs aspects de mes travaux et m'ont amené à me poser des questions pertinentes quant aux futurs développements de ces recherches.

Je tiens à remercier Professeur Jenny Benois-Pineau et Monsieur Didier Nicholson pour l'honneur qu'ils me font de s'intéresser à ces travaux et de participer à mon jury de thèse.

Un immense merci également à Mesdames Evelyne Taroni et Véronique Guy pour leur accueil, pour le temps qu'elles m'ont accordé au travers de longues discussions ainsi pour leur accompagnement irréprochable pour braver bon nombre d'obstacles administratifs rencontrés durant ces années de thèse.

J'adresse un remerciement particulier à Nicolas Rougon, Catalin Fetita, Dancho Panovski et Florence Kouvahe avec qui j'ai partagé l'immense majorité de mes déjeuners. Ces personnes ont été d'un grand soutien technique et psychologique tout au long de ces années de thèse. Un remerciement spécial également à Dancho Panovski et à Christian Tulvan qui ont fait preuve d'une grande force en partageant leur bureau avec moi pendant tant de temps.

Il m'est impossible d'oublier Rania Bensaied Ghaly et Mihai Mitrea pour leur expertise et leur aide précieuse lors du point charnière de mes travaux qu'a été l'évaluation de mes résultats. Je les remercie sincèrement pour cela.

Je remercie vivement Yazid Chir et Olivier Rousseau pour m'avoir fait confiance et m'avoir offert l'opportunité de réaliser cette thèse en partenariat avec Be Bound. Je remercie également du fond du cœur toute l'équipe de Be-Bound pour leur dynamisme, leur chaleur et leur enthousiasme. Chacune de mes journées de travail chez Be-Bound a été des plus enrichissantes en plus d'être une réelle bouffée d'air frais.

Un grand merci est adressé aux étudiants que j'ai pu accompagner lors de leurs projets de fin de majeur HTI. Ces étudiants en plus d'apporter une autre dimension plus que bienvenue à mes travaux, ont su à plusieurs reprises par leur travail et leurs idées être d'une grande aide dans l'avancement de mes recherches. Pour cela, je remercie encore Titus Zaharia et Mihai Mitrea pour m'avoir offert l'opportunité d'encadrer ces groupes d'étudiants.

Institut Polytechnique de Paris
91120 Palaiseau, France

Je tiens naturellement à remercier aussi mes parents, ma famille et mes amis proches pour leur soutien, pour leur écoute et pour la confiance qu'ils m'accordent depuis toujours.

Mes derniers remerciements vont à Clara Morlière, qui partage ma vie et qui m'a accompagné, m'a aidé et m'a soutenu infailliblement dans mon travail et même au-delà au cours de ces années de thèse.

Institut Polytechnique de Paris
91120 Palaiseau, France

Sommaire

Institut Polytechnique de Paris
91120 Palaiseau, France

Institut Polytechnique de Paris
91120 Palaiseau, France

Table des figures

Institut Polytechnique de Paris
91120 Palaiseau, France

Institut Polytechnique de Paris
91120 Palaiseau, France

Institut Polytechnique de Paris
91120 Palaiseau, France

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 1. Introduction

Aujourd'hui, la plupart de la population des pays développés possède un téléphone mobile si bien que le nombre d'utilisateurs de téléphones mobiles a dépassé les 5 milliards en 2019[1]. De plus, avec le développement fulgurant des infrastructures basées sur l'utilisation des réseaux 3G/4G, il est devenu possible d'avoir accès à l'intégralité du contenu d'internet sur téléphone, y compris aux images/vidéos. En revanche, les opérateurs mobiles limitent la bande passante, selon différents modèles économiques, ce qui entraîne une limitation de la consommation des données. D'un point de vue utilisateur, l'enjeu est alors de pouvoir lire et transmettre des images de bonne qualité, tout en espérant une consommation de données la plus faible possible.

De plus, les téléphones mobiles dépassent massivement, en termes d'utilisation quotidienne, tout type d'appareils photos. Une des raisons de ce phénomène est liée notamment à possibilité de partager instantanément les images acquises sur des réseaux sociaux et autres plates-formes, ainsi que de les stocker, parfois même simultanément, sur des clouds personnels.

Etant donné que les images représentent une des sources majeures des activités de stockage et de transmission mobile, ce type de comportement induit une forte consommation de ressources. En effet, d'une part, les photos prises par les utilisateurs se multiplient et d'autre part, les constructeurs de téléphones mobiles améliorent continuellement la qualité des dispositifs d'acquisition d'image qui équipent leurs produits, ce qui se traduit par une forte croissance de la résolution des photos prises/transmises. A titre d'exemple, les résolutions habituellement rencontrées aujourd'hui sur des *smartphones* grand public sont de l'ordre de quelques milliers de pixels de côté. De tels niveaux de résolution restent néanmoins très peu exploités dans le cadre d'une utilisation générique. En effet, la grande majorité de ces images ne seront jamais zoomées et/ou visionnées sur un autre support qu'un écran de téléphone mobile d'une taille d'environ 6 pouces. Nous nous retrouvons donc dans un contexte d'utilisation inadaptée et massive de ressources grand public, en dépit de leur accessibilité limitée.

Si dans les pays industrialisés ces habitudes de consommations sont rendues possibles par les infrastructures réseaux, cela n'est pas forcément le cas dans les pays en voie de développement, où à la fois bande passante et connectivité sont nettement inférieures. Mentionnons également qu'aujourd'hui, environ 4 milliards de personnes à travers le monde et notamment dans les pays émergeants sont en mesure de posséder un *smartphone* et d'avoir accès à un réseau téléphonique mobile, mais n'ont pas un accès fiable à l'internet via les données mobiles ou même fixes. Dans ces conditions, télécharger et mettre en ligne des images de résolutions et qualité aussi importantes devient un réel défi.

Afin de proposer une offre de téléphones mobiles dans ces zones, les opérateurs mobiles se doivent d'aller vendre physiquement leurs cartes SIM. Néanmoins, la plupart du temps, le processus n'arrive pas à terme car afin d'activer une carte SIM, les vendeurs doivent prendre une photo de la carte d'identité du client. Cette photo est ensuite transmise au centre de traitement puis validée manuellement, et ce, dans un délai limite (48 heures par exemple). Malheureusement, en raison de la mauvaise couverture réseau et de la taille de ces photos, un nombre très important de transactions échoue due aux erreurs de transmission des photos de cartes d'identité, privant ainsi les populations de ces zones reculées d'avoir accès à un téléphone mobile.

[1] https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/

Institut Polytechnique de Paris
91120 Palaiseau, France

La société Be-Bound [BeBound], qui est à l'origine de ces travaux de recherche, vise notamment à développer des technologies permettant d'accéder aux contenus textuels d'Internet en utilisant seulement un réseau téléphonique mobile. Il est possible aujourd'hui, grâce aux technologies proposées par Be-Bound de pouvoir accéder aux données de l'Internet seulement avec un réseau 2G disponible. Si, pour des données textuelles de bande passante relativement faible, les solutions proposées par Be-Bound offrent des modèles tout à fait viables pour un déploiement industriel grandeur nature, il reste à trouver des solutions adéquates pour les contenus visuels/image, intrinsèquement gourmands en bande passante.

Nous sommes donc ici face à un problème à deux faces. D'un côté, nous nous retrouvons dans le cadre d'un besoin de compression à très bas débit d'images numériques, tout en conservant leurs détails critiques, afin de permettre l'accès à ces contenus dans des zones avec une couverture internet très faible.

De l'autre côté, nous nous adressons à une utilisation grand public d'images sur téléphone mobile dans des pays industrialisés, avec pour objectif de réduire la consommation de stockage (et par conséquent, énergétique) due aux utilisations de ces images, tout en garantissant une qualité adéquate au regard des différentes applications.

Ces deux cas mettent en lumière le besoin réel de ces recherches et ce, à toute échelle de compression d'image grand public qui se trouve ne pas être assuré par les codecs images les plus répandus à travers Internet soit JPEG et PNG.

Si ces deux facettes ne semblent pas être particulièrement corrélées, nous montrons dans cette thèse qu'elles se rejoignent en plusieurs points, et que les mêmes éléments de solution sont totalement applicables à ces différentes applications, notamment grâce à une utilisation adaptée d'algorithmes d'apprentissage par réseaux de neurones profonds.

Durant ces dernières années, l'apprentissage profond, que nous nommons dans ce manuscrit par sa dénomination anglophone « *deep learning* » (DL), a connu un développement fulgurant grâce à la disponibilité croissante de puissantes cartes graphiques (GPU) grand public, permettant de faire fonctionner efficacement et dans un temps raisonnable, de réseaux complexes de neurones profonds. En associant l'efficacité de ces réseaux de neurones et l'abondance de très grandes bases de données publiques d'images d'une grande variété, des progrès significatifs ont pu être réalisés dans les domaines du traitement et de l'analyse d'images en général, et plus particulièrement dans le champ de la compression d'image.

Ainsi, un nouveau type de compression basé sur des réseaux de neurones est apparu [Ballé16], reprenant le principe des auto-encodeurs [Hinton94]. Le principe consiste à effectuer les processus de compression et de décompression à l'aide d'un même réseau. Un auto-encodeur tente de reproduire en sortie l'image présentée en entrée, en passant par une succession de couches de neurones intermédiaires. En choisissant une couche dont la carte de caractéristiques est significativement plus réduite en taille que celle de l'image initiale, le réseau est amené à créer sa propre représentation compressée. La suite des traitements dans les couches successives peut être interprétée comme un processus de décodage, où il s'agit de reconstruire un contenu le plus fidèle possible à l'original.

Ce même principe a été revisité dans le cadre des réseaux antagonistes génératifs (GAN – *Generative Adversial Networks*) [Goodfellow14], qui ont connu récemment un important essor. Les réseaux GAN ont conduit à un nouveau paradigme de compression, appelée compression générative (GC –

Generative Compression) [Santurkar18], qui s'est montrée particulièrement efficace, notamment pour la compression à très bas débit.

Néanmoins, malgré sa spectaculaire efficacité, la compression générative présente un inconvénient majeur, voire critique et ce, particulièrement en compression à bas débit. En effet, en raison de la nature des GAN, ce type de compression tend à remplacer les petits détails contenus dans une image par autre représentation « plausible » de ces derniers. Cela peut entrainer de forts désagréments pour l'utilisateur qui voit sa photo non pas simplement dégradée par la compression mais modifiée en certains points, comme illustré Figure 1.1.

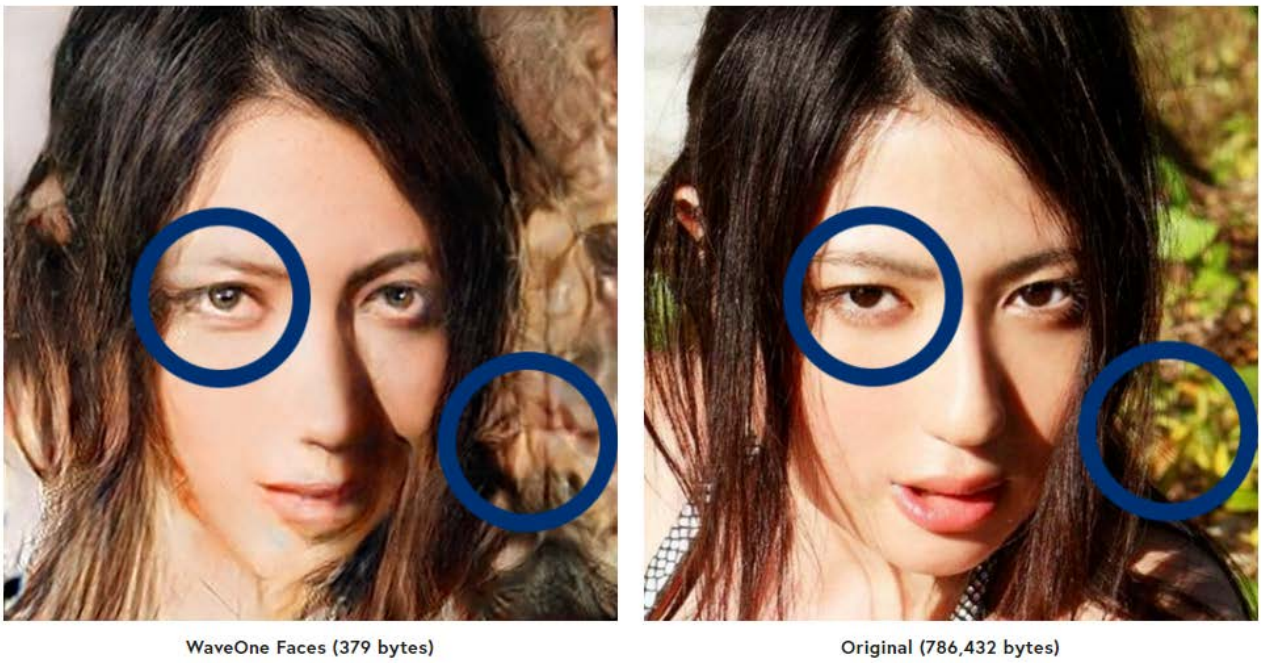


Figure 1.1 : Illustration des modifications que peut apporter la compression générative (Source : www.wave.one)

Dans le cadre d'une application de transmission de photos de cartes d'identité, comme celle évoquée précédemment, la technique conduit à des résultats inexploitables, car des informations critiques de la carte d'identité comme des chiffres ou des lettres se trouvent être remplacés par d'autres ou bien par des symboles inintelligibles (Figure 1.2).

Figure 1.2 : Exemple d'un cas où une inscription est remplacée par des symboles inintelligibles

Institut Polytechnique de Paris
91120 Palaiseau, France

Afin de répondre à ce type de limitations, nous proposons dans cette thèse des éléments de solution permettant une compression efficace de divers contenus images, tout en garantissant le respect de leur intégrité sémantique.

L'objectif de cette thèse est donc de concevoir, développer et mettre en œuvre de nouvelles méthodes de compression d'images à très bas débits, permettant notamment leur transmission/consommation sur des réseaux 2G via la technologie de Be-Bound. A travers cet objectif nous en sommes venus à identifier le problème connexe de mauvais dimensionnement des images mobiles par rapport aux habitudes de consommation dont elles font l'objet.

De par ces éléments, nous nous retrouvons face à un objectif possédant deux contraintes qui peuvent s'opposer. En effet, d'une part nous chercherons à mettre en place un système de compression puissant afin de permettre une compression efficace et sûre des contenus visuels. En revanche, cette compression doit prendre en considération que la plateforme destination sera en grande partie mobile, donc limitée en performances.

Pour répondre à ces contraintes, nous nous sommes orientés vers les normes de compression d'image (ou codecs) comme base. Ce choix a été retenu en raison de la faible complexité de ces derniers, ainsi que de leur fiabilité. En revanche, les codecs image traditionnels de compression avec pertes comme JPEG ou JPEG2000 ne permettent pas d'obtenir des taux de compression satisfaisants au regard de notre application. Des codecs plus récents, provenant de codecs vidéo, comme WebP et BPG permettent néanmoins d'obtenir de bien meilleurs taux de compression, en particulier dans la plage de très bas débits, tout en offrant une complexité de calcul tout à fait adaptée par rapport aux ressources disponibles sur les téléphones mobiles actuels. Pour nos recherches, nous nous sommes concentrés sur l'utilisation du format BPG, en raison de ses meilleures performances débit/distorsion par rapport au WebP et en raison de sa plus grande complexité d'artéfacts.

L'avantage principal que présente un codec standard par rapport à la compression générative, outre sa plus faible complexité de calcul, concerne le respect du contenu sémantique de l'image. Cela se paie en revanche par une qualité visuelle des images reconstruites bien inférieure. Dans nos recherches, nous montrons que l'utilisation du *deep learning* permet d'améliorer significativement la qualité des images obtenues par compression BPG, sans remplacer les détails critiques.

Les contributions proposées dans cette thèse s'articulent donc autour de l'utilisation conjointe de codecs existants, notamment BPG, et des méthodologies de *deep learning*.

Les différentes méthodes élaborées présentent l'avantage d'être génériques par rapport au codec utilisé et de se positionner à la fin de la chaîne de compression/décompression. Cela permet en outre une implémentation simple et robuste, qui ne requiert pas la modification des codecs standardisés. Enfin, soulignons également le côté réutilisable, malléable et améliorable de ces méthodes.

Institut Polytechnique de Paris
91120 Palaiseau, France

La suite de ce manuscrit est organisée de manière suivante.

Le chapitre II présente un état de l'art des normes de compression d'image Nous proposons notamment une analyse comparative, à la fois objective et subjective des codecs présentés. En ce qui concerne les aspects de performance, nous soulignons la difficulté d'établir des comparaisons sur des mesures purement objectives et montrons la nécessité de recourir à des protocoles d'évaluation subjective pour valider nos travaux.

Le chapitre III introduit les principes de base du *machine learning*, en se focalisant sur les techniques que nous exploiterons plus particulièrement dans nos développements, qui sont le *deep learning* et les réseaux de neurones convolutionnels (CNN – *Convolutional Neural Networks*), particulièrement adaptés aux traitements des images. Nous proposons ensuite une revue des différentes techniques liées à la compression d'image utilisant les réseaux CNN. Plus précisément, il s'agit des méthodes de super-résolution (SR), qui permettent d'augmenter la résolution d'une image en la dégradant le moins possible, des techniques dédiées de réduction d'artéfacts de compression ainsi que des algorithmes de compression de type auto-encodeurs.

Le chapitre IV introduit les réseaux de type GAN, qui sont au cœur de nos recherches. Ce type de réseau a la particularité de proposer des performances inégalées en termes de création de contenu visuel. Nous détaillons ensuite comment les GAN permettent d'obtenir de résultats supérieurs aux CNN classiques pour des objectifs à la fois de super-résolution, de réduction d'artéfacts, ou encore de compression générative.

Dans le chapitre V, nous présentons un nouveau schéma de compression associant super-résolution et compression BPG. Ce type de solution s'adresse principalement à la problématique de consommation d'images grand public sur mobiles. Ici, nous proposons de stocker/transmettre une version compressée de l'image à une résolution réduite. Cette image basse résolution compressée est ensuite reconstruite dans sa résolution originale au moment de sa visualisation. Ce type de solution permet alors de réduire fortement l'espace de stockage utilisé pour les images, sans dégrader sensiblement leur qualité.

Nous introduisons un schéma similaire au précédent mais qui abandonne l'étape de super-résolution, ce qui revient à effectuer une réduction d'artéfacts pour des contenus fortement compressés. Cette technique permet d'obtenir une meilleure fiabilité et permet notamment la restitution des détails fins d'une image comme les écritures ou numéros, répondant ainsi plus à la problématique liée aux applications d'accessibilité aux images numériques dans les zones à mauvaise couverture internet mobile et/ou fixe. Enfin, nous faisons état d'une amélioration des résultats obtenus précédemment, en utilisant comme base une amélioration ESRGAN du réseau initial SRGAN, proposée en septembre 2018. Nous montrons ici que l'utilisation de ce nouveau réseau, en conservant la même méthodologie permet d'obtenir des résultats supérieurs, tout en réduisant la complexité de calcul du processus d'apprentissage.

Une évaluation globale de nos résultats, incluant à la fois des métriques objectives et des protocoles subjectifs, est présentée au chapitre VI. Nous démontrons notamment l'importance de disposer des techniques d'évaluation subjective dans un contexte où les métriques objectives traditionnelles deviennent inappropriées, en raison des différentes fonctions de pertes utilisées par les réseaux de neurones.

Le chapitre VII traite le cas de transmission de cartes d'identité par le réseau SMS. Les techniques énoncées précédemment ne permettant pas d'assurer, en l'état, une qualité et une fiabilité acceptable pour ce cas d'usage, nous introduisons la nécessité d'utiliser des cartes de segmentation dans les schémas proposés afin d'obtenir une meilleure fiabilité dans les zones critiques d'une image.

Enfin, le chapitre VIII clôt ce mémoire de thèse, avec une conclusion sur l'ensemble de nos travaux, et une discussion sur les différentes pistes d'améliorations futures.

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 2. Normes de compression d'images : état de l'art

Résumé. Dans ce chapitre, nous étudions les principales normes de compression d'image qui puisse apporter des éléments de solution pour l'application envisagée dans nos travaux.

Nous commençons par présenter les formats de compression d'images avec pertes les plus répandus mais également les plus pertinents au regard de nos objectifs. Notre analyse inclut des normes/formats comme JPEG, JPEG200, WebP, AVIF ou encore BPG, dont les principes respectifs de fonctionnement sont rapidement exposés, en mettant en lumière les éléments clefs de la chaîne de traitement de chacun.

Nous procédons ensuite à une comparaison de ces différentes approches, toujours au regard de nos objectifs de recherche, ce qui nous permet de dresser une conclusion argumentée quant au choix du format BPG que nous avons retenu comme base pour nos futures expérimentations.

Mots clés : codecs, formats et normes de compression d'images, JPEG, JPEG2000, WebP, AVIF, BPG.

Institut Polytechnique de Paris
91120 Palaiseau, France

2.1 La norme ISO JPEG

La norme internationale JPEG (*Joint Photographic Experts Group*) [Wallace91], proposée par ISO (*International Standardization Organization*) est un format de compression avec pertes proposé en 1972 par Nasir Ahmed, et standardisé en 1992 par le *Joint Photographic Experts Group*.

Ce format permet d'atteindre une compression de 10:1 avec quasiment aucune perte visuelle perceptible et est aujourd'hui encore le format d'image le plus répandu à travers le monde.

Les principaux modules de la chaîne de compression JPEG sont illustrés Figure 2.1.

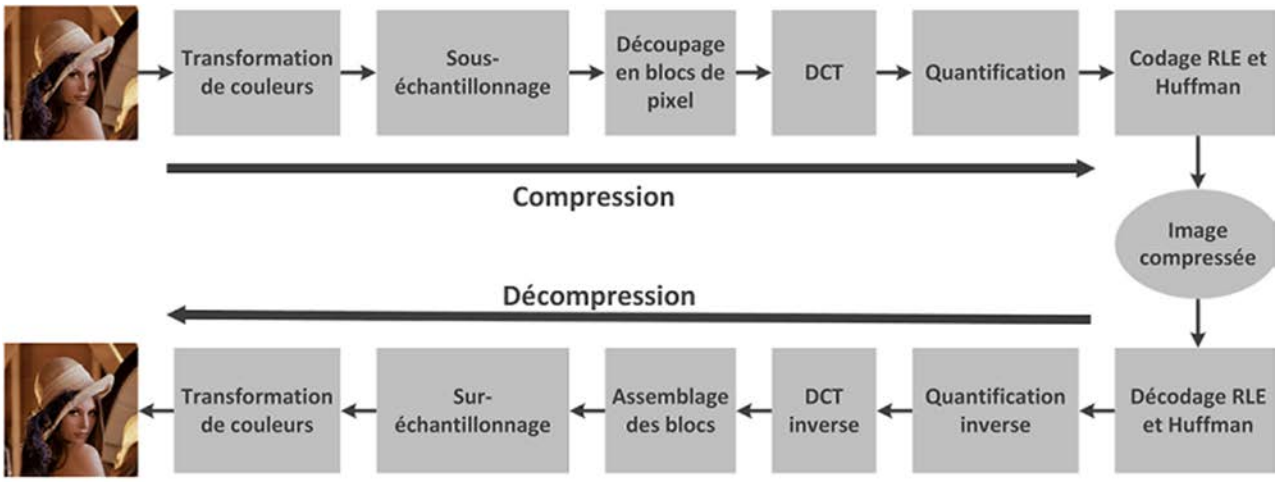


Figure 2.1 : Schéma de la compression/décompression JPEG

La première étape de la compression JPEG consiste au changement d'espace de couleur de l'image initiale (RVB le plus souvent) en un espace YCrCb. Le but de cette étape est que le format YCrCb propose une bien meilleure décorrélation des couleurs que le format RVB. Ceci est important dans la mesure où l'œil humain est beaucoup plus sensible au canal de Luminance Y qu'aux canaux de chrominances Cr et Cb (soit Chrominance Rouge et Chrominance Bleue).

Cela permet aussi d'appliquer une procédure de sous-échantillonnage des images de chrominance dans une seconde étape. En effet, l'œil humain est moins sensible aux hautes fréquences qui apparaissent dans les signaux de chrominance. Ainsi, une perte importante d'information à ce niveau-là reste relativement peu visible pour l'utilisateur. Par rapport à la manière dont l'échantillonnage est réalisé, trois modes de représentation différents sont retenus par la norme JPEG, notés par 4:4:4, 4:2:2 et 4:2:0 et explicités dans la suite.

Dans le mode 4:4:4 (Figure 2.2), aucun canal n'est sous-échantillonné. Il est utilisé notamment lorsque l'information colorimétrique est considérée comme très importante pour le contenu et doit être entièrement préservée.

Institut Polytechnique de Paris
91120 Palaiseau, France

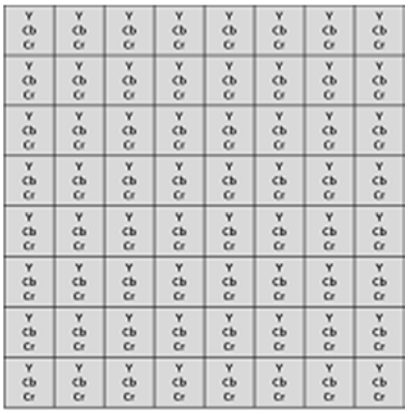


Figure 2.2 : Illustration des composantes YCrCb sur un bloc de 8×8 pixels en format 4:4:4

Le mode 4:2:2 (Figure 2.3) consiste en la suppression de l'information de chrominance une colonne sur deux, l'information de la colonne supprimée étant remplacée par la moyenne des valeurs des deux pixels ayant gardé leurs valeurs de chrominance au sein du groupement de 2×2 pixels.

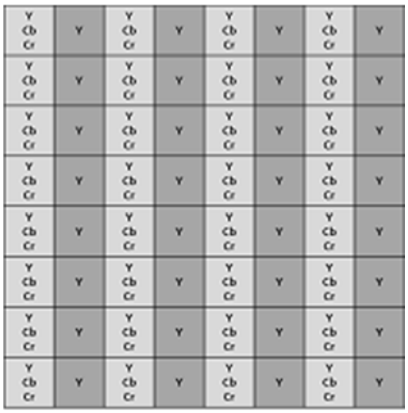


Figure 2.3 : Illustration des composantes YCrCb sur un bloc de 8×8 pixels en format 4:2:2

Enfin le mode 4:2:0 (Figure 2.4) reprend le principe du 4:2:2 mais en supprimant également l'information de chrominance une ligne sur deux.

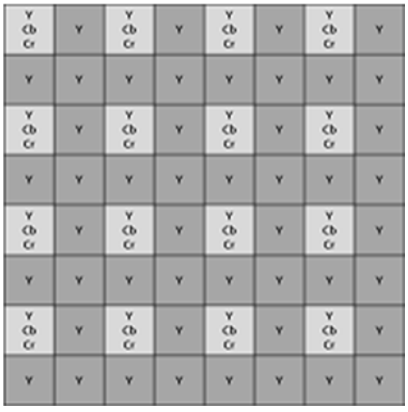


Figure 2.4 : Illustration des composantes YCrCb sur un bloc de 8×8 pixels en format 4:2:0

Institut Polytechnique de Paris
91120 Palaiseau, France

L'image est ensuite découpée en blocs de 8×8 pixels, sur lesquels un ensemble d'opérations successives est effectué.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

Figure 2.5 : Exemple de blocs de 8x8 pixels (un seul canal) d'une image

L'élément principal de la compression JPEG est la transformée en cosinus discrète (DCT – *Discrete Cosine Transform*), qui fournit une représentation fréquentielle fondée sur la transformée de Fourier et permet une séparation spatiale des hautes et basses fréquences de l'image.

$$DCT(i,j) = \frac{2}{N} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (2.1)$$

$$\text{avec } C(x) = \begin{cases} \frac{1}{\sqrt{2}} \; pour \; x = 0 \\ 1 \; pour \; x > 0 \end{cases}$$

Le cœur de la compression JPEG s'appuie sur le constat que l'œil humain apparait comme beaucoup moins sensible aux hautes fréquences d'une image qu'aux basses fréquences. Ainsi, comme la transformée DCT permet d'isoler les hautes fréquences, il est possible de les atténuer, voire de les supprimer, ce qui revient à réduire drastiquement la quantité de données à coder.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix} \Rightarrow F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

a. Exemple d'un bloc de 8×8 pixels (un seul canal) d'une image.

b. Sa transformée DCT (coefficients arrondis à la plus proche valeur entière).

Figure 2.6 : Exemple de transformation des pixels en coefficients DCT

Dans la Figure 2.6.b, les hautes fréquences selon les deux dimensions spatiales horizontale et verticale sont respectivement regroupées à gauche et en bas de la matrice des coefficients DCT. Nous pouvons également observer que les valeurs absolues des coefficients de hautes fréquences sont significativement inférieures à celles des basses fréquences. En appliquant une procédure de quantification des coefficients, il devient alors possible de diminuer de façon drastique la quantité d'information de la représentation obtenue.

Institut Polytechnique de Paris
91120 Palaiseau, France

La DCT étant une opération complétement réversible (aux arrondis près), c'est à partir de l'étape de quantification que le processus devient « avec pertes ». La quantification proposée par JPEG est réalisée de manière adaptative par rapport à la fréquence spatiale des coefficients DCT. Pour cela, avant une quantification uniforme contrôlé par un paramètre global de quantification q_{global}, on réalise au préalable une division des coefficients de la DCT par les coefficients d'une matrice de quantification.

Plus précisément, la quantification des coefficients DCT est effectuée comme décrit dans l'équation suivante :

$$F^*(u,v) = \left[\frac{F(u,v) + \left[\frac{Q(u,v)}{2}\right]}{Q(u,v)}\right] \cong entier\ le\ plus\ proche\ \left(\frac{F(u,v)}{Q(u,v)}\right) \tag{2.2}$$

De cette manière la quantification est réalisée avec des pas plus fins pour les coefficients de basse fréquence et au contraire, aves des pas plus importants pour les hautes fréquences.

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

a. Matrice de quantification JPEG.

$$F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix} \quad F^* = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

b. Coefficients DCT — Coefficients DCT après quantification

Figure 2.7 : Quantification adaptative en fréquence des coefficients DCT.

Dans l'exemple illustré Figure 2.7, nous pouvons constater qu'un grand nombre de coefficients est mis à zéro après la procédure de quantification. La dernière étape consiste à transformer la matrice des coefficients quantifiés en un code binaire. Pour cela, la norme JPEG s'appuie sur deux mécanismes de codage successifs, sans perte d'information.

Le premier est un codage RLE (*Run Length Encoding*) [Capon59]. Le codage RLE consiste à remplacer toutes les suites de caractères identiques dans une séquence de symboles par le nombre de caractères suivi dudit caractère. Par exemple, pour la chaîne "MMMMMMMBBBBMMM", un codage RLE donnerait "7M4B3M". Afin d'exploiter la forme particulière des coefficients DCT quantifiés et notamment le nombre important de coefficients quantifiés à zéro, la matrice DCT est scannée en zigzag pour obtenir une séquence de symboles, comme illustré Figure 2.8.

Institut Polytechnique de Paris
91120 Palaiseau, France

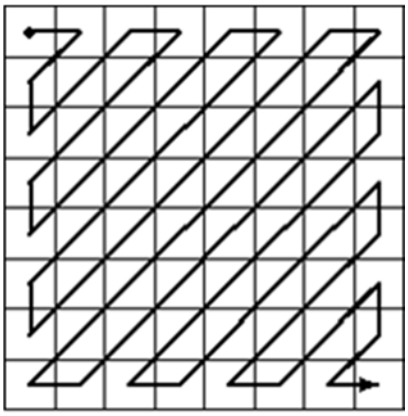


Figure 2.8 : Schéma de parcours en zigzag du bloc pour l'encodage RLE.

Enfin, la séquence des symboles RLE obtenue est codée entropiquement avec un codeur binaire de Huffman pour obtenir le flux de bits que représente l'image compressée.

Au niveau du décodeur, les opérations inverses sont successivement appliquées, comme illustré Figure 2.1.

Le format JPEG est aujourd'hui le format le plus répandu à travers le monde. Les raisons de cette popularité sont sa simplicité d'implémentation et le fait qu'il soit pionnier dans le domaine du stockage et de la diffusion d'images numériques. En outre, les trois principes génériques de décorrélation par transformée, quantification et codage entropique sont largement repris par les normes et formats plus récents et constituent d'une certaine manière les piliers de tout algorithme de compression d'images.

Néanmoins, ce format souffre de fortes limitations et est depuis de nombreuses années très largement dépassé en termes de capacité de compression par de multiples normes et formats, dont nous détaillerons les principales caractéristiques dans la suite.

2.2 JPEG 2000

La norme JPEG2000 [JPEG2000] a vu le jour en 2000, avec l'avènement notamment des appareils photos numériques et avec la motivation de dépasser les contraintes techniques imposées par le format JPEG. En effet, à très bas débit, soit en dessous de 0.25 bpp (*bits par pixel*), le format JPEG ne propose pas une qualité acceptable. En outre, grâce à ses propriétés de scalabilité, la norme JPEG200 proposait également un cadre unifié pour la compression avec ou sans pertes. Le schéma synoptique de la chaîne de traitement JPEG200 est illustré Figure 2.9.

Institut Polytechnique de Paris
91120 Palaiseau, France

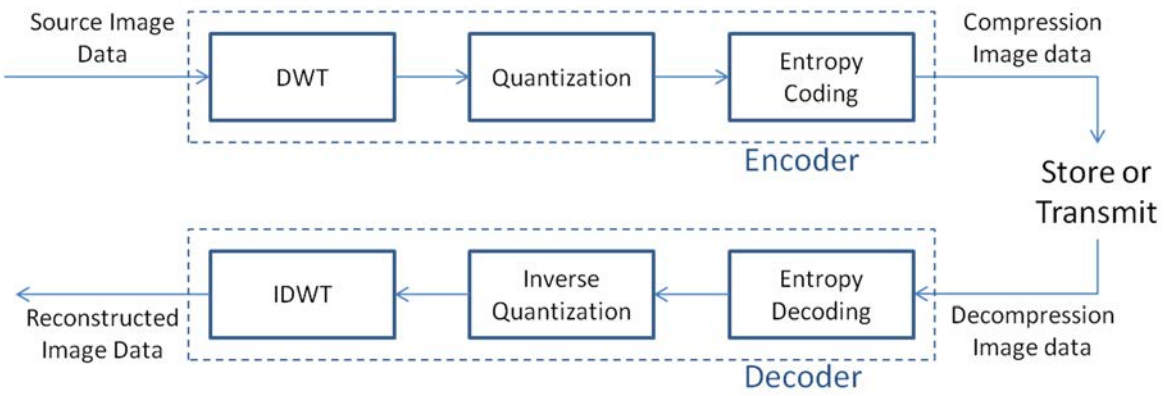


Figure 2.9 : Schéma d'encodage/décodage JPEG2000

La norme JPEG2000 abandonne le paradigme jusque-là largement utilisé de codage par blocs/macroblocs et utilise une représentation multi-résolution acquise à l'aide d'une transformée en ondelettes discrètes (DWT-*Discrete Wavelet Transform*) [Heil89] appliquée de manière globale sur l'image Figure 2.10. Le caractère multi-résolution de la représentation en ondelettes permet d'obtenir une transmission progressive et séparable de l'image.

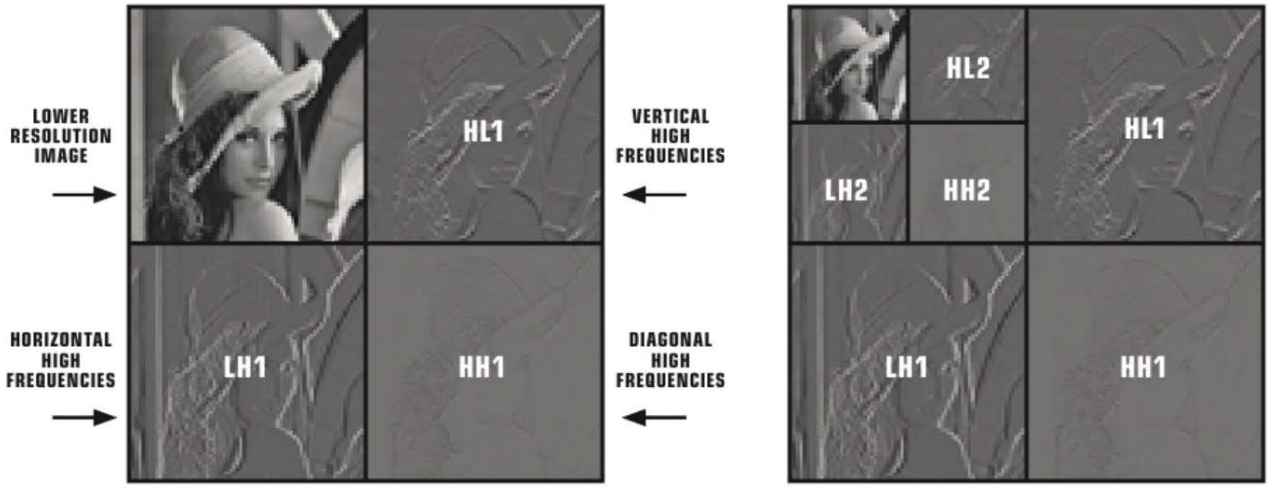


Figure 2.10 : Illustration de la décomposition successive de l'image (Source : https://www.slideshare.net/guestd38f1/intopix-everything-about-jpeg2000)

Les coefficients de la transformée sont quantifiés par un quantifieur scalaire uniforme à zone morte [Yu04]. Chaque sous-bande est ensuite divisée en blocs rectangulaires appelés *codeblocks*. Chaque *codeblock* est décomposé/quantifié en plans binaires, et encodé par la technique dite "*Embeded Block Coding with Optimized Truncation*" (EBCOT) [Baarir11].

Les performances de la norme JPEG2000, surpassent amplement celles de JPEG en termes de qualité visuelle et d'efficacité de compression. Néanmoins, son utilisation commerciale a été pénalisée par les droits de propriété intellectuelle associés aux technologies normatives considérées, qui en font, *de facto*, un standard payant.

Plus récemment, en 2010, les grands acteurs du marché du multimédia sur le web, ont proposé des technologies propriétaires. C'est le cas du format WebP de Google, brièvement décrit dans la section suivante.

2.3 Le format WebP

WebP [WebP] est un format de compression d'images représentant la partie « intra » (*still picture*) du VP8 [Bankoski11], codec vidéo propriétaire développé par Google. Il s'agit d'un codec reprenant le schéma classique de JPEG, avec prédiction spatiale et transformation des erreurs de prédiction par DCT (Figure 2.11).

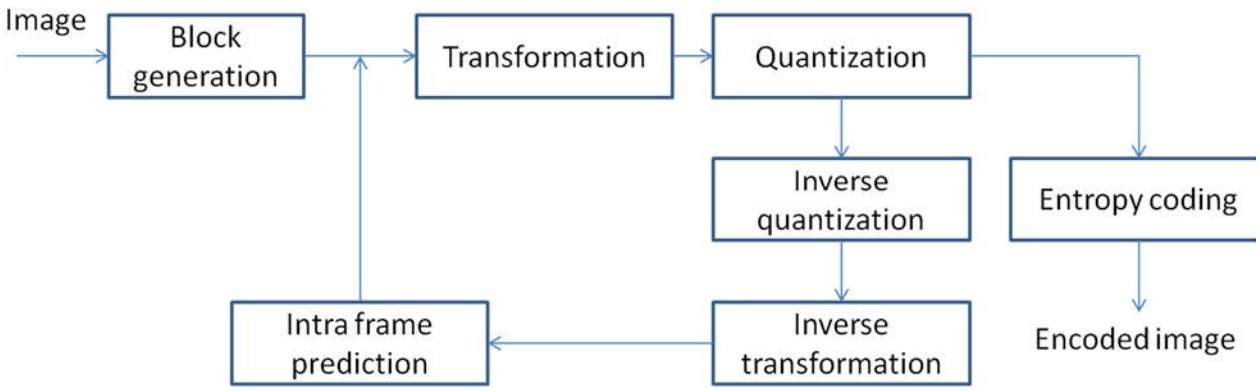


Figure 2.11 : Schéma d'encodage d'un codec "*block-based*" adopté par le format WebP

Les coefficients des résidus de prédiction sont transformés par DCT, puis quantifiés et enfin encodés par un codeur arithmétique binaire avec contextes (*Context-adaptive binary arithmetic coding* - CABAC) [Marpe03].

Notons ici qu'une différence majeure est présente par rapport au JPEG. Il s'agit de l'étape dite de prédiction spatiale, dont le principe consiste à estimer la valeur d'un pixel courant à partir d'un ensemble donné de pixels déjà codés, situés dans un voisinage causal par rapport à un ordre donné de parcours de l'image, comme illustré Figure 2.12.

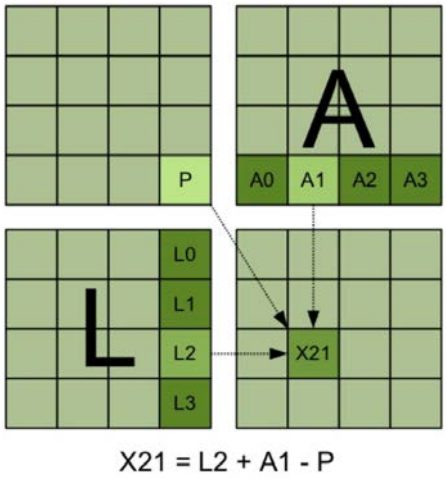


Figure 2.12 : Mode de prédiction "TM" de WebP/VP8 (source [VP8])

Institut Polytechnique de Paris
91120 Palaiseau, France

Cette étape de prédiction spatiale permet de décorréler davantage le signal image, ce qui réduit de manière considérable la quantité d'information à coder et à transmettre.

En pratique, prédire avec précision ces pixels est trop compliqué, voire impossible. Néanmoins, il est tout à fait possible de ne transmettre que la différence entre la valeur réelle du pixel et la valeur prédite, appelée **erreur de prédiction**. Cette erreur de prédiction est caractérisée par la décorrélation de ses échantillons ainsi que par une plage de valeur fortement concentrée autour d'une valeur nulle, d'entropie beaucoup plus faible que celle du signal d'origine.

Ce ne sont néanmoins pas directement les erreurs de prédiction qui seront transmises. En effet, un gain supérieur est possible si ces erreurs de prédiction sont transformées par DCT. Ce sont alors les coefficients DCT des erreurs résiduelles de prédiction qui sont quantifiés et finalement codés binairement à l'aide du codeur arithmétique CABAC.

Une autre différence majeure entre les schémas de compression WebP et JPEG concerne le découpage en blocs. Dans le cas de WebP, on abandonne la technique simpliste de partition de l'image en blocs de taille fixe, en faveur d'une partition en blocs de tailles variables, adaptés aux contenus de l'image. Ainsi, des régions adjacentes présentant une certaine homogénéité pourront figurer plus facilement au sein du même bloc à traiter, rendant la compression beaucoup plus efficace. De la même manière, si deux régions adjacentes n'ont rien à voir entre elles, la probabilité qu'elles soient traitées ensemble, et donc qu'elles bénéficient d'une prédiction peu pertinente, se voit fortement réduite.

WebP supporte 3 types de macroblocs : (4×4) et (16×16) *Luma* (*i.e.*, le signal de luminance) et (8×8) *Chroma* (pour les deux signaux de chrominance correspondant à l'espace de couleurs YUV). Il possède également 10 modes de prédictions différents. Parmi ces 10 modes, 4 sont partagés entre les 3 types de macroblocs retenus : vertical, horizontal, DC et TrueMotion-TM (Figure 3). Les 6 autres ne concernent que les blocs (4×4) Luma. Ces opérations sont effectuées sur des macroblocs de taille allant de (4×4) à (16×16) pixels.

2.4 Le format BPG

Le format BPG (*Better Portable Graphic*) [BPG] est un format crée en 2014 par Fabrice Bellard. Il représente la partie *intra* du codec vidéo MPEG-4 HEVC/H.265 [Sullivan12], créé en 2012. En plus d'utiliser exclusivement cette partie intra, BPG remplace le header de HEVC par un header bien plus compact de seulement 4 octets.

Le format BPG offre une très nette amélioration de performances de compression par rapport à ses prédécesseurs. Il n'est en revanche que très peu utilisé en raison de sa non-gratuité, due aux brevets déposés sur HEVC, et plus particulièrement sur l'encodeur x265 [x265] qui se trouve être une partie intégrante de l'implémentation de BPG.

La Figure 2.13 illustre le schéma synoptique de codage BPG, avec ses principales composantes.

Institut Polytechnique de Paris
91120 Palaiseau, France

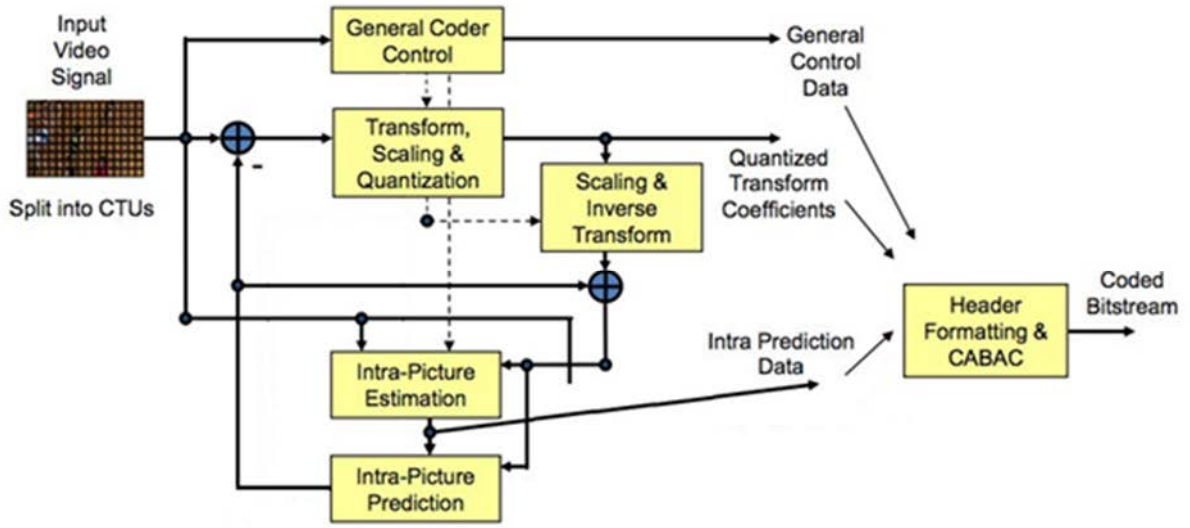


Figure 2.13 : Schéma d'encodage de BPG (Source : [Sullivan12])

Un des principes novateurs de BPG concerne la division de l'image en un arbre de blocs (*Coding Tree Blocs* - CTB) de taille variable et adaptée aux contenus de l'image. Ce processus est illustré Figure 2.14. Nous observons que pour des régions relativement uniformes, la taille des blocs est plus importante. Au contraire, des blocs de taille inférieure sont utilisés pour les régions hautement texturées ou au niveau des contours de l'image.

Les blocs obtenus sont finalement divisés à leur tour dans des blocs pouvant supporter l'application de différentes transformées, appelés TB (*Transform Blocs*), et ayant des tailles allant de (4×4) à (32×32) pixels pour le *Luma*, et de (8×8) à (32×32) pixels pour le *Chroma*. La partition en TB suit une structure de *quadtree* [Samet84], déterminée à l'aide d'un algorithme d'optimisation débit-distorsion (RDO – *Rate-Distortion Optimization*) [Sullivan98].

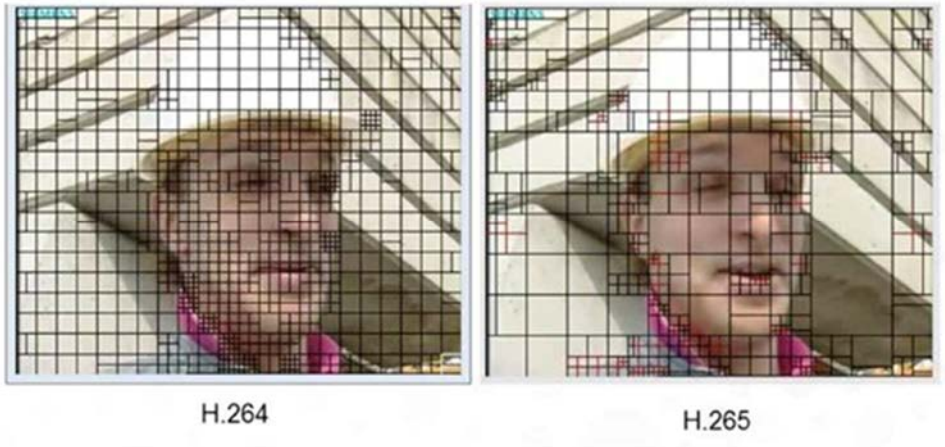


Figure 2.14 : Comparaison division en macroblocs h.264/h.265

Une prédiction *intra* est ensuite appliquée à ces macroblocs. Dans BPG, un nombre total de 37 modes de prédiction *intra* sont utilisés. Quatre de ces modes sont partagés entre les blocs *Luma* et *Chroma* (Vertical, Horizontal, DC et Plan). Les 31 autres modes directionnels sont utilisés exclusivement pour

Institut Polytechnique de Paris
91120 Palaiseau, France

les blocs *Luma* (Figure 2.15). Enfin, les deux autres modes restants (*Intra Bloc Copy* et *Palette Coding*) sont utilisés quasi exclusivement pour les images de type *Screen Content* [Zhu14], *i.e.*, créées par ordinateur ou dessinées.

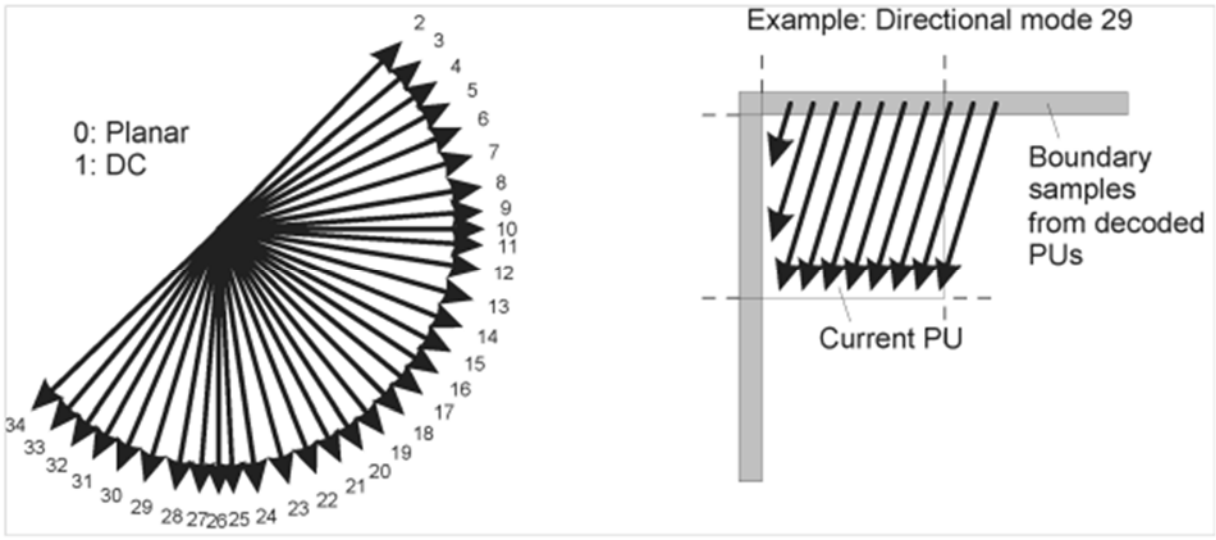


Figure 2.15 : Modes de prédiction intra de HEVC

Notons que le choix du mode de prédiction est réalisé en même temps que celui du macrobloc à utiliser, toujours à l'aide du mécanisme RDO.

Enfin, comme pour WebP, les erreurs de prédiction sont transformées par DCT (ou DST – *Discrete Sine Transform* - pour le cas particulier des blocs *Luma* de taille (4×4)), quantifiées, et encodées par un codeur CABAC.

2.5 AVIF (AV1 Still Image Format)

AV1 Chen18 est un codec vidéo ouvert et libre de droits, sorti officiellement le 28 mars 2018 par AOMedia, *l'Alliance for Open Media* (www.aomedia.org).

De grands acteurs d'internet comme Youtube ou Netflix ont déjà annoncé leur intention de transcoder leurs contenus dans le format AV1. Sa prise en charge par tous les navigateurs a été annoncée pour 2019, et en 2020 sont attendues les premières implantations hardware (Intel ayant également participé au développement de AV1).

Comme pour ses prédécesseurs, un format d'image fixe (AVIF) représentant la partie intra de AV1 a été développé et spécifié en version 1.0 en février 2019.

La différence majeure entre BPG et AVIF, à très bas débit, concerne le temps d'encodage en AVIF (plusieurs secondes), qui est bien plus important que celui de BPG (quelques millisecondes).

AV1 reste techniquement exactement dans la même lignée que ses prédécesseurs (VP8-9) dont il s'inspire, en proposant un algorithme fondé sur une répartition en blocs. En effet, dans AV1, la partition en superblocs va cette fois-ci d'une taille de (128×128) à (4×4) pixels et suit un schéma de partition permettant d'obtenir et d'utiliser des blocs rectangulaires, comme illustré Figure 2.16.

Institut Polytechnique de Paris
91120 Palaiseau, France

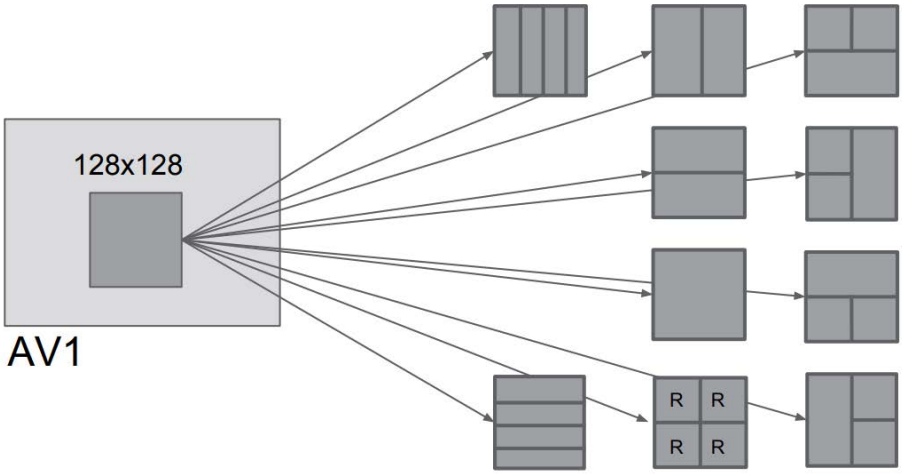


Figure 2.16 : Division en superblocs dans AV1 (source [Massimo17])

De nombreux modes de prédiction ont aussi été ajoutés. Ainsi, pour la partie *intra*, un nombre de 56 modes de prédiction directionnelle sont utilisés. L'ancien mode « *True Motion* » de VP8/9 a été remplacé par un mode *Paeth Predictor*. Enfin, des modes de type *Smooth Predictors*, dont la prédiction est effectuée en réalisant une somme pondérée par interpolations quadratiques des pixels de référence, ont été également ajoutés. Ces *Smooth Predictors* sont particulièrement efficaces sur les blocs contenant des gradients de valeurs relativement légers. Les modes *Paeth* et *Smooth Paeth* sont illustrés Figure 2.17, $P_{i,j}$ étant le pixel à prédire, les autres pixels colorés représentant les pixels de référence. Dans le cas du *Smooth Paeth*, ici, $P_{i,j}$ est déterminé par une somme pondérée de T_i, R_j, B_i et L_j .

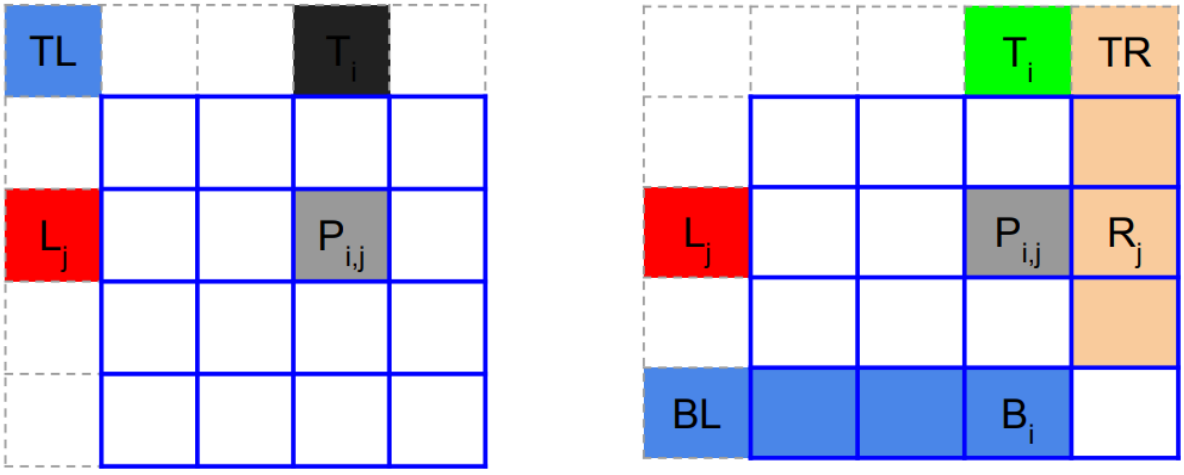


Figure 2.17 : *Paeth Prediction* en version standard et *Smooth* (source [Massimo17])

Un nombre de 4 types de transformées différentes (DCT, DST, Flipped AsymetricDST et Identité) sont utilisées pour coder les erreurs résiduelles de prédiction.

Enfin, un paramètre de quantification global allant de 0 à 63 est proposé, comme pour le format VP9 [Mukherjee15].

Institut Polytechnique de Paris
91120 Palaiseau, France

2.6 Évaluation comparative préliminaire

Pour cette évaluation préliminaire, nous avons utilisé le bien connu corpus Kodak [Kodak], qui propose 24 images de très bonne qualité représentées en « *true color* » (24 bpp). Pour l'heure, nous n'avons pas retenu les mesures de qualité traditionnelles comme le PSNR (*Peak Signal to Noise Ration*) ou SSIM (*Structural Similarity*) [Zang04]. Ce choix s'appuie sur le fait que, d'une part, ces mesures ne semblent plus être adaptées aux nouveaux formats et techniques de compression, comme expliqué dans [Wang09][Lin11][Chikkerur11]. D'autre part, ce genre de type de comparaison a déjà été réalisé de nombreuses fois auparavant [Harikrishnan17][WyohKnott], ces dernières mettant aussi BPG et AV1 comme état de l'art.

Pour illustrer ces propos, la Figure 2.18 présente une image compressée en deux versions, JPEG et BPG, pour des valeurs PSNR équivalentes (34,5 et 34,7 dB, respectivement). La qualité visuelle de l'image BPG est à l'évidence bien supérieure à celle compressée en JPEG. Ainsi, la mesure PSNR n'arrive pas dans ce cas à prendre en compte la qualité perceptuelle globale de l'image.

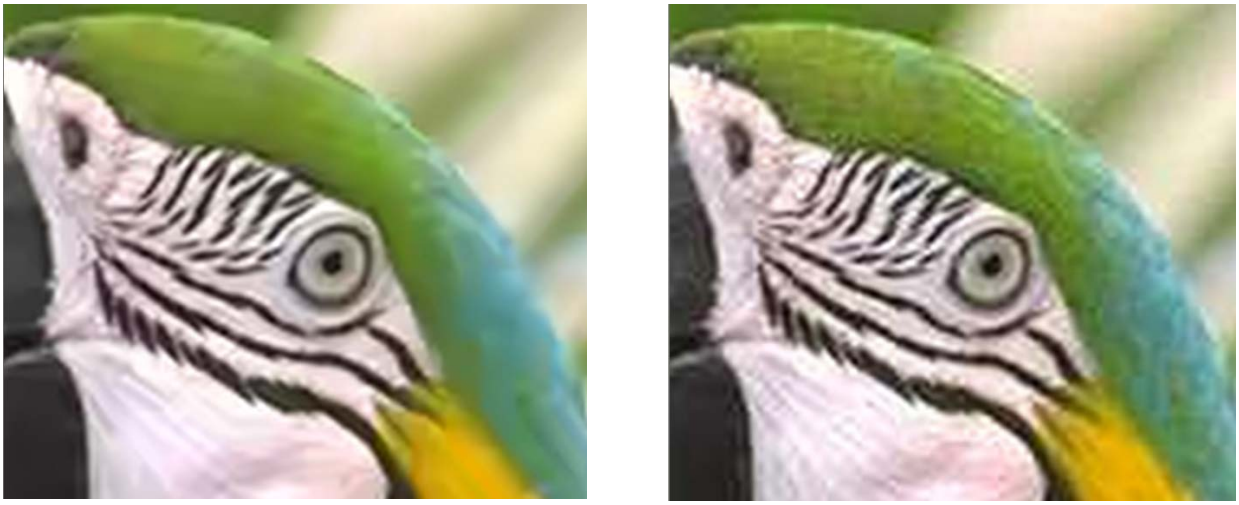


Figure 2.18 : Comparaison BPG (à gauche) et JPEG (à droite) à PSNR quasiment égal (34,5 et 34,7 dB)

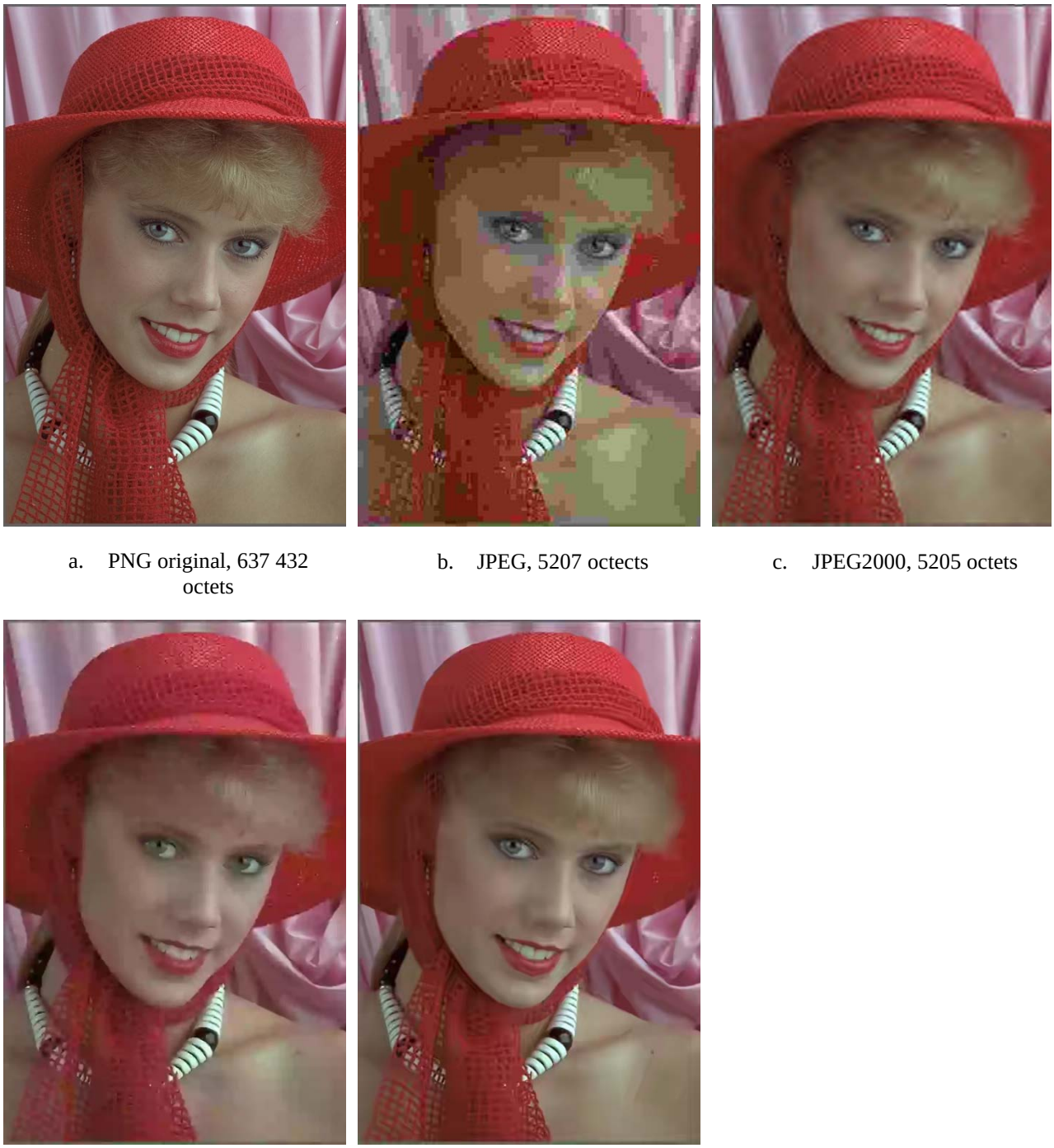
Nous avons donc pris le parti d'utiliser pour l'évaluation uniquement des critères subjectifs de pertinence visuelle. Ce choix se justifie également par rapport au cas d'usage demandé par notre partenaire industriel. L'objectif est d'assurer les débits cibles, pour une qualité d'image lisible et considérée comme acceptable de point de vue visuel.

En outre, un protocole plus rigoureux visant à déterminer la qualité de nos travaux sera mis en place ultérieurement, la métrique utilisée étant le *Mean Opinion Score* (MOS) [Huynh-Thu10]. Ce type de protocole étant régit par des normes internationales, possède de nombreuses contraintes comme l'environnement de visualisation. Concernant le nombre de personnes à convier pour ce protocole, les normes recommandent un panel de minimum 15 personnes.

La notion d'évaluation et de qualités objective et subjective sera traitée en profondeur plus loin dans le manuscrit, au Chapitre 6.

La comparaison visuelle présentée Figure 2.19 permet de mettre en évidence les différences de qualité à 0,1 bpp des différents formats introduits précédemment.

Institut Polytechnique de Paris
91120 Palaiseau, France



a. PNG original, 637 432 octets

b. JPEG, 5207 octects

c. JPEG2000, 5205 octets

d. WebP, 5028 octets

e. BPG, 4720 octets

Figure 2.19 : Comparaison des formats de compression d'image à 0,1 bpp (résolution de l'image : 512×768 pixels).

De ce premier exemple, il ressort que d'une part, le format JPEG apparait comme totalement non-adapté aux problématiques de compression à très bas débit. Cette constatation n'est pas un incident isolé mais un phénomène récurrent pour l'ensemble d'images de test que nous avons utilisées.

Quelques autres exemples sont également présentés Figure 2.20, Figure 2.21 et Figure 2.22, pour différents débits et qualités.

Institut Polytechnique de Paris
91120 Palaiseau, France

a) Originale PNG

b) JPEG, 7975 octets

c) AV1, 7913 octets

d) BPG, 7457 octets

e) WebP, 7518 octets

f) JPEG2000, 7850 octets

Figure 2.20 : Comparaison des différents formats sur image zoomée à 0,15 bpp

Institut Polytechnique de Paris
91120 Palaiseau, France

a) Originale PNG

b) JPEG, 6915 octets

c) JPEG2000, 6997 octets

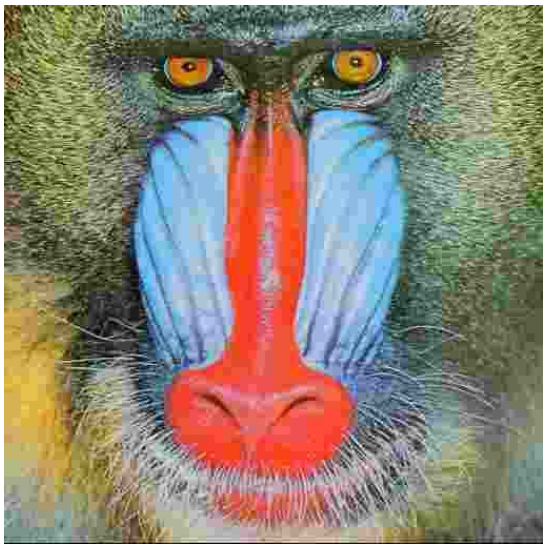
d) WebP, 6948 octets

e) BPG, 6432 octets

f) AVI, 6849 octets

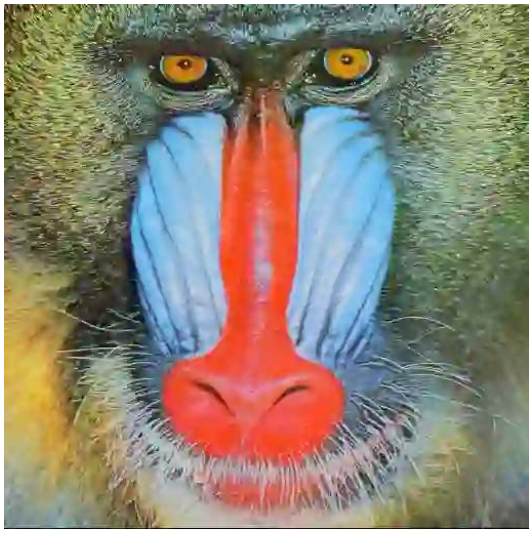
Figure 2.21 : Comparaison des différents formats sur image non zoomée à 0,16 bpp

a) Originale PNG



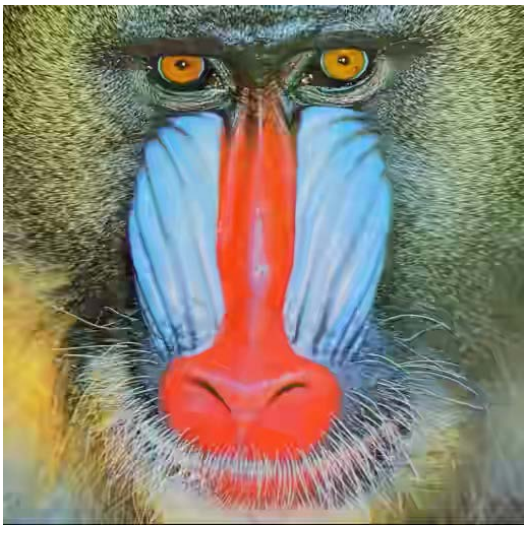
b) JPEG, 11378 octets

c) JPEG2000, 10465 octets



d) WebP, 10618 octets

e) BPG, 10892 octets



f) AV1, 10226 octets

Figure 2.22 : Comparaison des formats sur image non zoomée à 0,32 bpp (Compression AV1 au maximum, q=63)

Institut Polytechnique de Paris
91120 Palaiseau, France

Nous pouvons également insister que sur une image zoomée (Figure 2.20.et Figure 2.21) et sur une image non-zoomée (Figure 2.22 et Figure 2.23), il est très difficile, voire impossible de différencier une image compressée à très bas débit en BPG ou en AV1.

a. AVIF 12,9 ko 0,09 bpp b. BPG 12,4 ko, 0,09 bpp

Figure 2.23 : Comparaison entre AVIF et BPG à très bas débit

Des comparaisons et tests plus détaillés sont également disponibles sur Xiph.org [AV1Demo]. Ces tests ont permis de montrer que BPG et AV1 proposent actuellement, la meilleure qualité de compression.

Néanmoins, malgré l'aspect très prometteur de AV1 (AVIF), un problème a été soulevé par rapport à notre objectif, ceci étant bien apparent dans le cas de la Figure 2.23. Dans ce cas, AV1 été poussé au maximum de ses capacités de compression (paramètre de quantification fixé à 63) et n'a pu fournir ici qu'une image de 12,9 ko, ce qui représente un débit de 0,09 bpp.

En revanche, BPG avec paramètre de quantification maximal (q=51) pour cette même image a pu descendre jusqu'à 3,5 ko, soit 0,025 bpp. Dans ce cas, la qualité visuelle est faible, mais le contenu sémantique de l'image reste globalement préservé. (Figure 2.24).

Figure 2.24 : Image BPG compressée au maximum (paramètre de qualité à 51), 3,5 ko, 0,025 bpp

Institut Polytechnique de Paris
91120 Palaiseau, France

Cette évaluation préliminaire confirme les résultats rapportés dans la littérature [Harikrishnan17], [WyohKnott], qui établissent le format BPG comme état de l'art des normes de compression, au moins par rapport aux taux de compression obtenus. Néanmoins, pour atteindre l'objectif d'application considéré (transmission sur paquet d'environ 10 SMS), il est nécessaire d'augmenter davantage les taux de compression.

L'amélioration attendue doit représenter une réduction de 50% du débit, pour une qualité visuelle équivalente, par rapport à BPG. Après étude et validation des objectifs avec le partenaire industriel, nous avons établi qu'une résolution d'image de (256×256) pixels serait suffisante pour une visualisation sur *smartphone*. A cette résolution, le format BPG offre actuellement une qualité acceptable pour un débit autour de 2.5-3 ko. Or, un débit maximum de 1.4 ko est requis.

Pour cette raison, nous avons retenu une première piste de développement, qui concerne l'optimisation/l'adaptation des différentes étapes impliquées dans la chaîne de traitement de la norme BPG. Plusieurs travaux de recherche visant à apporter des améliorations à la norme BPG ont déjà été rapportés dans la littérature ces dernières années. Ils sont passés en revue dans la section suivante.

Institut Polytechnique de Paris
91120 Palaiseau, France

2.7 Amélioration de la norme BPG : état de l'art

L'état de l'art considère de manière plus globale les améliorations qui peuvent être apportées à la norme MPEG-4/HEVC (H265), dont BPG est la sous-partie correspondant au codage en mode *intra*. Il fait ressortir deux grands axes de développement méthodologique, un premier qui concerne la réduction de la complexité de calcul et un second portant sur la réduction du débit.

2.7.1 Réduction de la complexité

Un des champs de recherche le plus actif sur les améliorations qui peuvent être apportées à la norme HEVC concerne la réduction de la complexité de calcul aussi bien au niveau du codeur que du décodeur. La plupart du temps, la réduction de la complexité est obtenue en éludant certaines étapes chronophages comme le choix de la taille des macroblocs [Kibeya14], ou la sélection du mode de prédiction [Lei16].

En effet, dans la version initiale de HEVC, tous les partitionnements en blocs sont possibles. De plus, tous les modes de prédiction sont calculés afin de définir la combinaison conduisant à la meilleure compression, au sens de l'optimisation RDO. Il est en revanche possible de s'appuyer sur les caractéristiques spécifiques de l'image à traiter afin de réduire le nombre de combinaisons à tester.

Une autre piste consiste à utiliser des transformées moins complexes comme par exemple une *Integer Cosine Transform* à la place de la DCT [Fong15].

La réduction de la complexité n'est pas un champ concernant directement notre problématique. En revanche, beaucoup de ces recherches ont permis d'améliorer le format au fil des années, permettant une utilisation moins contraignante et rendant notamment son utilisation envisageable sur une plate-forme mobile.

2.7.2 Réduction du débit

Relativement peu de travaux s'attaquent à la réduction du débit. Dans [Fracastoro16], il est proposé de garder la transformée DCT, tout en l'enrichissant à l'aide d'une approche directionnelle, pour l'adapter au contenu local de l'image. Le principe consiste à effectuer une rotation des vecteurs de la base DCT selon un ensemble de directions déterminées selon un critère RDO (on parle alors de *Steerable DCT*). Cette méthode démontre des résultats supérieurs à la transformée DCT lorsqu'elle est appliquée à des macroblocs de taille 16x16 minimum.

De manière plus générique, dans [Selesnick11], un ensemble de transformées suivant plusieurs directions diagonales sont déterminées en fonction du bloc courant. L'inconvénient majeur de cette méthode est lié au caractère non-séparable des transformées obtenues, qui, du coup, ne se prêtent pas à la mise en œuvre d'algorithmes rapides.

Un autre axe de développement concerne le cas des images de type "*Screen Content Images* - SCI" [Zhu14]. Popularisées par l'émergence d'applications comme le partage de l'écran, les jeux en réseaux ou encore la vidéoconférence, ces images sont le résultat d'un rendu hybride de plusieurs types de contenus visuels, incluant texte, éléments graphiques et naturels. Cela est caractéristique aux pages web et aux images affichées habituellement sur les écrans d'ordinateurs.

Dans Zhang16, les auteurs proposent d'utiliser une nouvelle transformée, appelée *Adaptive Color-Space Transform* (ACT), destinée à explorer les redondances entre les canaux de couleurs dans une

image. Cette transformée est utilisée en amont de la DCT et permet de prédire les valeurs de *Chroma* en fonction des valeurs de *Luma*.

Les types de codage adaptés aux contenus de type SCI ont permis d'obtenir des gains de débit de l'ordre de 5-10% et de temps d'encodage de l'ordre de 20%.

De manière générale, l'état de l'art montre que ce type d'optimisation arrive à apporter un certain gain en débit, qui reste néanmoins marginal par rapport à nos objectifs de compression. Cela démontre la nécessité de repenser entièrement les méthodes, pour assurer une vraie rupture technologique. Ces dernières années, les nouvelles techniques émergentes fondées sur le paradigme de l'apprentissage statistique et notamment par réseaux de neurones profonds (*deep learning*), ont trouvé leur application dans le domaine de la compression d'images, en lui apportant un nouveau souffle.

2.7.3 Approches par machine learning (ML)

Les techniques de *machine learning* sont susceptibles d'être appliquées à presque toutes les étapes d'un schéma de compression classique. De manière générale, elles offrent l'avantage de pouvoir adapter ces différentes étapes aux propriétés spécifiques de certaines catégories/familles de contenus.

2.7.3.1 Application à la prédiction

La prédiction est une étape clef de toute chaîne de compression d'images, qui ne cesse d'être enrichie et diversifiée par les différents schémas et formats de codage.

Des travaux récemment réalisés dans ce domaine sont rapportés dans [Li18]. Ici, les auteurs utilisent un réseau *fully-connected* afin d'effectuer la prédiction sur un bloc courant à partir d'une, ou plusieurs lignes de référence adjacentes au bloc à prédire. L'entrainement a été réalisé en utilisant le corpus de données *New York city library images* [Wilson14]. Les résultats obtenus sont illustrés Figure 2.25.

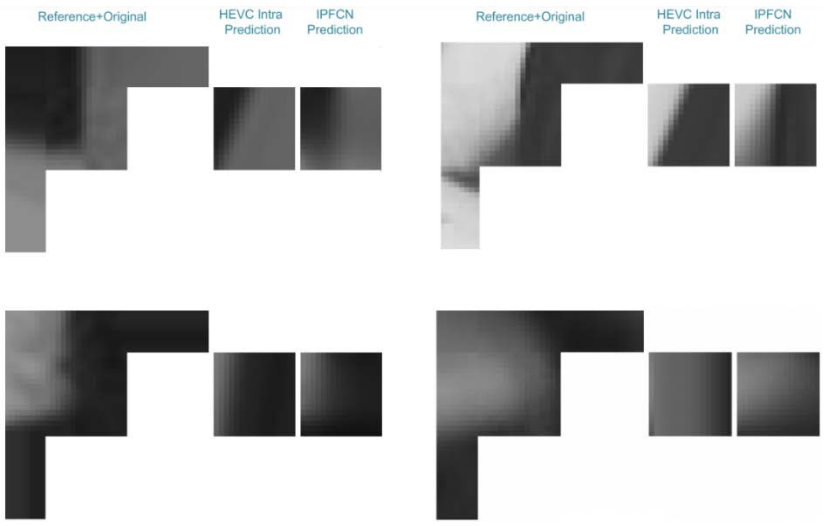


Figure 2.25 : Comparaison des prédictions obtenues par *machine learning* (gauche) et HEVC (droite) (Source [Li18])

Nous pouvons observer que la prédiction obtenue est très fine et bien plus précise que celle obtenue par celle obtenue par HEVC. Cela est également confirmé numériquement : en utilisant une métrique MSE, les résultats obtenus sont de 2 à 6 fois plus faibles que ceux obtenus par HEVC. Néanmoins, les gains en termes de réduction de débit rapportés par les auteurs restent relativement modestes (environ 4% par rapport à HEVC). Il serait intéressant d'explorer à l'avenir comment ce gain en

qualité de la prédiction pourrait être exploité pour obtenir, globalement, une compression plus efficace.

Des travaux visant à améliorer la prédiction inter-frames, et plus précisément à la réduction du coût de transmission des vecteurs de prédiction de mouvements, utilisant des réseaux de neurones ont également fait leurs preuves [Birman20]. Ici, une réduction de 34% du nombre de bits nécessaires à la transmission de ces vecteurs est obtenue sans perte de qualité. Bien que ne concernant pas directement les images, explorer la méthodologie présentée et l'adapter à la prédiction intra pourrait s'avérer être une piste intéressante.

2.7.3.2 Apprentissage des transformées

Dans [Sezer08], un nouvel ensemble de transformées, remplaçant la DCT, est obtenu à l'aide d'une technique conjointe de classification et calcul de transformées orthonormales optimales. A partir d'un premier ensemble de classes, les transformées optimales sont déterminées pour chaque classe. Ces transformées servent ensuite à recalculer l'assignation des blocs en classes. L'algorithme est itéré jusqu'à convergence.

L'entrainement est réalisé à partir de 1 million de blocs de taille 8×8 issus d'images naturelles. Ces transformées s'appuient sur l'exploitation des singularités directionnelles des objets de l'image. Les résultats de ces transformées se montrent assez concluants, notamment pour des images contenant de fortes structures géométriques comme des immeubles par exemple. Les gains en débit à PNSR constant sont ainsi d'environ 10% par rapport à une transformée DCT à un niveau de débit de 0,5 bpp.

Ce même principe est repris et dans [Puri16], où les auteurs proposent de l'appliquer notamment pour optimiser la chaîne de compression HEVC. Les transformées ainsi obtenues conduisent à un gain en débit d'environ 2%.

2.8 Discussion

Dans ce chapitre, nous avons énoncé les différents codecs les plus présents et les plus intéressants, du moins dans le cadre de nos recherches.

A la vue des résultats de nos tests, nous avons décidé de nous concentrer sur l'utilisation du codec BPG car il permet d'atteindre les débits les plus bas, tout en offrant la meilleure qualité dans ces zones.

Dans l'optique de travailler sur une amélioration de BPG, nous avons fait état des recherches réalisées sur ce sujet. Malheureusement il s'avère que la plupart des travaux concernant le BPG/HEVC visent à une amélioration de la complexité et non du gain de débit. Or, étant donné que notre principal défi concerne la réduction du débit, nous ne sommes pas parvenus à trouver des pistes de recherches suffisamment prometteuses.

En effet, les quelques techniques proposées dans la littérature n'offrent que des gains trop faibles par rapport aux objectifs que nous nous fixons.

Pour pallier ces difficultés, nous avons décidé d'aborder une autre piste qui est celle du *machine learning*, notamment appliqué aux traitements des images et plus précisément à des tâches pouvant s'accommoder à des problématiques de compression. Ces nouvelles méthodologies sont abordées en détails dans le chapitre suivant.

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 3. Compression par techniques d'apprentissage profond

Résumé. Dans ce chapitre, nous introduisons dans un premier temps les notions fondamentales de *machine/deep learning* ainsi que leurs applications diverses. Nous abordons également la déclinaison de ces techniques dans le cas d'applications liées aux images, en particulier à base de réseaux de neurones convolutifs (CNN – *Convolutional Neural Networks*). Un aspect important que nous analysons en détail concerne la définition de fonction de pertes, qui constitue un élément clé pour tout problème utilisant des réseaux de neurones et plus particulièrement quand il s'agit de travailler avec la notion de qualité d'images.

Enfin, nous dressons un état de l'art des différentes techniques à base de réseaux de neurones, notamment convolutifs, liées directement à la compression d'images. Dans ce cadre, nous détaillons les méthodes dédiées de super-résolution, de compression de bout en bout par réseaux de neurones ou encore de réduction d'artéfacts de compression.

Une discussion est enfin proposée pour déceler les pistes de développement les plus intéressantes, à retenir dans le cadre de nos travaux.

Mots clés : machine/deep learning, CNN, fonctions de pertes, Super-Résolution, Réduction d'artéfacts de compression, compression bout à bout

Institut Polytechnique de Paris
91120 Palaiseau, France

3.1 Machine learning, deep learning et réseaux de neurones convolutifs

3.1.1 Machine et deep learning

Le *machine learning* (ML) est une catégorie d'algorithmes qui permet aux applications logicielles de devenir plus précises dans la prédiction des résultats, en exploitant des bases de données annotées avec vérité terrain. La promesse de base de l'apprentissage machine est de construire des algorithmes qui peuvent recevoir des données d'entrée et utiliser l'analyse statistique pour prédire une sortie capable de propriétés de généralisation pour de nouvelles données.

Parmi les différents algorithmes de ML, nous distinguons trois familles principales qui sont les techniques d'apprentissage supervisé, les méthodes d'apprentissage non-supervisé ou encore les approches d'apprentissage par renforcement. Dans nos travaux, nous allons nous concentrer exclusivement sur la notion d'apprentissage supervisé.

Dans ce cas, il est nécessaire de disposer d'un corpus d'entraînement avec vérité terrain, constitué d'un ensemble de données étiquetées, de la forme $\Omega = (x_i, l_i)_i$, avec x_i représentant les données (le plus souvent des vecteurs dans une espace multidimensionnel) et $l_i \in \mathcal{L}$ les étiquettes correspondantes. Les étiquettes peuvent correspondre à un ensemble discret $\mathcal{L}$ de catégories prédéfinies (dans le cas notamment des applications de classification) ou à des valeurs scalaires ou vectorielles (typiquement pour des objectifs de régression).

L'objectif est alors de déterminer une fonction de de mappage, capable de prédire pour toute donnée d'entrée x, la valeur $l(x)$ de son étiquette.

Parmi les méthodes traditionnelles de ML, mentionnons les techniques comme les *k-means clustering* [Hartigan79], les *random forests* [Segal04] ou les SVM (*Support Vector Machines*) [Hearst98]. Ces techniques ont été appliquées avec succès dans les années 2000 pour différentes applications nécessitant la mise en œuvre d'une classification supervisée.

Ces dernières années, la disponibilité de bases des données avec vérité terrain de plus en plus importantes ainsi que le progrès des technologies de calcul massivement parallèle (cartes GPU) grand public ont permis l'émergence d'une nouvelle méthodologie de classification, fondée notamment sur les réseaux de neurones profonds.

Institut Polytechnique de Paris
91120 Palaiseau, France

3.1.2 Apprentissage par réseaux de neurones profonds (*deep learning*)

Les réseaux de neurones profonds revisitent le principe de la classification supervisée, en s'appuyant sur une méthodologie inspirée des systèmes biologiques. L'élément de base est le concept de neurone (Figure 3.1), qui effectue une opération de classification binaire.

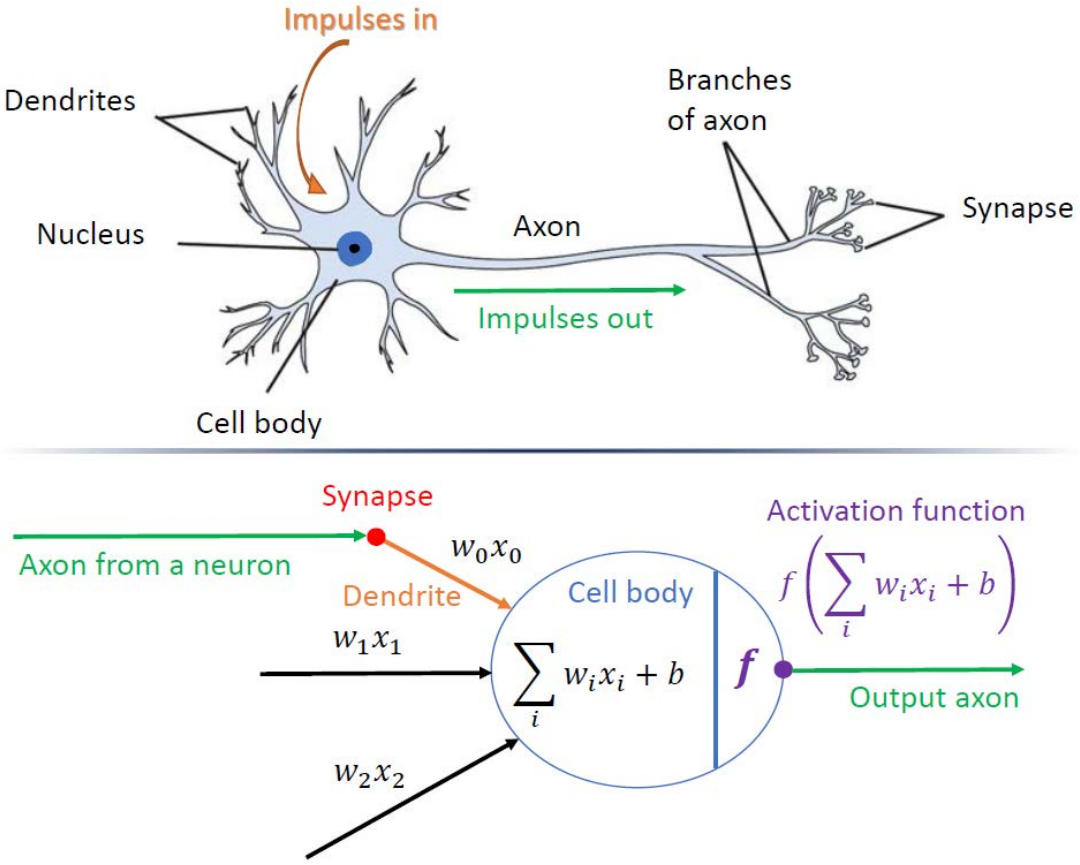


Figure 3.1 : Concept de neurone avec principe associé de classification linéaire par neurone

La fonction de classification binaire est définie dans ce cas comme décrit dans l'équation suivante :

$$l(x) = \begin{cases} 1 \; si \; f(w \cdot x + b) > 0 \\ 0 \qquad sinon \end{cases}, \tag{3.1}$$

où x désigne le vecteur d'entrée, w le vecteur de paramètres associé aux dendrites du neurone, b un terme de biais et f une fonction d'activation.

En pratique, la sortie du neurone peut être gouvernée par différentes fonctions d'activation, avec propriétés variées. Parmi les plus populaires, citons les fonctions de type sigmoïde, tangente hyperbolique (tanh), ReLU (*Rectified Linear Unit*), *Maxout*, *Leaky* ReLU ou encore ELU (Figure 3.2).

Institut Polytechnique de Paris
91120 Palaiseau, France

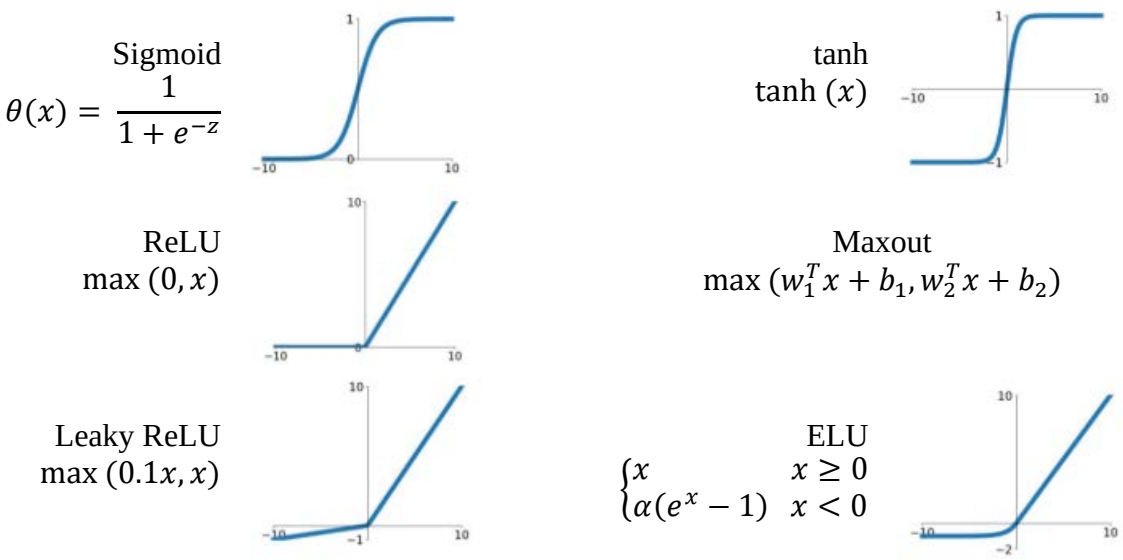


Figure 3.2 : Fonctions d'activation

Un réseau de neurones est alors constitué par un ensemble de neurones interconnectés, structuré en couches interconnectées successives. Une architecture typique inclut, entre les couches d'entrée (dont la dimension correspond à a dimension des vecteurs à classifier) et de sortie (dont le nombre est lié à celui des classes considérées dans le processus de classification) un certain nombre de couches intermédiaires, appelées couches cachées (*hidden layers*). Classiquement, chaque neurone d'une couche est connecté à tous les neurones des couches précédente et suivantes. On parle alors d'un réseau de type *fully-connected*. La Figure 3.3 illustre l'exemple d'un réseau de neurones à une seule couche intermédiaire.

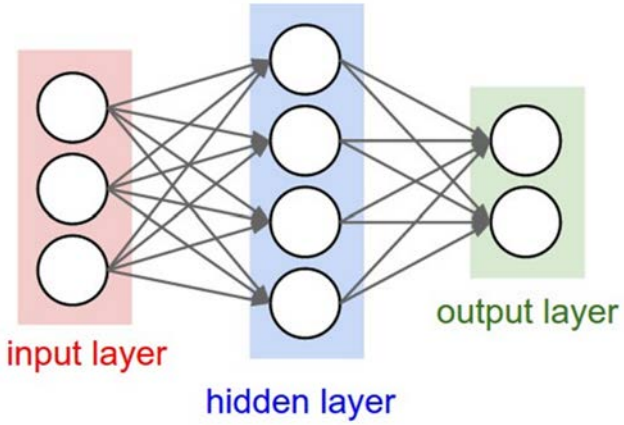


Figure 3.3 : Architecture d'un réseau à une couche intermédiaire

Notons que, selon le théorème d'universalité [Baker98], un réseau de neurones à une seule couche est théoriquement capable d'approcher toute fonction continue, à condition que le nombre de neurones présents dans la couche intermédiaire soit suffisamment grand.

Cependant, ce résultat théorique est peu exploitable en pratique, car le nombre de neurones intermédiaires nécessaires peut être très important, rendant ainsi impossible l'apprentissage du réseau. La solution consiste à considérer des architectures composées d'un nombre plus important de couches intermédiaires. On parle alors de réseaux profonds. Un exemple d'architecture profonde est illustré Figure 3.4.

Institut Polytechnique de Paris
91120 Palaiseau, France

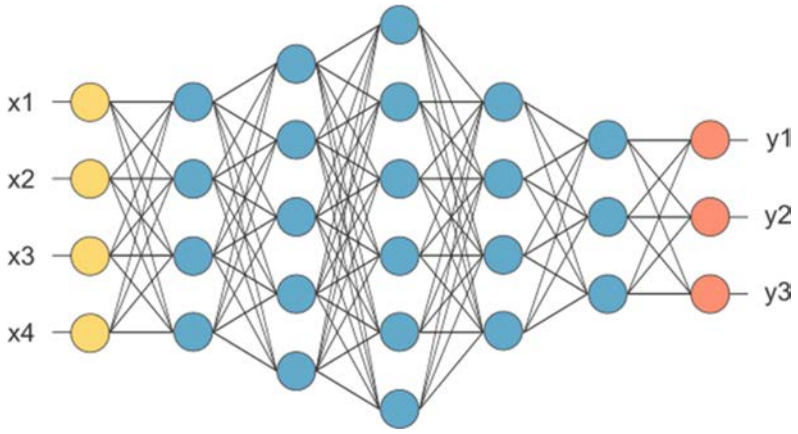


Figure 3.4 : Exemple d'un réseau de neurones dit « profond »

L'objectif du processus d'apprentissage est de déterminer l'ensemble des paramètres du réseau (poids et biais) minimisant une *fonction de pertes*, qui mesure la différence globale entre la sortie du réseau et la vérité terrain. Comme fonctions de perte, on peut utiliser notamment des distances L_1 ou L_2. Une discussion plus approfondie sur les fonctions des pertes, qui sont déterminantes pour les performances du réseau, est proposée dans la Section 3.1.4.

Pour apprendre ces paramètres, la technique utilisé est dérivée de la méthode de descente de gradient et consiste à retro-propager les gradients à travers les différentes couches du réseau, à partir de la couche de sortie. On pare alors du *back-propagation*. Notons toutefois que le nombre de paramètres à déterminer peut être très important en pratique (millions ou même milliards), pour des problèmes de classification complexes. Pour éviter les problèmes de blocage dans des minima locaux propres à toute descente de gradient, il est alors nécessaire de considérer des versions plus avancées.

Parmi les différents algorithmes d'optimisation, les plus utilisés à ce jour citons :

- la décroissance du taux d'apprentissage au fur et à mesure des itérations [Smith17]
- la descente de gradient avec inertie [Qian99]
- l'algorithme RMSprop [Tieleman12]
- l'algorithme ADAM (mélange de descente de gradient avec inertie et de RMSprop) [Kingma14].

Notons qu'à ce jour, en raison de ses performances, ADAM semble être de loin l'algorithme le plus utilisé dans la plupart des réseaux de neurones modernes existants.

Un autre problème qui peut apparaître lors de l'entrainement du réseau est lié au phénomène de sur-apprentissage (*overfitting*). Ainsi, du fait du nombre très important de connexions neuronales, et donc de paramètres d'un réseau *fully-connected* profond, une prédiction extrêmement fidèle aux données d'entrainement peut être obtenue.

L'*overfitting* se trouve être le cas où le réseau fournit une prédiction extrêmement fidèle aux données d'entrainement, au point de ne plus être en mesure de réaliser quelconque généralisation pour des données extérieures, non prises en compte dans le processus d'apprentissage. A l'autre extrême, se trouve le phénomène complémentaire de sous-apprentissage (*underfitting*), dans quel cas, le réseau n'est pas capable de fournir de classification correcte même pour les données de l'ensemble d'apprentissage. Ces deux phénomènes sont illustrés Figure 3.5.

Institut Polytechnique de Paris
91120 Palaiseau, France

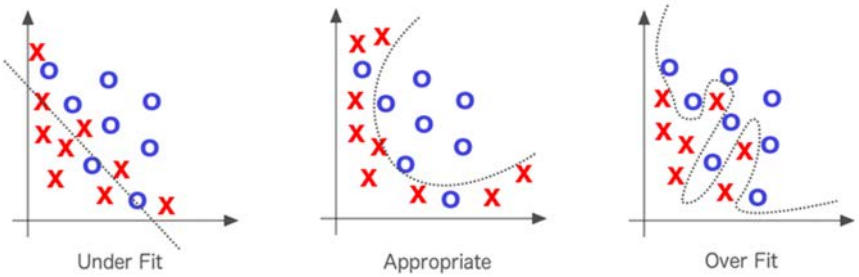


Figure 3.5 : Exemples de résultats d'*underfitting*, d'*overfitting* et d'entrainement adapté

Deux principales solutions sont employées pour réguler l'*overfitting*. La première étant la régularisation L_2, appliquée sur l'ensemble des paramètres, qui consiste en l'ajout d'un terme à la fonction de pertes visant à pénaliser la complexité du modèle.

La deuxième concerne le *dropout* (ou abandon), qui consiste en la désactivation d'un sous-ensemble de neurones de manière aléatoire à chaque itération. Cela permet d'à la fois limiter la capacité du réseau et d'encourager chaque neurone à apprendre plus indépendamment par rapport aux autres.

Soulignons aussi que la meilleure méthode pour éviter l'*overfitting* est de disposer d'un corpus d'entrainement à la fois suffisamment grand et varié.

Enfin, une dernière notion utile et fréquemment utilisée dans le processus d'apprentissage est celle de *batch*.

Afin d'améliorer les résultats des prédictions lors d'un entrainement, il est possible de fournir plusieurs éléments en entrée d'un réseau à la fois pour une même itération. Il est par exemple possible de fournir 8 images à un réseau pour une même itération. On parle alors d'un entrainement avec un *batch* de taille 8.

L'utilisation de *batchs* présente un double intérêt : elle permet d'une part de réduire le temps d'entrainement, et d'autre part d'accroître la capacité de généralisation du réseau, en considérant plusieurs éléments hétérogènes simultanément.

Les principales limitations sous-jacentes sont d'ordre matériel, car l'utilisation de mémoire vive d'un réseau par itération est multipliée par la taille du batch considéré.

3.1.3 Application aux images : les Réseaux de Neurones Convolutifs

Dans le cas du traitement d'image, les réseaux de neurones tels que présentés présentent un inconvénient majeur. En effet, une image se trouve être composée d'une multitude de vecteurs, associés aux pixels la constituant, et donc de données. Cela entraine une très grande masse de données, qui nécessite d'être gérée et traitée. Il serait inconcevable de construire des réseaux de neurones de type *fully-connected* pour des images dont les résolutions couramment utilisées aujourd'hui dépassent souvent (1000 ×1000) pixels.

Or, bien que les images possèdent effectivement beaucoup de données, elles ont la particularité d'être fortement corrélées localement. Cela signifie que, comme les pixels proches d'une image ont une forte corrélation spatiale, il n'est ni nécessaire ni pertinent de les traiter comme des informations indépendamment les unes des autres.

Institut Polytechnique de Paris
91120 Palaiseau, France

Pour cela, un nouveau type d'opérateur de pondération a été adopté afin de construire des réseaux adaptés à ce type de données : il s'agit de la convolution.

La convolution entre une image $f(x,y)$ avec un noyau $w(x,y)$ est définie comme décrit dans l'équation (3.2):

$$(f * w)(x,y) = \sum_{(u,v)} w(u,v) \cdot f(x-u, y-v) \tag{3.2}$$

L'opération de convolution est définie par le noyau w associé. Le plus souvent, ces noyaux correspondent à des filtres à réponse impulsionnelle finie et présentent donc un support compact. La taille du support de ces filtres définit leur champ de vision.

Ce mécanisme s'inspire notamment du système visuel humain, où différents récepteurs réalisent ce type d'opération avec des champs de vision correspondants plus ou moins larges.

Une opération de convolution appliqué à une image est illustrée Figure 3.6.

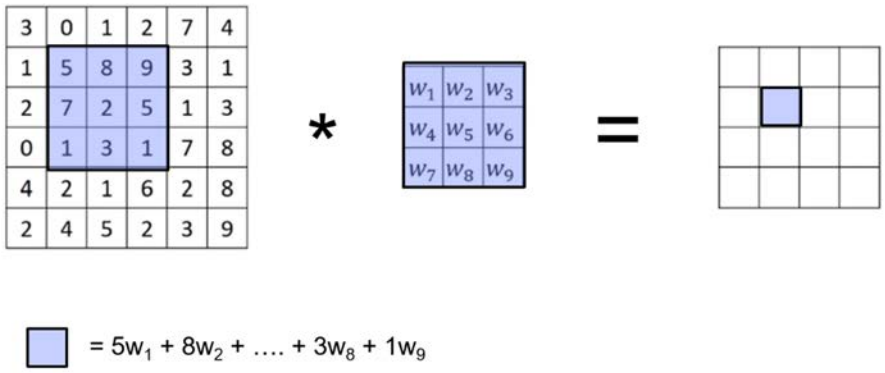


Figure 3.6 : Exemple d'une convolution avec un noyau 3×3

Les réseaux de neurones faisant appel aux convolutions sont appelés réseaux de neurones convolutifs ou CNN (*Convolutional Neural Networks*).

Au niveau de chaque couche de neurones, un ensemble de convolutions est appliquée. Les noyaux de convolution deviennent dans ce cas les paramètres à apprendre par le réseau. Cela conduit à une structure tridimensionnelle, où à chaque couche est appliqué un certain nombre de filtres de convolution.

La Figure 3.7 illustre un réseau neuronal complètement convolutif.

Institut Polytechnique de Paris
91120 Palaiseau, France

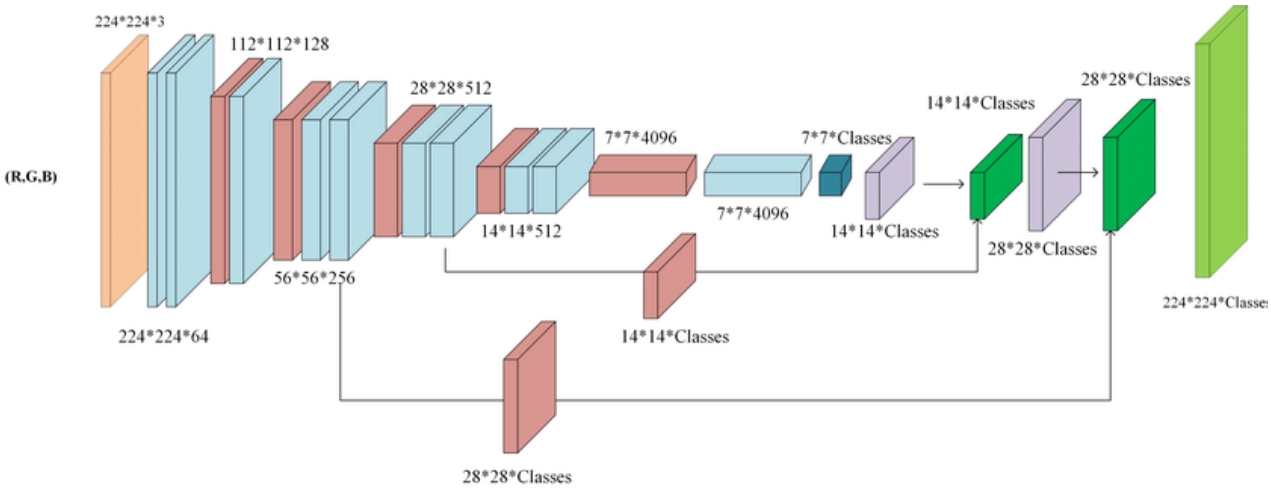


Figure 3.7 : Exemple d'un réseau de neurones complétement convolutif [Piramanayagam18]

Le réseau prend ici la forme d'une structure 3D, de type ($H \times W \times n$), où H et W représentent les dimensions du signal image et n le nombre de filtres de convolution considérés.

Notons également que la réduction de dimension entre couches successives peut être contrôlée par un paramètre supplémentaire, appelé *stride*, qui définit tout simplement le pas de translation des filtres sur le signal image à traiter.

Ainsi, une couche de convolution sera renseignée dans le manuscrit sous la forme d'un triplet (k, n, s), pour désigner un noyau de convolution de taille ($k \times k$), une profondeur de n filtres et un *stride* de s pixels. Par exemple, la notation (*k3*, *n64*, *s1*) désigne une couche de 64 filtres de convolution de taille (3×3), appliqués avec un *stride* de 1.

Enfin, notons que des couches « classiques » dites denses, dans le sens où elles sont complètement connectées, peuvent être ajoutées en fin de chaîne de traitement, notamment pour des objectifs de classification, lorsque la taille des couches, et donc le nombre de paramètres associés, deviennent suffisamment faibles. Ce mécanisme est illustré Figure 3.8.

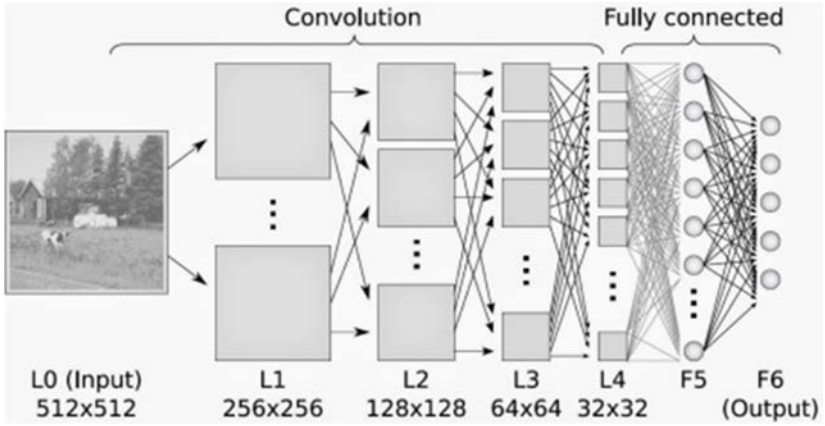


Figure 3.8 : Exemple d'un réseau CNN enrichi avec couches denses

Institut Polytechnique de Paris
91120 Palaiseau, France

3.1.4 Fonctions de pertes pour la compression d'image

La fonction de pertes considérée est essentielle pour tout processus d'apprentissage, déterminant les propriétés et le comportement du réseau. Des choix adaptés sont par conséquent nécessaires afin d'assurer que le réseau fournit les résultats attendus.

En compression d'image, et même plus généralement dans tout traitement affectant la qualité d'une image, les fonctions de pertes ont pour but d'orienter le réseau vers la reconstruction d'une image le plus fidèle possible à l'image faisant office de vérité-terrain. En d'autres termes, une fonction de pertes va permettre au réseau de rendre la meilleure qualité d'image possible, par rapport à une référence qui est celle définie par la vérité terrain.

Dans le cadre de nos travaux de recherche, il s'est avéré que la fonction de pertes devient un élément majeur. Pour cette raison, détaillons à présent les différents types de fonctions de pertes que nous rencontrerons tout au long de ce manuscrit.

3.1.4.1 Fonctions de pertes « orientées pixel »

Les fonctions de pertes dites « orientées pixel » sont des fonctions qui vont comparer directement l'image reconstruite, au niveau de la valeur des pixels, à la vérité terrain. Pour cela, le plus souvent, des fonctions globales mesurant les distances entre contenu reconstruit et image cible sont utilisées. Parmi les distances le plus couramment utilisées, mentionnons en premier lieu la distance Euclidienne, ou norme L_2, qui apparait comme la solution la plus intuitive. Cela conduit à la fonction de pertes d'Erreur Quadratique Moyenne (EQM), définie comme décrit dans l'équation (3.3).

$$\mathcal{L}oss_{\mathrm{L2}} = \sum_{i,j=1}^{H,W} (y_{verité\ terrain}(i,j) - y_{prédit}(i,j))^2 \tag{3.3}$$

La distance de Manhattan, ou norme L_1, a également très vite été adoptée en raison de sa complexité de calcul moins importante. Il apparait également que l'utilisation de la norme L_1 comme fonction de pertes apporte de meilleurs résultats en termes de PSNR [Zhao16] que l'EQM. Elle est définie comme décrit dans l'équation (3.4).

$$\mathcal{L}oss_{\mathrm{L1}} = \sum_{i=1}^{n} \left| y_{verité\ terrain} - y_{prédit} \right| \tag{3.4}$$

Les limitations des fonctions de perte comme l'EQM ou L_1 viennent du caractère trop global dont les différences entre pixels individuels sont comptabilisées et prises en compte par ces mesures. Cela peut conduire, dans des certains cas à des résultats aberrants. Par exemple, si l'on change la valeur de chaque pixel d'une image de 5 unités, le résultat apparaitra comme identique à la vérité terrain pour l'œil humain, mais sera fortement pénalisé par ce type de fonction.

Dans [Wang04], les auteurs mettent en avant que l'œil humain soit bien adapté pour extraire les informations structurelles d'une scène. Ainsi, il serait bien plus sensible à des défauts de structure au sein d'une image altérée plutôt qu'à des variations des valeurs des pixels individuels. Pour pallier cet inconvénient, l'auteur propose un système d'évaluation alternatif au PSNR, permettant une bien meilleure fidélité quant à l'évaluation de la qualité d'une image par le système de vision humain, nommé Structural SIMilarity (SSIM).

La fonction de pertes est alors définie comme décrit dans l'équation (3.5) suivante :

$$\mathcal{L}oss_{\text{SSIM}} = 1 - SSIM(y_{verité\ terrain}, y_{prédit}) \tag{3.5}$$

Une variante multi-échelle est également introduite, appelée Multi Scale-SSIM (MS-SSIM), qui combine les calculs des SSIM de plusieurs versions d'une image à multiples échelles. Cette métrique offre alors une meilleure robustesse que le SSIM simple. Notons que les métriques SSIM et MS-SSIM sont aujourd'hui les mesures de similarité les plus utilisées dans l'état de l'art.

3.1.4.2 Fonctions de pertes perceptuelles

Les fonctions de pertes orientées pixels permettent d'évaluer rapidement les résultats, et fournissent ainsi un moyen simple et direct de pouvoir effectuer des comparaisons entre différentes méthodes pour des objectifs de *benchmarking*. Malheureusement, même si l'introduction du SSIM a permis une première avancée, en prenant en compte d'une manière plus fine certaines caractéristiques du système visuel humain, ces méthodes d'évaluation dites « objective » sont encore très loin de pouvoir retranscrire la qualité d'une image du point de vue de la perception humaine.

De plus, le fait d'utiliser le même type de fonction à la fois pour l'entrainement et pour l'évaluation fait apparaître de forts biais. En effet, l'objectif d'un réseau de neurones est de minimiser sa fonction de pertes de manière incrémentale à travers des nombres très importants d'itérations, et ce, totalement indépendamment du contenu qu'il traite. Ceci est encore plus vrai avec la multiplication de la puissance des moyens de calcul grand public, permettant des réseaux de neurones à plusieurs millions, voire milliards de paramètres. De par ce fait, il est de plus en plus aisé de minimiser une fonction de pertes, si bien qu'il est très aisé d'obtenir d'excellents scores de PSNR ou de SSIM en les considérant comme fonctions de pertes et ce, indépendamment de la qualité de l'image. Cette idée est renforcée par l'étude menée dans [Chikkerur11]. Ainsi, on peut statuer qu'en continuant d'utiliser ces fonctions de pertes, nous nous retrouvons à atteindre un plafond ne permettant plus d'améliorer encore la qualité de contenus reconstruits.

Face à ce défi, un nouveau type de fonction de pertes, dites fonctions de perte « perceptuelle » est introduit dans [Johnson16]. Ici, les auteurs proposent deux applications pour valider ces fonctions de pertes, qui sont la super-résolution et le transfert de style. Néanmoins, ce type de perte peut s'appliquer également à d'autres problèmes de traitement d'images prenant en considération la qualité visuelle du résultat.

Le principe sous-jacent est le suivant : au lieu de calculer les fonctions de pertes directement dans le domaine spatial (soit le domaine des pixels de l'image), on réalise leur évaluation dans le domaine des paramètres du réseau. Pour des applications de reconstruction d'images par exemple, de très bons résultats sont obtenus en utilisant comme réseau de référence un réseau de classification d'images bien entrainé. L'exemple le plus utilisé, en raison de sa disponibilité et de son efficacité, se trouve être le réseau VGG (*Visual Geometry Group*) [Simonyan14], illustré Figure 3.9.

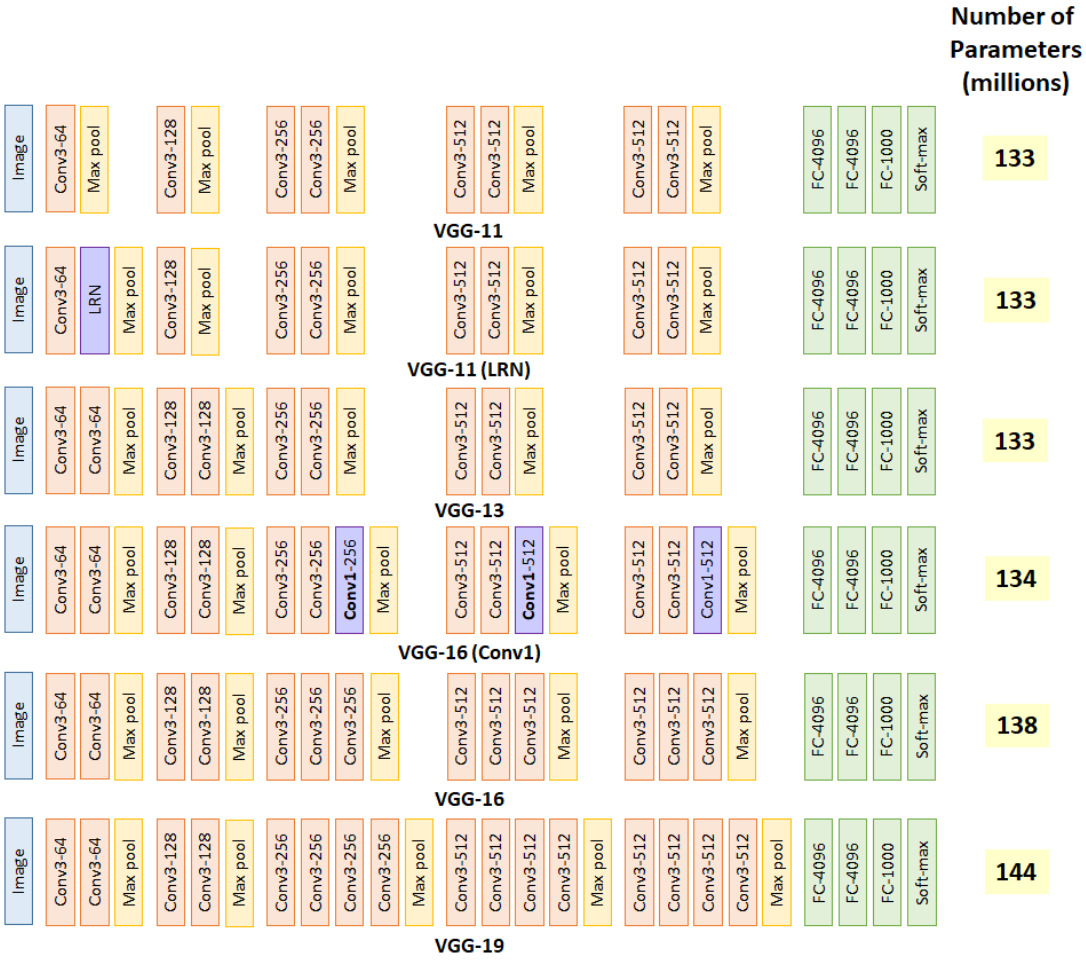


Figure 3.9 : Différentes architectures de réseaux VGG [Kumar19]

Le réseau VGG19 a été initialement développé dans le cadre de la compétition de classification d'images ImageNet ILSVRC-2014 [Russakovsky15]. Avec un taux d'erreur de 7.3%, le réseau VGG avait ainsi réalisé la percée de l'année 2015, en termes de classification d'images. VGG19 présente l'avantage d'être simple, rapide et facile à comprendre/manipuler. Il s'appuie sur une architecture à 5 couches de convolution, suivies par des couches denses, qui sont chargées du travail final de classification. De leur côté, les couches de convolution réalisent des tâches « d'extraction de caractéristiques », déterminant avec une complexité progressive les patterns/formes géométriques apparaissant au sein des images. De cette manière, les couches de convolution d'un réseau VGG pré-entrainé agissent comme un puissant extracteur de formes. Dans les faits, plus la couche considérée est profonde, plus les caractéristiques extraites sont génériques et de haut-niveau.

Ainsi, il est possible de définir une fonction de pertes perceptuelle, en calculant la distance (euclidienne) entre les caractéristiques obtenues au niveau des différentes couches d'un réseau VGG étalon, après activation, et celle obtenues par le réseau à entrainer.

Les résultats montrent que ce type de fonction de pertes a pour caractéristique de permettre des résultats correspondant bien plus à la perception humaine que ses homologues orientés pixel [Johnson16].

Après avoir passé en revue ces différentes notions de base de machine/*deep learning*, intéressons-nous à présent aux méthodologies directement liées à nos travaux de recherche, qui peuvent jouer notamment un rôle important dans la chaîne de compression d'images. Dans ce cadre, une première piste de développement concerne les techniques de super-résolution, décrites dans la section suivante.

3.2 Super-Résolution

Les techniques de super-résolution (SR) visent à obtenir des résolutions supérieures, aussi fidèles que possible à l'image originale, à partir de leur contrepartie en basse résolution (BR). La super-résolution se trouve être un domaine large et varié, avec de nombreuses applications concrètes comme l'imagerie médicale [Huang17] ou la vidéosurveillance [Uiboupin16]. En plus de réaliser sa fonction primaire, la SR permet aussi d'aider à obtenir de meilleurs résultats dans d'autres domaines du traitement d'images [Dai16] comme la reconnaissance d'objets [Haris18a] ou de gestes [Zhang18]. Naturellement, ces techniques peuvent trouver un intérêt certain dans une chaîne de compression, permettant de réduire considérablement la quantité d'information à transmettre.

La Figure 3.10 présente un premier exemple d'image obtenue par super-résolution en la comparant, avec la même augmentation de résolution, avec une image obtenue par la technique classique des *Nearest Neighbors* (Plus Proches Voisins) [Friedman75].

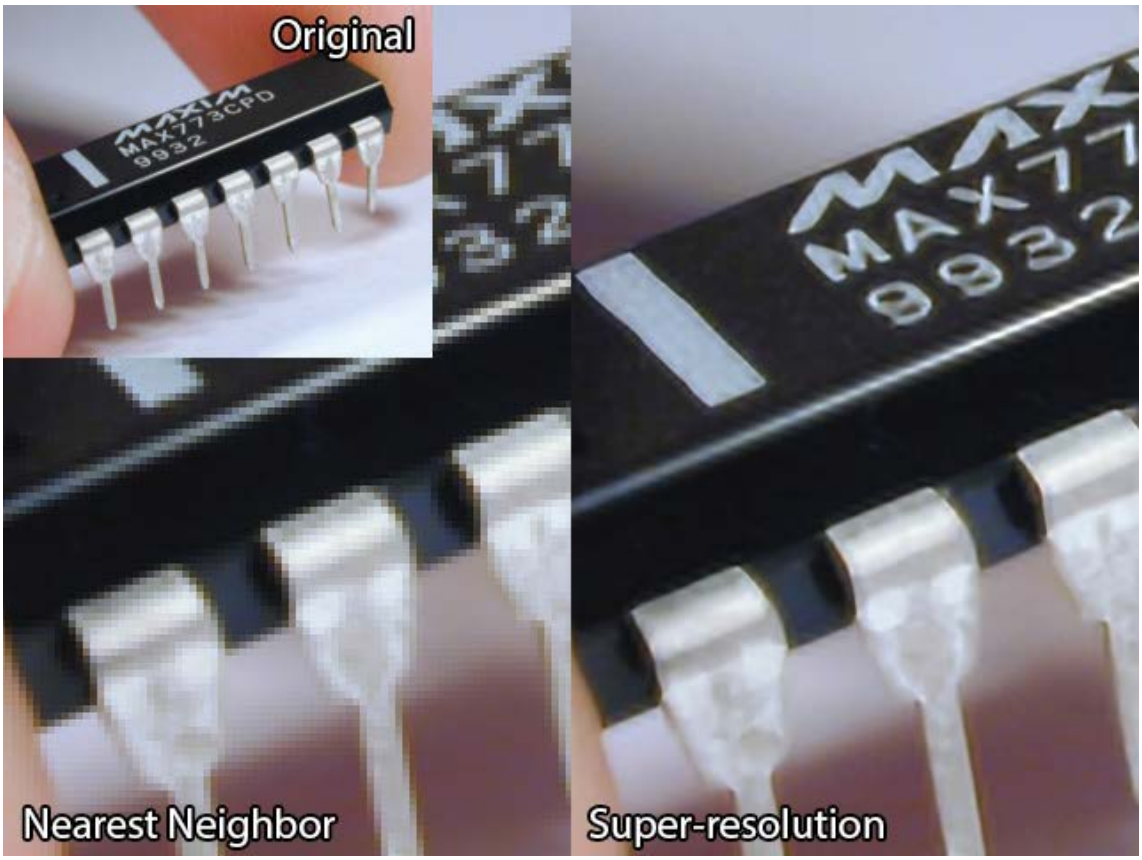


Figure 3.10 : Comparaison visuelle entre agrandissement par SR (gauche) et *Nearest Neighbors* (droite) à augmentations de résolution égales

Au fil des années, de nombreuses méthodes algorithmiques ont été proposées pour répondre aux problèmes de SR. Ces méthodes peuvent être basées sur de la prédiction [Irani91], sur l'utilisation des arrêtes [Freedman11], ou même être statistiques [Sun08] ou encore par représentations parcimonieuses [Yang10].

Parmi ces méthodes, citons tout d'abord la très populaire *Anchored Neighborhood Regression* (ANR) [Timofte13]. ANR est une méthode de SR basée sur l'utilisation de dictionnaires pré-entrainés. Ici, les auteurs proposent deux techniques permettant d'obtenir une SR aussi performante, voire légèrement supérieure à ses prédécesseurs par rapport au score PSNR, tout en réduisant significativement la complexité de calcul (jusqu'à 110 fois plus rapides par rapport à [Yang10]). Une version simplifiée mais offrant de résultats légèrement inférieurs, appelée *Global Regression* est aussi présentée. Enfin, une version améliorée d'ANR proposée par les mêmes auteurs et nommée A+ [Timofte14] a été également proposée.

Des avancées significatives ont été réalisées dans ce domaine avec la démocratisation du *deep learning*, et notamment des réseaux de neurones convolutifs (CNN). L'utilisation de ce type de réseaux a permis d'obtenir directement des performances surpassant celles des techniques précédentes et ce, sur un panel de benchmark varié.

Parmi les premiers travaux utilisant les CNN pour des tâches de SR, il convient de citer l'approche SRCNN introduite par Dong *et al*. [Dong15]. La technique proposée inclue les 4 étapes successives suivantes.

- Après un pré-traitement de l'image basse résolution, une première image haute-résolution est obtenue par interpolation bicubique.
- Extraction des patchs qui sont représentés par des vecteurs de grande dimension. Un ensemble cartes de paramètres dont le nombre est égal à la dimensionnalité des vecteurs est ainsi obtenu.
- Mappage non linéaire de chaque vecteur, donnant alors un autre ensemble de cartes.
- Reconstruction de l'image par agrégation des patchs obtenus, représentant l'image finale en haute résolution (HR), supposée fidèle à la vérité terrain.

Ce processus est illustré Figure 3.11.

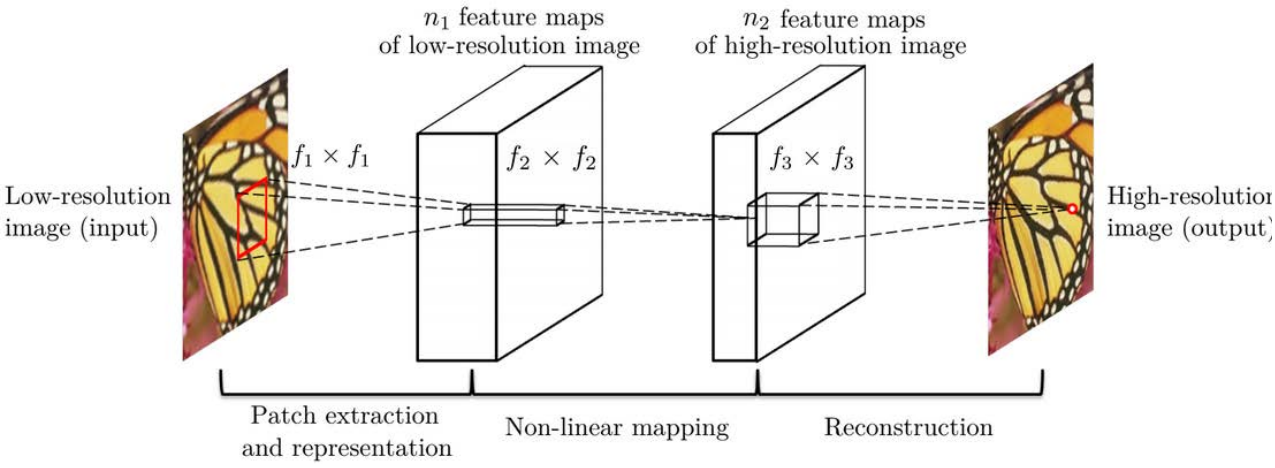


Figure 3.11 : Architecture du réseau SRCNN [Dong15]

Il est également montré dans le même article que les méthodes de SR qui s'appuient sur des représentations parcimonieuses (*sparse coding*) peuvent être reformulées sous forme de CNN, permettant ainsi une meilleure compréhension du problème et offrant des pistes prometteuses pour créer de nouveaux types de SRCNN.

Concernant les conditions d'entrainement, une fonction de pertes de type Erreur Quadratique Moyenne (EQM) est utilisée, optimisée par une « décente de gradient stochastique » classique.

Les expérimentations ont été conduites sur deux bases d'image, l'une étant très petite, contenant seulement 91 images, l'autre étant constituée de 395909 images en provenance du corpus ILSVRC 2013 ImageNet [Russakovsky15]. Ces images ont ensuite été subdivisées afin d'obtenir des corpus d'entrainement de respectivement 24800 et plus de 5 millions de sous-images.

Les corpus de validation Set5 et Set14 ont également été utilisés et bien que les résolutions de ces images soient différentes, les tendances restent similaires. Ainsi, les résultats obtenus montrent une amélioration significative par rapport aux techniques de SR traditionnelles, aussi bien en termes de score PSNR que de point de vue visuel, et ce malgré un réseau relativement réduit (8032 paramètres).

La technique RAISR (*Rapide and Accurate Image Super Resolution*) [Romano17], actuellement utilisée dans la version mobile de Google+ [Nach17], se trouve être plus rapide et présente des résultats visuellement comparables.

L'approche RAISR met en place quasiment le même procédé que la méthode SRCNN, avec pour corpus d'entrainement 10000 images de bannières publicitaires. La fonction de pertes utilisée n'est pas exactement la MSE mais reprend un principe quasi identique.

Le RAISR offre une bonne adaptabilité à l'image obtenue, en appliquant une subdivision des *patchs* de l'image en agrégats et un filtrage différent à chacun de ces agrégats. Afin de conserver une complexité réduite, l'opération est effectuée par l'utilisation d'une procédure de hachage basée sur un critère de gradient plutôt que par des algorithmes de regroupement trop complexes (K-Means [Jeong10], GMM [Sandeep16] …).

Enfin, une approche très intéressante concernant la base d'entraînement est décrite. Ici, les auteurs avaient aussi pour but d'obtenir une réduction des artéfacts de compression JPEG, en complément des objectifs de super-résolution. Pour cela, d'une part, les images en basse-résolution ont été préalablement compressées en JPEG avec des facteurs de qualité allant de 85 à 100. D'autre part, le corpus en haute-résolution a été « augmenté » en y ajoutant un filtrage destiné à améliorer les détails des images. La méthode utilisée pour cette « augmentation » est une fonction basée sur les Différences de Gaussiennes [Bundy84].

Pour résumer, l'approche RAISR fait intervenir les 3 étapes suivantes :

- Agrandissement par interpolation classique (*e.g.*, bilinéaire),
- Utilisation de la table de hachage contenant un ensemble de filtres pré-appris et dont la clé est une fonction de gradient,
- Mappage des filtres avec les *patchs* présents dans l'agrandissement par interpolation.

Institut Polytechnique de Paris
91120 Palaiseau, France

De cette manière, en superposant un entraînement de réseau de neurones peu coûteux en temps et des opérations algorithmiques peu complexes, l'approche RAISR permet d'obtenir des résultats comparables au SRCNN tout en étant 2 fois plus rapide.

Nous pouvons remarquer que dans ces deux cas, le travail d'augmentation de la résolution de l'image est réalisé à l'aide d'une opération d'interpolation algorithmique classique. Ainsi, la super-résolution se trouve ici être principalement une opération de filtrage visant à corriger les artéfacts obtenus après augmentation de la résolution. C'est précisément ce principe de filtrage/correction d'artéfacts qui sera au cœur de nos recherches et que nous détaillerons dans les futures sections de ce manuscrit.

Il existe à ce jour 4 grands types de méthodes de SR, qui s'appuient sur du *deep learning*. Ces méthodes diffèrent essentiellement par le moment où l'augmentation de résolution est réalisée dans le processus d'entraînement (Figure 3.12).

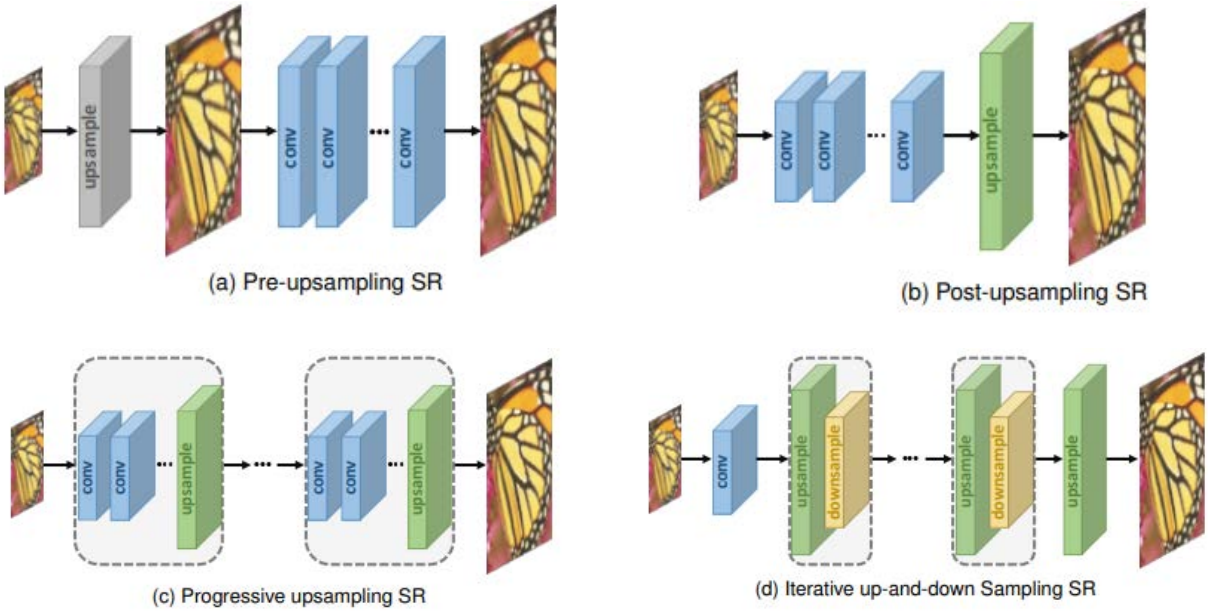


Figure 3.12 : Illustration des 4 grandes méthodes de SR (Source : [Wang20])

Afin de réduire la complexité afférante à l'opération de mappage sur une image en haute-résolution, le modèle post-augmentation (Figure 3.12.b) a été proposé. Dans ce modèle, la plupart des opérations sont réalisées sur l'image en basse-résolution et une couche est ajoutée à la fin afin d'augmenter la résolution de l'image. Ce principe a été premièrement exposé dans le travail de Dong *et al.*[Dong16], proposant une version rapide du SRCNN et a été ultérieurement adopté comme le modèle le plus populaire de SR. Les raisons majeures de ce choix résultent de la réduction très importante de coût de traitement, puisque dans ce cas les convolutions sont réalisées dans un espace plus réduit, correspondant aux images en basse-résolution.

L'inconvénient majeur de la méthode « post-agrandissement » est que l'agrandissement est toujours réalisé en une seule étape, ce qui implique des difficultés de performances pour les grands facteurs d'agrandissement comme ×4 ou ×8 par exemple. De plus, comme une seule opération d'angrandissement est réalisée, cela implique la nécessité de disposer de réseaux différents pour divers facteurs d'agrandissement.

Pour palier à ces contraintes, un modèle nommé « Laplacien pyramid SR network » (LapSRN) [Lai17], utilisant un agrandissement progressif de l'image au sein du réseau est proposé (Figure 3.12.c). Ici, à l'issue de chaque agrandissement, une image exploitable est créée servant ainsi de base pour le prochain agrandissement. On met ainsi en place un principe de cascade de CNN.

Une autre méthode très similaire nommée Progressive SR (ProSR) [Wang18a] ne se sert pas des images intermédiaires après chaque agrandissement mais seulement des informations principales de ces dernières. Ainsi, il devient possible d'utiliser le même réseau afin d'obtenir des super-résolutions avec des facteurs différents. De plus, par la décomposition d'une tâche lourde qui serait un agrandissement par un grand facteur, en plusieurs tâches moins complexes, de meilleures performances ont pu être obtenues autant en terme de qualité que de rapidité. En revanche, ce type de stratégie se trouve être bien plus complexe quant à la conception du réseau et à l'optimisation de son entraînement.

Le quatrième type de méthode est la SR par agrandissement-rétrécissement intéractif (Figure 3.12.d). L'objectif ici de d'obtenir le plus d'information possible de la dépendance du couple haute et basse résolution de l'image. Ce procédé bien que très prometteur [Haris18b] pose encore de nombreux soucis de conception et reste relativement peu renseigné à ce jour. Néanmoins, en raisons de son grand potentiel et de l'accroissement des moyens de calculs, il pourrait devenir un des prochains standards de la super-résolution par *deep learning*.

La super-résolution, et particulièrement les approches par CNN se trouve être un sujet très vaste et très actif. Pour un état de l'art détaillé de ce domaine citons l'article particulièrement riche proposé par Wang *et al.* [Wang20].

Enfin, ces dernières années, une nouvelle famille d'approches pour la SR a vu le jour. Elle concerne notamment les réseaux de type GAN (*Generative Adversial Networks*). Cette partie sera reprise plus tard dans la Section 4.4, en raison de son lien direct avec nos recherches et nos résultats.

Une autre famille d'approches abandonne le principe de la super-résolution et s'intéresse aux aspects de reconstruction d'image par réduction d'artéfacts.

Institut Polytechnique de Paris
91120 Palaiseau, France

3.3 Réduction d'artéfacts

La compression d'images avec pertes, par sa nature, entraine une distorsion de l'image, plus ou moins perceptible par l'œil humain, après traitement. Ces distorsions sont appelées « artéfacts de compression ». Ces artéfacts sont la conséquence de la suppression, le plus souvent lors de l'étape de quantification, de données utiles de l'image.

Par conséquent, un des objectifs majeurs pour les formats de compression, et plus généralement pour les algorithmes de compression, est de réduire au maximum ces artéfacts, principalement par le biais d'une étape de post- traitement, appliquée une fois l'image décompressée.

Intéressons-nous tout d'abord aux principaux types d'artéfacts que l'on puisse rencontrer.

3.3.1 Types d'artéfacts

Ringing

Les « *ringing artifacts* » sont dus à la suppression des hautes fréquences dans une image lors de l'opération de compression avec pertes. Ces artéfacts sont principalement visibles aux alentours des arêtes et sur les bords d'une image.

Flou

Ces artéfacts ont la même cause que les « *ringing* » mais présentent un aspect différent et ne sont pas localisés aux alentours des arrêtes.

Blocking

Ce type d'artéfacts apparaît lorsqu'une compression utilisant des *transform blocks* est utilisée, comme dans le cas des formats JPEG ou BPG. Dans ce cas, plus la compression est forte, plus ce type d'artéfacts sera visible. La raison principale de ces *blocking artifacts* est due aux différents traitements réalisés sur des blocs adjacents. Par exemple, l'étape de prédiction se trouve en pratique être réalisée dans des sens différents d'un bloc à un autre. Cela entraine une forte discontinuité visuelle entre les deux blocs décodés. Un autre facteur concerne l'étape de quantification qui elle aussi est appliquée individuellement à chaque bloc, avec des paramètres différents. Ces artéfacts sont particulièrement visibles sur les zones lisses d'une image, où il n'y a pas de texture pour atténuer visuellement cette discontinuité.

Institut Polytechnique de Paris
91120 Palaiseau, France

Ces différents types d'artéfacts sont illustrés Figure 3.13

a) Original

b) Artéfacts de « ringing »

c) Artéfacts de flou

d) Artéfacts de « blocking »

e) Combinaison des différents artéfacts de compression énoncés sur une image en couleurs

Figure 3.13 : Différents artéfacts sur une image fortement compressée en JPEG

Institut Polytechnique de Paris
91120 Palaiseau, France

De nombreux travaux ont été réalisés afin de réduire ces artéfacts, et en particulier ceux de *blocking*.

Plusieurs filtres ont été proposés pour la réduction d'artéfacts de compression (RAC) dans le domaine spatial. Par exemple dans [List03] est proposé un filtrage qui deviendra par la suite une partie intégrante du codec vidéo AVC/H.264.

Plus récemment, les techniques d'apprentissage profond ont fait également leur apparition dans le domaine de la réduction d'artéfacts, comme décrit dans la section suivante.

3.3.2 Deep learning et réduction d'artéfacts

Dans le domaine du *deep learning*, les problèmes de RAC et de SR se trouvent être étroitement liés. Les artéfacts de compression, en plus d'être un problème complexe, représentent une contrainte forte pour le bon fonctionnement de la SR. En effet, ces artéfacts peuvent être augmentés par le processus de SR, entrainant ainsi des résultats visuellement désagréables.

Une autre motivation quant à l'utilisation de CNN pour la réduction d'artéfacts est due au fait que, avant ces techniques, la grande majorité des algorithmes étaient seulement dédiés à diminuer les artéfacts de *blocking* et de produire un résultat plus ou moins flou à la place.

Afin de répondre à ces problèmes [Dong15b] propose de directement adapter le SRCNN créant ainsi le réseau dit *Artifacts Reduction*-CNN (AR-CNN). Pour cela, deux étapes principales ont été modifiées par rapport à la version SRCNN initiale. Premièrement, l'étape d'agrandissement par interpolation bi-cubique a été enlevée. Deuxièmement, une couche supplémentaire a été ajoutée afin d'enrichir les caractéristiques extraites par la première. Cette dernière se situe alors entre la première et la deuxième couche du SRCNN, créant ainsi un réseau à 4 couches, illustré Figure 3.14.

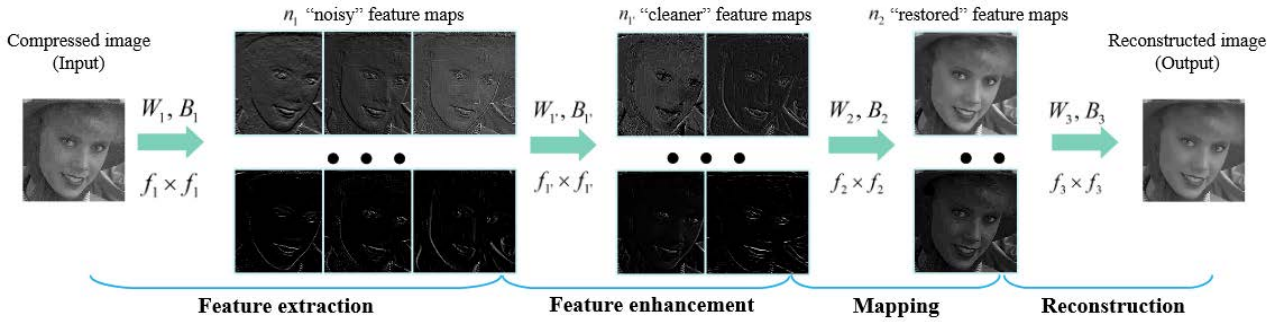


Figure 3.14 : Descriptions du rôle des 4 couches du AR-CNN

Pour l'entrainement du réseau, la fonction de pertes EQM est encore une fois utilisée, avec une optimisation par décente de gradient stochastique. Cette fois-ci, l'entrainement a été réalisé en utilisant des *batchs* de taille 128.

Le corpus d'entrainement utilisé correspond à une adaptation de la base BSDS500 [Set11], conduisant à un total de 400 images, compressées avec le MATLAB JPEG Encoder avec des facteurs de qualité variant entre q = 10 et q =20 (pour deux entrainements différents).

Ces images ont ensuite été subdivisées en 537600 patches de 32×32 pixels. Un élément particulier de la démarche proposée est que seuls les centres des patches, d'une taille de 20×20 pixels sont utilisés pour le calcul de la fonction de pertes.

Des expérimentations de transfert d'apprentissage sont également présentées. Les auteurs proposent trois types différents de transfert.

Le premier consiste à utiliser un réseau entrainé avec un facteur de qualité moins important (q=20) et de se servir des paramètres appris pour obtenir de meilleurs résultats pour un autre entrainement sur des images compressées plus fortement (q=10).

Le deuxième exploite les résultats d'un AR-CNN à 4 couches afin d'entrainer autre AR-CNN mais cette fois avec 5 couches.

Enfin, ils proposent d'utiliser l'entrainement avec compression JPEG afin d'entrainer un modèle comprenant des artéfacts de compression plus complexe, comme la compression Twitter par exemple.

Dans ces trois cas, un gain est obtenu en termes de temps de convergence et de qualité. Néanmoins, ce gain se trouve être très peu visible au regard du PSNR (de l'ordre du 10ème voir du 100ème de dB).

Les résultats obtenus par le AR-CNN se trouvent en revanche être bien meilleurs visuellement, et ce dans tous les cas (*blocking*, *ringing*, JPEG, Twitter), par rapport aux méthodes retenues pour comparaison, soit le Shape Adaptative-DCT [Foi07], le SRCNN classique ainsi qu'une version plus profonde de ce dernier. Ainsi, les auteurs démontrent à la fois la pertinence quant à l'utilisation de CNN pour la réduction d'artéfacts, et à la fois l'intérêt de réaliser des entrainements « progressifs » et d'utiliser ces résultats par transfert d'apprentissage.

Dans [Chen18], la relation entre SR et RAC est encore une fois soulignée. En effet, du fait que la SR d'une image compressée s'avère être une tâche compliquée, la solution intuitive serait de traiter indépendamment les deux problèmes. Néanmoins il se trouve que des détails importants pour la bonne réalisation de la SR peuvent facilement être supprimés lors de la RAC. Pour cette raison, il est préférable de traiter les problèmes conjointement via par exemple un seul et unique réseau de neurones, nommé ici *Compressed Images Super-Resolution Deep*-CNN (CISRDCNN).

Le CISRDCNN est composé de 3 modules distincts, représentant respectivement les modules de dé-*blocking* (DBCNN), d'agrandissement (USCNN) et d'amélioration de la qualité (QECNN). Ici, les auteurs proposent d'entrainer d'abord les 3 modules séparément à leurs tâches spécifiques, puis de réaliser un entrainement supplémentaire d'optimisation sur l'ensemble du réseau.

Pour l'entrainement, un ensemble de 291 images composées de 200 images issues de BSDS500 et 91 images issues de [Yang08]. Afin d'augmenter ce nombre, les auteurs ont eu recours à des techniques d'augmentation de données. Les fonctions de pertes utilisées sont ici des dérivées d'EQM.

La méthode proposée est comparée avec huit autres méthodes, incluant le (Fast)SRCNN, l'interpolation bicubique le A+ et l'ARCNN. Les résultats obtenus apparaissent meilleurs que ceux obtenus via toutes les autres techniques comparées à la fois via les métriques objectives (PSNR, SSIM, IFC) que par rapport aux aspects visuels. Ces expérimentations ont été réalisées sur des images compressées en JPEG, avec 3 facteurs de qualité différents (10, 20, 30). La diversité des facteurs de qualité proposés montre la robustesse de l'approche. Nous pouvons néanmoins noter qu'un entrainement séparé a été réalisé pour chacun des trois facteurs différents. Un même entrainement ne montre donc pas une robustesse sur différents facteurs de qualité.

Ainsi, un pipeline alternatif de compression est proposé, alliant super-résolution et compression par codecs standards. En effet, les images sont stockées à la fois à basse-résolution et compressées et ce, même à très bas débit. La reconstruction se fait alors au moment de la visualisation de l'image, cette dernière étant alors agrandie en plus de recevoir une réduction des artéfacts.

Une technique similaire est proposée dans [Yu18]. Cette fois-ci, les auteurs abordent un problème proche du notre qui est la réduction d'artéfacts de compression BPG. Dans ce cas-ci, le réseau *Enhanced Deep Residual Networks for Super-Resolution* (EDSR) [Lim17] est utilisé comme réseau de base.

Concernant la structure du réseau, certaines modifications ont été apportées au EDSR comme par exemple la suppression des couches de normalisation de *batchs*, qui est remplacée par une normalisation des valeurs des paramètres appris. Le réseau contient *16* blocs résiduels et utilise des patchs de 224×224 pixels. L'entrainement a été réalisé avec le *dataset* DIV2K avec un paramètre d'apprentissage de 10^{-3}.

La fonction de pertes utilisée est une fonction L_1. Cette dernière tend à remplacer la fonction EQM car elle offre une complexité de calcul plus réduite, une meilleure qualité visuelle et surtout un meilleur PSNR. Les résultats obtenus montrent une amélioration par rapport au BPG notamment en PSNR. Nous pouvons toutefois noter que la réduction des artéfacts se trouve être, bien que perceptible, très faible, ce qui souligne la complexité de la tâche.

Une autre approche qui peut s'avérer utile à des objectifs de compression s'appuie sur un principe de re-colorisation.

3.4 Ré-colorisation

Le principe consiste à transmettre uniquement le canal de luminance, ainsi qu'un minimum d'information de couleur, permettant néanmoins de reconstruire une version colorée de l'image au niveau du décodeur. Ce minimum d'information peut correspondre à une carte de segmentation, accompagnée d'une indication de couleur pour chaque objet segmenté.

Parmi les techniques qui s'attaquent à cette problématique, citons les travaux présentés dans [Zhang16]. Ici, les auteurs présentent un réseau à base de CNN, entrainé avec plus d'un million d'images en couleurs provenant du corpus de données ImageNet [Deng09]. Une implantation de ce papier a été utilisée pour tester notre pipeline. Comme pour beaucoup de réseaux actuels, le principe consiste à se servir de paramètres haut-niveaux appris sur des réseaux « de référence », principalement inspirés des techniques de reconnaissance d'images, comme VGG [Simonyan14] ou ResNet [He16]. Dans [Baldassarre17], les auteurs présentent un principe similaire, cette fois-ci en utilisant des caractéristiques extraites du réseau Inception-ResNet [Szegedy17]. L'entraînement est ici réalisé avec environ 60000 images.

Enfin, dans [Baig17], les auteurs soulignent l'intérêt d'une alliance entre la colorisation et la compression en proposant une colorisation avec ou sans informations de couleurs supplémentaires à transférer dans le processus de colorisation.

De manière générale, dans ce chapitre, nous avons fait état de l'application de techniques de *machine/deep-learning* et des CNN aux applications liées à la compression d'images que sont la super-résolution, la réduction d'artéfacts de compression, la compression bout-à-bout ou encore la ré-colorisation. Dans la section suivante, nous allons montrer une première approche visant à combiner certaines de ces techniques afin d'avoir un premier aperçu de la pertinence de l'alliance de ces dernières afin d'établir un nouveau schéma de compression à très bas-débit.

3.5 Premier pipeline envisagé

Une première piste de développement concerne l'optimisation/l'adaptation de la norme BPG. Le schéma envisagé est illustré Figure 3.15.

D'une part, il s'agit d'améliorer les étapes de prédiction et de transformée, en s'appuyant sur les techniques de *deep learning*.

D'autre part, nous proposons d'ajouter des étapes de pré- et post-traitement, visant à réduire au maximum la quantité d'information présente dans l'image. Deux étapes sont notamment envisagées :

- Ré-colorisation : dans ce cas, il s'agit d'identifier un minimum d'information couleur à transmettre, permettant, au niveau du décodeur, d'effectuer un processus de re-colorisation conduisant à une version aussi proche que possible des couleurs initiales.
- Sous-échantillonnage : ici, il s'agit de coder et transmettre des images à basse résolution. La résolution initiale sera reconstruite au niveau du décodeur, à l'aide d'un processus de super-résolution.

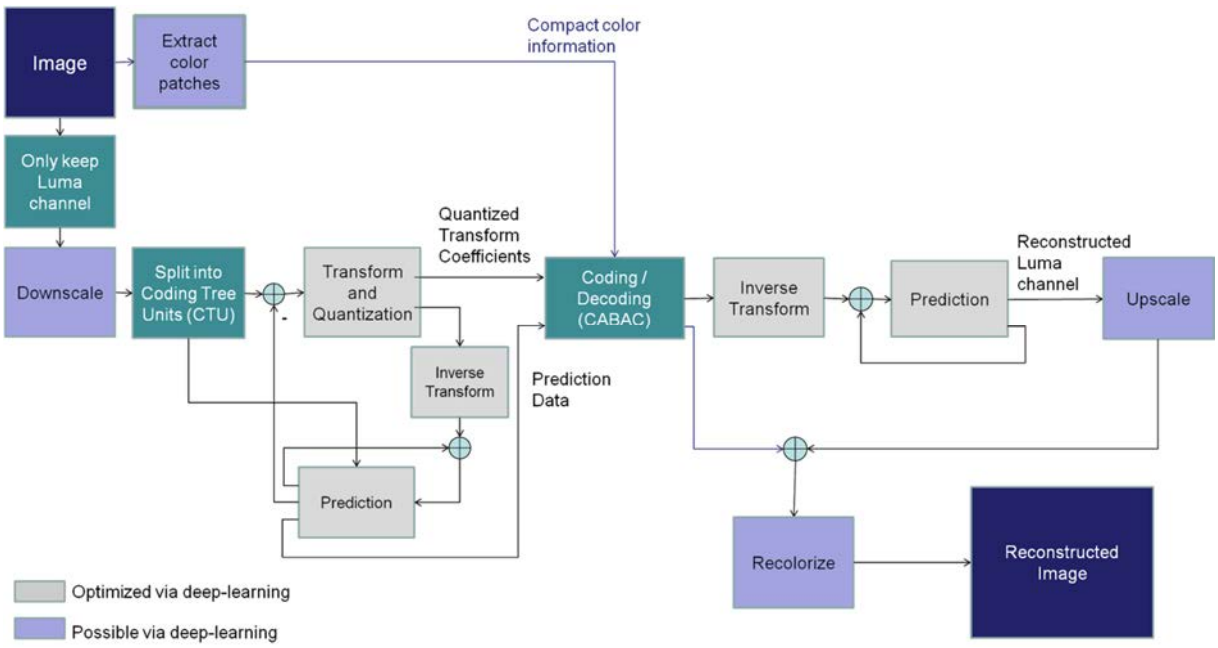


Figure 3.15 : Pipeline BPG avec super-résolution et ré-colorisation

Institut Polytechnique de Paris
91120 Palaiseau, France

3.6 Premiers tests

Les premiers résultats obtenus en appliquant notre pipeline étape par étape, avec des réseaux pré-entraînés, sont illustrés Figure 3.16 (les mêmes images en plus haute résolution sont disponibles en Annexe 1. Comparaison BPG, SRCNN et ré-colorisation, Figure 10.1 et Figure 10.2).

Figure 3.16 : Images avant (à gauche) et après (à droite) passage dans le pipeline BPG+Super Résolution+Colorisation

Nous avons considéré ici l'algorithme de colorisation proposée dans [Zhang16] avec l'implantation [Zhang]. Une carte de segmentation est dans ce cas utilisée comme indice de couleur. Une étape de super-résolution [Ledig17] a été également appliquée, avec un sous-échantillonnage d'un facteur 4 sur les deux dimensions de l'image.

Nous pouvons observer que pour la première image de la Figure 3.16, la colorisation obtenue est totalement erronée, d'une part dans les couleurs restituées, et d'autre part dans le non-respect de la sémantique de l'image (ville confondue avec du feuillage). Cela fait notamment suite à une erreur dans la segmentation de l'image, due au fort lissage résultant de la compression BPG.

Dans la suite, nous avons étudié le comportement de la méthode de super-résolution SRCNN [Dong15] (toujours avec un facteur de sous-échantillonnage de 4 selon les deux directions), en retirant l'étape de colorisation. Les résultats montrent que les images obtenues en combinant SRCNN et compression BPG ne présentent pas une meilleure qualité visuelle, à débit équivalent, que celles obtenues uniquement par BPG, comme illustré Figure 3.17 et Figure 3.18 .

a) b)

Figure 3.17 : Comparaison Lena à 2,1 ko, a) Pipeline BPG+SRCNN, b) BPG seulement

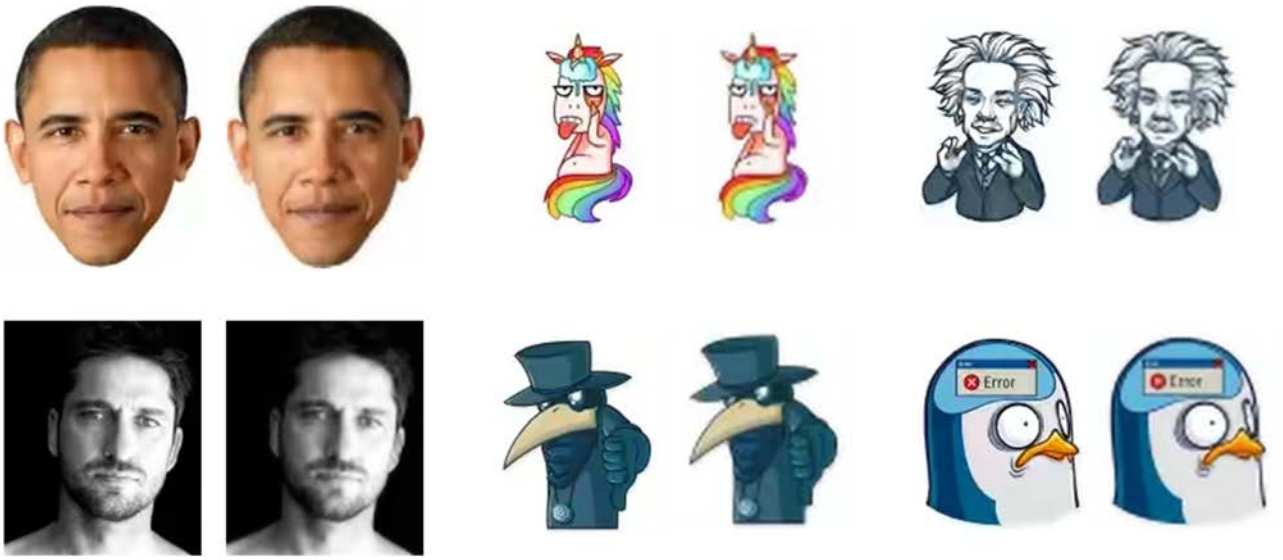


Figure 3.18 : Comparaisons d'images "stickers" à débit égal (entre 1 ko et 1,4 ko),
BPG seul (à gauche) et BPG+SRCNN (à droite)

Ces premiers tests ont été réalisés avec des réseaux pré-entrainés et avec des images ne présentant pas d'artéfacts de compression marqués. Il est tout à fait envisageable d'effectuer l'entraînement aussi bien pour la super-résolution que pour la colorisation, à partir d'images présentant des artéfacts de compression. Cela sera repris dans le plan de développement de la suite de la thèse (*cf.* Section 4.5).

Institut Polytechnique de Paris
91120 Palaiseau, France

Au regard des premiers résultats obtenus et du manque d'articles traitant de la colorisation par réseaux de neurones dans l'état de l'art, nous avons donc décidé d'exclure l'étape de colorisation de notre schéma. Une autre motivation majeure de ce choix est la faiblesse du gain de débit obtenu par transmission d'uniquement le canal de luminance de l'image par rapport aux altérations que peut subir une image lors de la ré-colorisation. Néanmoins, cela n'empêche aucunement d'ajouter cette étape dans de futurs travaux si une application spécifique s'y prête et si de meilleures techniques de ré-colorisation sont mises au point.

Nous avons également écarté l'idée de développer des optimisations pour des étapes spécifiques de la compression BPG. Premièrement, comme énoncé précédemment, d'après l'état de l'art, les gains à espérer en termes de réduction de débit demeurent faibles. Cela est notamment dû au fait que la première version du codec HEVC a été développée pour traiter d'une manière aussi exhaustive que possible tous les cas de figures qui puissent apparaître pour l'optimisation débit/distorsion. Deuxièmement, dans l'optique d'une utilisation générique des méthodes que nous développons, il ne parait pas raisonnable de se passer d'un encodeur et d'un décodeur standard, d'autant plus que le paysage des codecs évolue de plus en plus vite et il devient de plus en plus difficile de prédire la pérennité d'un format.

Pour ces raisons, nous avons orienté nos recherches à partir de ce point vers une méthode plus standard, visant à améliorer des images déjà encodées ; sous forme de pré- et post- traitements.

Avec l'utilisation de réseaux CNN classiques, l'idée de développer une méthode générique quel que soit le codec utilisé et quel que soit le type d'image ne semble pas raisonnable. Néanmoins un nouveau type de réseaux offrant de bien meilleures performances pour la reconstruction d'images est récemment apparu : il s'agit des réseaux GAN (*Generative Adversarial Networks),* présentés au Chapitre suivant.

.

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 4. Réseaux GAN et compression : état de l'art

Résumé. Les réseaux de type *Generative Adversarial Networks* (GAN) sont un type d'architecture de réseaux de neurones offrant des performances, pour des tâches notamment de reconstruction d'image dans des contextes d'application aussi variés que la compression, la super-résolution, la réduction d'artéfacts ou encore le transfert de style, dépassant complètement celles obtenues via les réseaux de neurones traditionnels.

Dans le cadre de nos travaux, nous avons adopté ce type d'approche pour répondre à deux objectifs distincts que sont la super-résolution et à la réduction d'artéfacts. Après un bref exposé des principes fondamentaux des réseaux GAN, nous étudions leur apport, qui s'avère majeur, dans le domaine de la super-résolution, de la réduction d'artéfacts de compression et plus généralement, dans la compression d'images, à travers le concept de compression générative. Les méthodes les plus représentatives de l'état de l'art sont notamment passées ici en revue, avec principe de fonctionnent et analyse des avantages et limitations. Une attention particulière est dédiée aux différentes fonctions de perte utilisées ainsi qu'à leur impact sur les performances du réseau.

Mots clés : réseaux GAN, super-résolution, réduction d'artéfacts de compression, compression générative.

Institut Polytechnique de Paris
91120 Palaiseau, France

4.1 Réseaux GAN (*Generative Adversarial Networks*) : définition et principe

En 2014, Ian Goodfellow propose un nouveau type de réseau de neurones, dit réseau antagonistes génératifs (GAN - *Generative Adversarial Nets*) [Goodfellow14]**.** Cette idée fut vite appréciée, notamment par Yann LeCun, directeur de la recherche IA chez Facebook pionnier du domaine de l'apprentissage neuronal profond, la décrivant comme « *L'idée la plus intéressante de ces 10 dernières années dans le domaine du machine learning* » [LeCun16]. En pratique, ce nouveau type de réseau a permis d'ouvrir une nouvelle ère dans le domaine de la compression d'images.

Le principe d'un réseau de type GAN, illustré Figure 4.1, est le suivant. Deux différents réseaux fonctionnent en parallèle et sont mis en concurrence. Un premier est le *générateur*, qui a pour mission de générer, à partir d'un bruit aléatoire fourni comme entrée, une « fausse » image, qui est sensée approcher au maximum les images de l'ensemble d'apprentissage considéré. Le générateur vise ainsi à maximiser la probabilité que l'image synthétisée appartienne à l'ensemble des données d'apprentissage. Le second réseau, appelé *discriminateur*, a pour mission d'estimer la probabilité d'une image fournie en entrée d'appartenir à l'ensemble d'apprentissage. Son objectif est de minimiser la probabilité d'appartenance à l'ensemble d'apprentissage pour les images synthétisées par le générateur. Intuitivement, nous avons donc d'un côté, un expert (le discriminateur) qui s'entraîne à distinguer le « vrai » du « faux », et de l'autre un faussaire (le générateur) qui s'entraîne à « duper » l'expert.

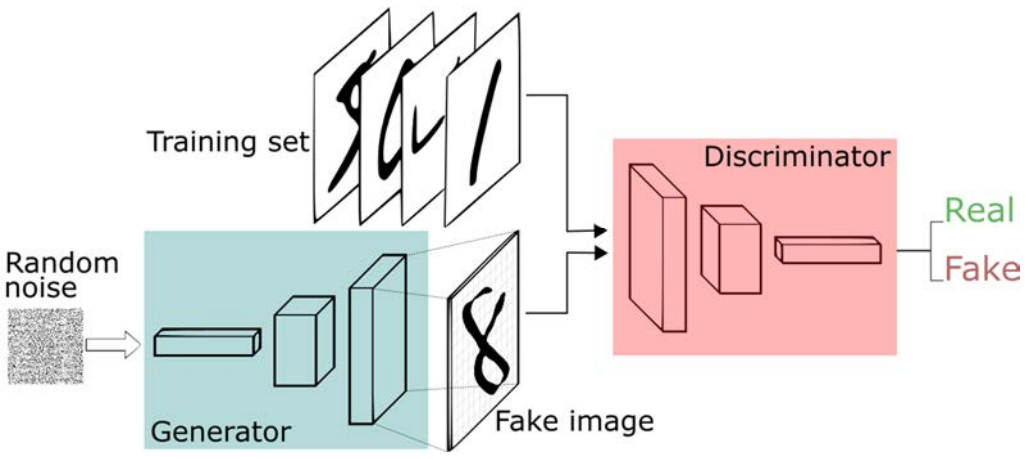


Figure 4.1 : Schéma de principe d'un GAN (source [Wiki18])

Les deux réseaux sont entraînés simultanément sur un même ensemble d'apprentissage, en utilisant les techniques usuelles de retro-propagation. Par ailleurs, un des enjeux majeurs des réseaux de type GAN concerne notamment les aspects de convergence du processus d'apprentissage, mais quelques solutions sont proposées dans [Goodfellow16], [Mescheder18].

Dans sa version originale, le générateur du réseau GAN produit des images à partir d'un bruit aléatoire. Mais il est tout à fait possible de l'aider, en lui fournissant en entrée des informations liées à l'image que l'on souhaité générer.

Analysons donc à présent comment les réseaux de type GAN peuvent être utilisés pour des objectifs de compression d'images.

4.2 Compression générative par réseaux GAN

Dans [Rippel17], une première utilisation d'un réseau GAN à des objectifs de compression d'image est introduite.

Ici, les auteurs proposent un modèle (Figure 4.2) en trois étapes : (1) extraction de caractéristiques, (2) quantification et compression des caractéristiques et (3) reconstruction par GAN.

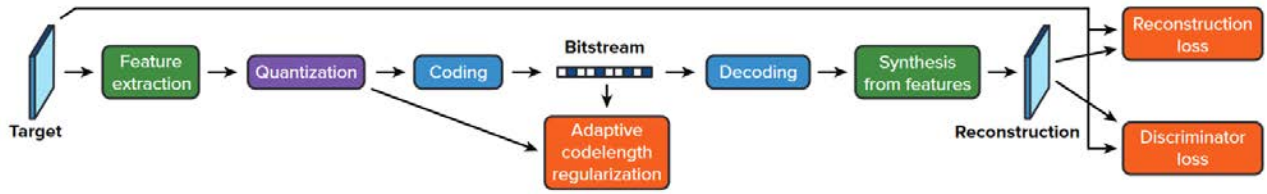


Figure 4.2 : Schéma général de la compression proposée dans [Rippel17]

Le principe utilisé pour l'extraction des caractéristiques s'inspire de l'utilisation d'une transformée en ondelettes par analyse multi-résolution. Pour cela, plusieurs résolutions d'image sont fournies en entrée du réseau et soumises à une série de convolutions. Il en résulte un ensemble de paramètres pour chaque résolution, représentés par des tenseurs, qui sont ensuite regroupés et moyennés par alignement inter-résolution sous la forme d'un unique tenseur. Ces étapes sont illustrées Figure 4.3.

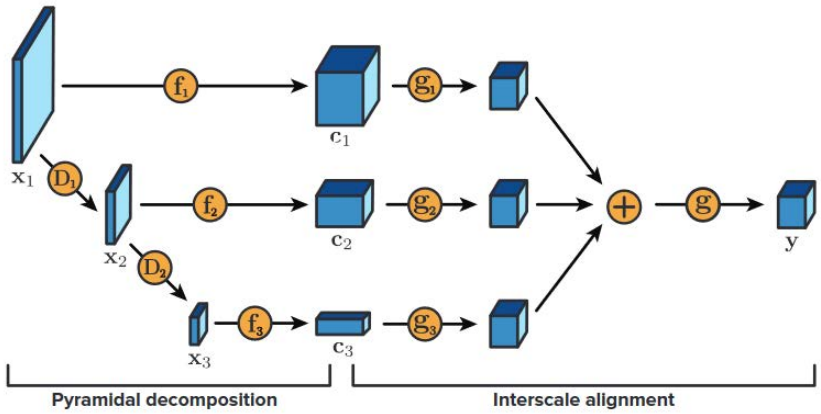


Figure 4.3:Illustration de la décomposition pyramidale et de son encodage par *interscale alignment* utilisé dans [Rippel17]

Enfin, ce tenseur, représentant un ensemble compact de caractéristiques multi-résolution de l'image est ensuite quantifié et codé entropiquement. Un problème est ici souligné par les auteurs car, dans ce principe de compression, le débit de l'image compressée est fixé à l'avance, quelle que soit la complexité de l'image et des textures qui y sont présentes. Les auteurs mettent alors en évidence qu'un codage entropique classique ne suffit pas pour ce type d'application. Pour répondre à cette contrainte, un nouveau principe de régularisation adaptative de taille de code (ACR – *Adaptive Code Regularization*) est introduit, afin de pouvoir agrandir la taille du code pour les patterns complexes et la réduire pour les plus simples.

Concernant la reconstruction, un entrainement utilisant une fonction de pertes SSIM est exploitée, pour un réseau GAN basique de type *vanilla* [Goodfellow14]. Un corpus comprenant des *patchs* de taille 128×128 pixels, extraits de manières aléatoire du Yahoo Flickr Creative Commons 100 Million *dataset* [Mao15] est utilisé pour l'entrainement, le nombre de *patchs* utilisé n'étant pas précisé.

Les résultats obtenus par cette technique appelée WaveOne surpasse à la fois les codecs WebP et JPEG2000 d'un point de vue à la fois visuel et objectif (métrique SSIM). De plus, l'approche permet des performances temps-réel, seulement quelques dizaines de ms étant nécessaires pour effectuer le codage. Un surpassement du BPG est également énoncé sans pour le moins être effectivement démontré dans l'article. Une comparaison entre différents formats et l'approche WaveOne est illustrée Figure 4.4.

Figure 4.4 : Comparaison visuelle entre JPEG, JPEG2000, WebP et WaveOne à débits équivalents (Source : [Rippel17])

Une approche suivant un principe similaire, illustrée Figure 4.5, est présentée dans [Agustsson19]. Des différences majeures avec [Rippel17] sont toutefois à noter. Tout d'abord, une couche de quantification (qui malheureusement n'est pas détaillée) est utilisée après le réseau en charge de l'encodage.

Une autre différence majeure concerne l'utilisation d'une fonction de pertes perceptuelle, définie comme une somme pondérée d'une fonction EQM et d'une perte VGG.

Ici, les auteurs proposent deux modes distincts de compression. Le premier, nommé Compression Générative (CG) effectue une compression de bout-à-bout à partir d'un model pré-entrainé sans information supplémentaire relative au contenu à traiter.

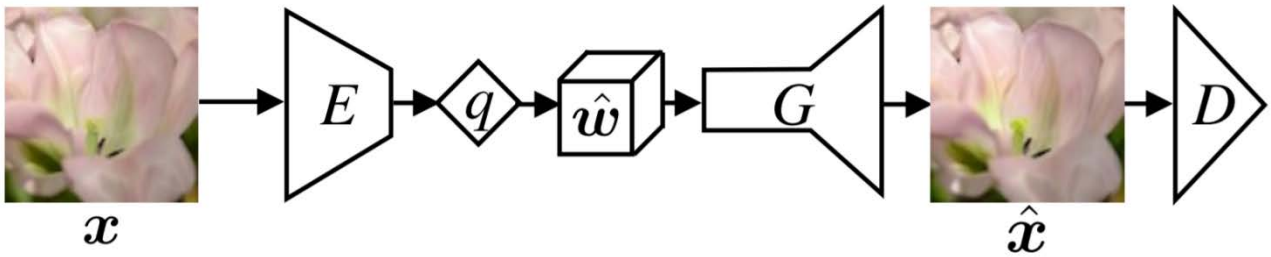


Figure 4.5 : Architecture globale de la Compression Générative proposée dans [Agustsson19]

Institut Polytechnique de Paris
91120 Palaiseau, France

La deuxième est la Compression Générative Sélective (CS) (Figure 4.6). Le principe ici consiste à conserver les parties que l'on souhaite préserver aussi fidèlement que possible via une carte de segmentation, et de reconstruire le reste de l'image par un réseau GAN. D'un point de vue application, il est suggéré que l'utilisateur sélectionne ou dessine les parties à « conserver » avant de lancer la compression.

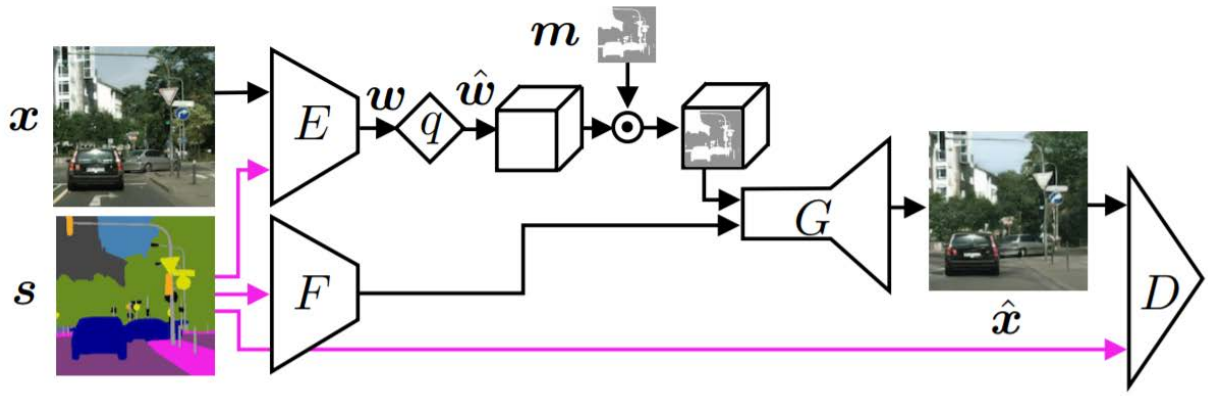


Figure 4.6 : Schéma illustrant la Compression Générative Sélective de [Agustsson19]. Cette est obtenue en ajoutant une carte de segmentation comme contrainte à la CG simple

Pour la CG, le réseau a été entrainé avec 188k images venant du Open Images *dataset* [Kuznetsova20]. Concernant la CS, comme des images appariées avec une carte de segmentation sont nécessaires, le *dataset* Cityscapes [Cordts16] a été utilisé pour l'entrainement.

Les résultats obtenus outrepassent clairement les codecs standards d'un point de vue visuel. Une comparaison avec un autre réseau réalisant un type de compression similaire mais utilisant une fonction de pertes de type EQM/SSIM est également réalisé.

4.3 GAN et réduction d'artéfacts

Les réseaux GAN ont également été appliqués à des fins de RAC. Dans [Galteri17], les auteurs proposent d'utiliser un GAN, possédant une architecture très proche de celle du SRGAN (*cf.* Section 4.4), afin de réduire les artéfacts de compression JPEG. Pour accélérer l'entrainement, l'image est préalablement sous-échantillonnée d'un facteur deux avant d'être passée dans les blocs résiduels et enfin remise à sa résolution d'origine en fin de processus. L'entrainement est réalisé sur des *batchs* de 16 images représentant des *patchs* de 128×128 pixels extrais du corpus d'entrainement MSCOCO [Lin14]. Ce corpus inclue 328k images contenant 91 types d'objets reconnaissables par un enfant âgé de 4 ans. Ces *patchs* sont compressés avec le MATLAB JPEG à différents facteurs de qualités.

Institut Polytechnique de Paris
91120 Palaiseau, France

Une fonction de pertes perceptuelle est utilisée pour obtenir de meilleures performances. Ici, les auteurs soulignent encore une fois la pertinence de ce type de fonction de pertes par rapport aux pertes orientée PSNR et SSIM. Pour démontrer cela, le même réseau est entrainé avec 3 fonctions de pertes différentes afin de confronter les résultats. Trois évaluations différentes sont également réalisées. La première est une évaluation objective s'appuyant sur les scores PSNR et SSIM. Sans surprise, les meilleurs résultats dans ces catégories sont obtenus par le réseau entrainé respectivement avec les fonctions de pertes EQM et SSIM. La deuxième est une évaluation subjective répondant à la norme ITU-R BT.500-13, plus précisément dans la configuration DSIS (*Double-Stimulus Impairment Scale*). Elle a été réalisée sur un panel de 10 évaluateurs chargés d'évaluer les images présentées sur un une échelle continue de 0 à 100, le nombre d'images évaluées, extraites du corpus BSD500 [Martin01] étant de 50. Les résultats obtenus montrent une nette supériorité du modèle entrainé avec une fonction de pertes perceptuelle, tout en présentant un écart-type inférieur.

Enfin, une dernière évaluation très intéressante est proposée. Le principe consiste en l'évaluation de la capacité d'un réseau tierce, ici le *Faster* R-CNN [Ren15], à détecter des objets (avions, chiens, canapés...) présents dans les images traitées. Dans ce cas, le réseau entrainé avec une perte perceptuelle présente de bien meilleurs résultats que ses concurrents pour des contenus fortement compressés. Néanmoins, il est important de noter que plus l'image à tester est de bonne qualité, donc avec un nombre réduit d'artéfacts de compression marqués, moins la différence est notable.

4.4 Super-résolution avec réseaux GAN

Dans ce cas, l'entrée du générateur va représenter une version basse résolution de l'image à traiter. La compression par GAN a notamment pu connaître un grand essor grâce à des travaux proposant des nouvelles fonctions de perte pour la super-résolution et le transfert de style [Johnson16]. Ces fonctions de pertes sont ici testées en utilisant des fonctions de base de VGG, et offrent des résultats bien plus fidèles du point de vue de la perception humaine que les métriques/fonctions de pertes utilisées précédemment. Bien que n'étant pas initialement créées pour des réseaux GAN, ces fonctions de perte se trouvent en être une partie intégrante d'un point de vue pratique et notamment pour la super-résolution.

Une première approche de super-résolution à base de réseaux GAN, dite SRGAN, est proposée dans [Ledig17]. Ici, les auteurs introduisent un réseau réalisant une super-résolution d'un facteur d'agrandissement ×4.

Notons que plusieurs propositions majeures et déterminantes ont été posées. Tout d'abord, il est souligné que, pour des applications de SR **photo-réalistes**, utiliser une fonction de pertes de type EQM est loin d'être la meilleure solution. Plus généralement, une métrique objective de type PSNR ne permet pas de capturer la qualité perceptuelle d'un résultat. Les auteurs proposent donc une nouvelle fonction de pertes dite « perceptuelle », plus optimisée pour les applications photo-réalistes, notamment basée sur les travaux présentés dans [Johnson16].

Ensuite, la pertinence de l'utilisation d'un réseau de type GAN, ici basée sur une architecture ResNet, à des applications de SR est démontrée.

Enfin, une vaste évaluation subjective a été réalisée afin d'apporter une preuve des assertions précédentes.

Institut Polytechnique de Paris
91120 Palaiseau, France

La structure du réseau SRGAN est illustrée Figure 4.7.

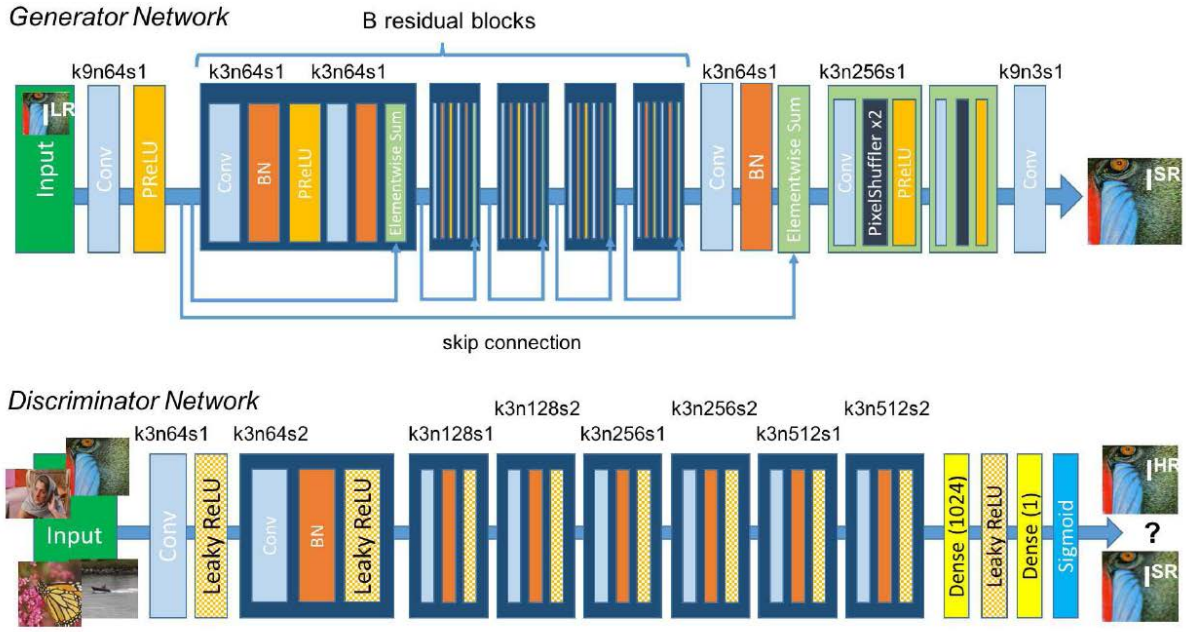


Figure 4.7 : Schéma détaillé de l'architecture du générateur et du discriminateurs du SRGAN (source [Ledig17])

Le générateur possède une architecture très profonde avec un nombre donné de blocs résiduels, ayant une structure s'inspirant des travaux de [Gross16]. Ces blocs résiduels sont donc composés d'une couche de convolution ayant 64 petit noyaux de taille 3×3, qui est suivie d'une couche de normalisation de *batch* puis enfin d'une activation par ParametricReLU. Nous pouvons noter qu'une structure de type « post-agrandissement » est ici proposée, et que l'agrandissement est réalisé par deux couches de convolution sub-pixeliques successives comme proposé dans [Shi16].

Le discriminateur possède une architecture avec 8 couches de convolution avec un nombre de noyaux de taille 3×3 allant de 64 à 512 en étant multiplié par 2 à chaque fois, comme cela est architecturé dans le réseau VGG. Ici une fonction de type LeakyReLU est utilisée pour l'activation. Après la succession de convolutions, deux couches denses sont appliquées, puis une fonction d'activation sigmoïde afin d'obtenir la décision finale du réseau.

La fonction de pertes est un élément majeur dans le bon fonctionnement d'un réseau de neurones. A notre avis, la majeure contribution de cet article concerne notamment la définition d'un nouveau type de fonction de pertes dite perceptuelle. Cette fonction intègre deux composantes distinctes, une première étant la perte de contenu (en rapport avec le contenu traité par le générateur) et la seconde concernant la perte antagoniste (en rapport avec la décision du discriminateur). Plus précisément, la fonction de pertes est définie comme décrit dans l'équation (4.1).

$$l^{SR} = \underbrace{\underbrace{l_{\mathrm{X}}^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}} \tag{4.1}$$

Institut Polytechnique de Paris
91120 Palaiseau, France

Concernant la perte de contenu, les auteurs proposent plusieurs versions. La mesure classique EQM peut être utilisée ici (équation (4.2)).

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2 \tag{4.2}$$

Néanmoins, les auteurs soulignent que bien qu'elle soit actuellement la plus répandue, elle conduit à des résultats insatisfaisants visuellement. A la place, une fonction de pertes, appelée **perte VGG**, basée sur les travaux de [Gatys15] et [Johnson16] est proposée. Elle est calculée en fonction d'une version pré-entrainée des 19 couches du VGG, après une activation par ReLU. La perte VGG est définie comme la distance euclidienne **dans l'espace des paramètres** entre l'image reconstruite par le générateur et l'image de référence, comme décrit dans l'équation (4.3).

$$\begin{aligned} l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} \\ - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \end{aligned} \tag{4.3}$$

La perte antagoniste est quant à elle définie comme décrit dans l'équation suivante :

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{4.4}$$

L'entrainement a été réalisé avec 10^5 itérations, avec un taux d'apprentissage de 10^{-4} puis avec 10^5 itérations supplémentaires avec un, taux d'apprentissage de 10^{-4}, sur un panel de 350 000 images extraites de la base de données ImageNet. Chaque mini-batch est composé de 16 patches de 96×96 pixels, extraits des images d'entrainement. L'algorithme d'optimisation utilisé est Adam. La structure du réseau adoptée utilise un nombre de 16 blocs résiduels et le facteur d'agrandissement étudié est de ×4.

Pour le modèle proposé, un important ensemble d'images d'évaluation a été considéré, incluant les populaires corpus d'évaluation Set5, Set14, BSD100, et BSD300 [Martin01].

Les méthodes retenues pour comparaisons sont les suivantes : *Nearest Neighbor*, interpolation bicubique, SRCNN, SelfExSR [Huang15], DRCN [Kim16], ESPCN [Shi16], SRResNet-MSE [Ledig17], SRResNet-VGG22, SRGAN-MSE, SRGAN-VGG22 et SRGAN-VGG54.

Institut Polytechnique de Paris
91120 Palaiseau, France

Pour les méthodes SRResNet et SRGAN :
-MSE signifie que ces réseaux ont été entrainés avec une fonction d'EQM pour la composante perte de contenu.
-VGG22 signifie que la perte VGG a été utilisée et calculée sur des couches de bas-niveau du VGG.
-VGG54 est similaire au VGG22 mais a été calculé sur des couches de plus haut niveau, sur un VGG plus profond.

L'intérêt de ce dernier est que des couches de plus haut niveau présentent un plus gros potentiel de contenir des informations plus précises quant au contenu des images [Simonyan14]. D'après les auteurs, les résultats les plus convaincants visuellement sont effectivement obtenus en utilisant le VGG54 plutôt que le VGG22, comme illustré Figure 4.8.

Figure 4.8 : Illustration des différences de détails obtenus via le SRGAN par utilisation respectivement du VGG22 et 54 (Source : [Ledig17])

Les auteurs ont réalisé une évaluation selon trois critères, soit le PSNR, le SSIM et une vaste évaluation subjective de *Mean Opinion Score* (MOS) conduite sur un panel de 26 personnes ayant été amenées à évaluer 1128 images chacune (12 méthodes évaluées sur les Set5 et Set14 et 9 méthodes évaluée sur le BSD100). Pour cette évaluation, un score entre 1 (Mauvaise qualité) et 5 (Excellente qualité) a été attribué à chaque image. Une calibration des évaluateurs a été réalisée au préalable avec 20 images issues du corpus BSD300.

Les conclusions des résultats obtenus sont les suivantes :

- Concernant les scores PSNR et SSIM, les (largement) meilleurs résultats ont été obtenus à la fois pour le SRResNet et pour le SRGAN avec l'entrainement utilisant l'EQM comme fonction de pertes. Nous reviendrons en détails sur l'explication de ce phénomène dans le Chapitre 6.
- L'évaluation MOS a quant à elle donnée des résultats bien différents. Même si elle n'a pas su s'imposer significativement sur les Set5 et Set14 (au point que les auteurs ne parviennent pas à une conclusion quant à la meilleure fonction de pertes dans ce cas), elle domine largement sur le corpus BSD100, comme illustré Figure 4.9.

Set5	nearest	bicubic	SRCNN	SelfExSR	DRCN	ESPCN	**SRResNet**	**SRGAN**	HR
PSNR	26.26	28.43	30.07	30.33	31.52	30.76	**32.05**	29.40	∞
SSIM	0.7552	0.8211	0.8627	0.872	0.8938	0.8784	**0.9019**	0.8472	1
MOS	1.28	1.97	2.57	2.65	3.26	2.89	3.37	**3.58**	4.32
Set14									
PSNR	24.64	25.99	27.18	27.45	28.02	27.66	**28.49**	26.02	∞
SSIM	0.7100	0.7486	0.7861	0.7972	0.8074	0.8004	**0.8184**	0.7397	1
MOS	1.20	1.80	2.26	2.34	2.84	2.52	2.98	**3.72**	4.32
BSD100									
PSNR	25.02	25.94	26.68	26.83	27.21	27.02	**27.58**	25.16	∞
SSIM	0.6606	0.6935	0.7291	0.7387	0.7493	0.7442	**0.7620**	0.6688	1
MOS	1.11	1.47	1.87	1.89	2.12	2.01	2.29	**3.56**	4.46

Figure 4.9 : Evaluation du SRGAN par comparaison des MOS, PSNR et SSIM sur différents sets d'évaluation

Mentionnons enfin que deux ans après l'introduction du réseau SRGAN, une version améliorée de celui-ci, nommée *Enhanced* SRGAN (ESRGAN) [Wang18b] a été proposée dans le cadre du *Challenge* international PIRM2018-SR [Blau18]. Elle sera détaillée dans la Section 5.5.1.

4.5 Discussion et travaux futurs sur la compression par GAN

Nous avons pu voir à partir de cet état de l'art que l'arrivée des GAN et de leur association avec des fonctions de pertes perceptuelles ont permis une avancée majeure dans le domaine de la « transformation d'images ». Ces techniques en plus de promettre de très bons résultats, s'avèrent en pratique êtres plutôt aisées à utiliser, décliner et manipuler.

Les réseaux de type GAN offrent des promesses spectaculaires dans le domaine de la compression d'images. Le défi à relever consiste à déterminer la représentation minimale, en termes d'entropie des données à coder, qui puisse conduire à un schéma optimal de compression.

Actuellement, la tendance semble aller vers un schéma de compression/décompression s'appuyant entièrement sur des réseaux de neurones. Cette logique reprend l'idée des auto-encodeurs [Theis17], très présents dans la littérature mais n'ayant jusque-là pas montré assez d'efficacité de compression.

La compression de bout-à-bout par CNN et plus particulièrement en utilisant des réseaux GANs pour des objectifs de reconstruction offre des résultats visuellement spectaculaires, et ce même à très bas-débit. Ces derniers se posent aujourd'hui comme état de l'art en offrant de bien meilleures performances que n'importe quel codec standard, et ce, dans la grande majorité des cas. Néanmoins, malgré la puissance de ces nouveaux schémas de compression, un problème majeur apparaît. Il concerne notamment les aspects de fidélité du rendu. En effet, même si d'un point de vue visuel l'image est beaucoup plus plaisante, ce type de traitement peut facilement conduire à la destruction au ou remplacement d'éléments d'information critiques présents dans l'image, comme illustré Figure 4.10.

a) Original PNG b) BPG at 0,034 bpp c) GC at 0,034 bpp

Figure 4.10 : Re-création de détails par la compression générative. Le texte présent dans l'image est préservé par une compression BPG pure (au milieu) mais est complètement remplacé par le réseau GAN (à droite). (Source : [Agustsson19])

Dans cet exemple, le texte est conservé par la compression BPG, même à de forts taux de compression. En revanche, il est totalement détruit par la compression générative. De plus, d'un point de vue plus général, nous pouvons supposer que dans un cadre d'utilisation grand public, l'utilisateur peut facilement ne pas souhaiter avoir des éléments de son contenu totalement remplacés lors de la compression/décompression.

Face à ce problème et compte tenu de notre cas d'application visant à la transmission de contenus critiques comme des cartes d'identité ou formulaires, nous avons orienté nos travaux vers un schéma de compression offrant le meilleur rapport débit/distorsion mais en posant la contrainte que les contenus sensibles doivent être conservés lors du traitement. Les codecs standards possèdent justement la faculté d'offrir de très bonnes performances quant à la conservation de contenus écrits. Pour ces raisons, la décision a été prise d'écarter la compression générative et d'inclure directement les codecs standards, en particulier le BPG, en tant que partie intégrante de nos schémas de compression.

Tentons-donc une autre approche : d'une part, nous savons que les réseaux de type GAN excellent dans la génération/reconstruction de contenus visuels. D'autre part, le codec BPG est relativement peu coûteux en ressources logicielles et permet d'obtenir des taux de compression approchant les objectifs de notre application industrielle, au prix d'une dégradation significative de l'image. Nous orienterons donc nos travaux sur l'amélioration de cette qualité. Ainsi, dans un premier temps, nous canaliserons nos efforts sur la partie décodage/reconstruction de contenus fortement compressés, et donc fortement dégradés, en commençant par la compression BPG. Cette piste s'appuie sur de précédents travaux, comme ceux présentés dans [Svoboda16], [Dong15b], qui visent à réduire les artéfacts de compression JPEG en utilisant des réseaux de neurones profonds.

Pour cela nous allons considérer deux approches différentes La première méthode concerne l'utilisation des techniques de super-résolution par SRGAN. Dans ce cas, l'objectif est de profiter de la réduction massive de débit naturellement obtenue par le processus de sous-échantillonnage, puis d'appliquer une compression BPG relativement légère, afin de ne pas trop dégrader l'image et de permettre au réseau SRGAN de rendre une reconstruction de qualité acceptable. L'enjeu dans ce cas est de déterminer, de manière automatique, le bon équilibre entre compression BPG et reconstruction par SRGAN.

La deuxième, consiste à entrainer finement un réseau de neurones que nous allons construire à partir du SRGAN destiné à uniquement réaliser une réduction des artéfacts de compression. Nous espérons ainsi exploiter la combinaison d'une fonction de pertes perceptuelle couplée à la capacité d'apprentissage d'un réseau de type GAN afin de corriger de forts artéfacts de compression sans pour autant réaliser de super-résolution.

Ces approches sont détaillées au Chapitre suivant.

Chapitre 5. Compression par réseaux GAN (*Generative Adversarial Networks*)

Résumé. Dans ce chapitre, nous détaillons nos principales contributions liées à l'utilisation de réseaux de types GAN à des fins de compression très bas débit d'images. L'objectif principal est d'améliorer sensiblement la qualité des images décodées, sans pour autant augmenter leur débit. La solution proposée se positionne ainsi comme une étape de post-traitement, qui apparaît en bout de la chaîne de décodage et réalise une reconstruction d'image à partir de contenus fortement compressés.

Dans ce cadre, nous proposons plusieurs combinaisons d'approches de super-résolution et de réduction d'artéfacts de compression, entraînées de manière à garantir une généricité aussi grande que possible des résultats. Un des défis majeurs que nous adressons dans ce chapitre concerne l'équilibre à déterminer entre le niveau de réduction de la quantité d'information à coder, qui se traduit dans notre cas par le facteur de super-résolution considéré, la qualité visuelle des reconstructions et enfin la fidélité des contenus dits critiques qui peuvent apparaître dans une image et qui doivent être à tout prix préservés.

Les approches GAN retenues et adaptées pour nos objectifs concernent les réseaux SRGAN et ESRGAN. Une attention particulière est dédiée aux conditions d'entrainement nécessaires pour atteindre des résultats performants, ainsi qu'aux différentes fonctions de perte, à la fois orientées pixels et perceptuelles, qui peuvent être utilisées.

L'impact de ces différentes conditions d'utilisation et paramètres sous-jacents est étudié en détail, afin d'en dériver les pistes les plus prometteuses pour nos objectifs de compression.

Mots clés **:** SRGAN, ESRGAN, réduction d'artéfacts de compression, super-résolution, reconstruction d'image, fonctions de pertes.

Institut Polytechnique de Paris
91120 Palaiseau, France

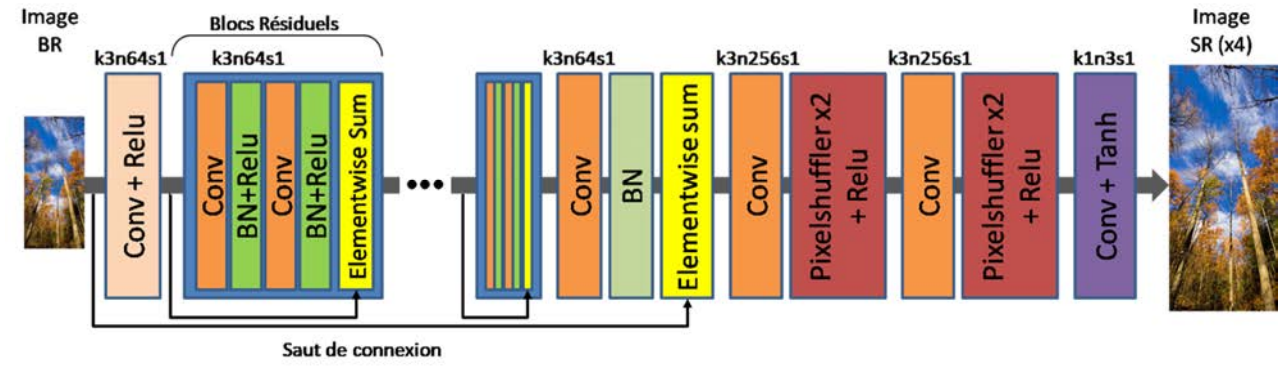
5.1 Approche naïve : SRGAN pré-entrainé et compression BPG

Afin d'obtenir un premier aperçu de la pertinence de notre idée, nous avons considéré une première approche directe, qui ne nécessite pas de paramétrage particulier. L'objectif ici est de vérifier si une méthode SRGAN combinée à une compression BPG permettrait d'obtenir de meilleurs résultats que la compression BPG seule et ce, malgré un modèle pré-entrainé uniquement à des fins de super-résolution.

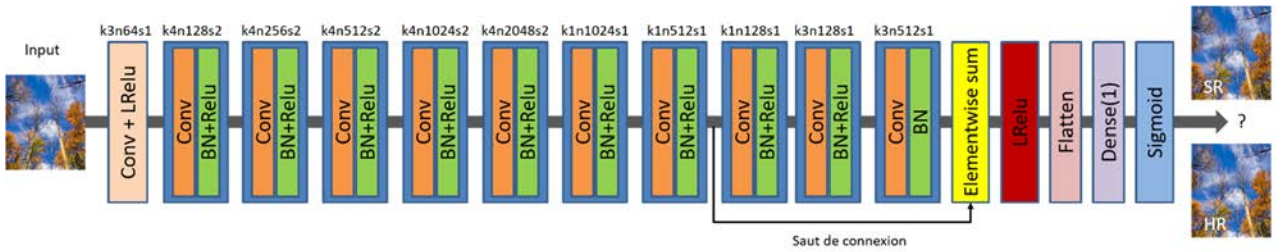
Pour cela nous utilisons le réseau SRGAN introduit dans [Ledig17], en adoptant l'implantation et le modèle pré-entrainé disponible dans [ZsDongHao] et implanté via le *framework* Tensorlayer [Dong17]. Ici, le modèle est programmé en utilisant les *frameworks* Tensorflow et Tensorlayer, et l'entrainement a été réalisé en utilisant les 800 images du *dataset* DIV2K [Agustsson17].

L'entrainement a été réalisé pour une SR ×4, la taille de batch étant de 8, avec un nombre de 2000 itérations. Dans cette implantation, des *patchs* de taille 384×384 pixels sont extraits des images en haute résolution. Ces patchs sont ensuite sous-échantillonnés à 96×96 pixels. La méthode d'optimisation employée est ADAM, avec un paramètre β_1 de 0.9, un taux d'apprentissage de 10^{-4} et un déclin de 0,1 au bout de 1000 *epochs*.

Notons qu'ici le modèle adopté présente de légères différences par rapport au papier d'origine. L'architecture du réseau est illustrée Figure 5.1 : Architecture du réseau SRGAN utilisé



a) Schéma du générateur



b) Schéma du discriminateur

Figure 5.1 : Architecture du réseau SRGAN utilisé

Institut Polytechnique de Paris
91120 Palaiseau, France

La fonction de pertes est également légèrement différente de celle proposée dans les travaux de Ledig *et al*. Ici, la « perte de contenu » utilisée est mixte, composée à la fois d'une perte VGG et d'une perte EQM. La fonction de pertes utilisée, qui sera la même dans toutes les expérimentations concernant le réseau SRGAN, se présente alors sous la forme suivante :

$$Gen_{loss} = \underbrace{\alpha\, EQM_{loss} + \beta\, VGG_{loss}}_{\text{Perte de contenu}} + \underbrace{\lambda\, GAN_{loss}}_{\text{Perte antagoniste}} \tag{5.1}$$

Avec α=1, β=2.10^{-6} et λ=10^{-3}

Pour la composante de perte VGG, le réseau VGG19 est utilisé (Figure 5.2).

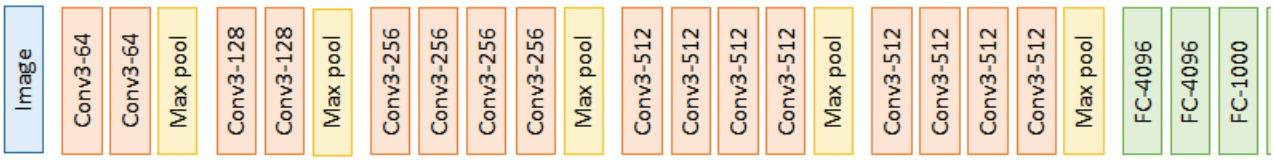


Figure 5.2 : Architecture du réseau VGG19 utilisé pour la perte VGG

Les poids du réseau VGG19 pré-entrainé sont quant à eux ceux disponibles en téléchargement depuis la page du Github du réseau SRGAN utilisé.

Un premier exemple de résultat obtenu avec cette approche, notée dorénavant par SRGAN+BPG est illustré Figure 5.3.

Figure 5.3 : Comparaison SRGAN + BPG (en haut) et BPG seul à 6,35 ko (0,13 bpp) (en bas)

Sur ce premier exemple, l'approche SRGAN+BPG donne de bien meilleurs résultats que la compression BPG seule. Les traits sont très nets et les artéfacts de compression peu apparents, malgré la forte réduction de débit (environ 1/100 par rapport au PNG initial). Ainsi, la méthode arrive à reconstituer les éléments de détails initialement perdus par la compression BPG.

Institut Polytechnique de Paris
91120 Palaiseau, France

Pour investiguer encore plus loin les possibilités offertes par cette approche, nous avons considéré la même expérimentation et sur la même image, cette fois avec une compression encore plus sévère, correspondant à 0,013 bpp. Les résultats obtenus sont illustrés Figure 5.4.

Figure 5.4 : Comparaison SRGAN + BPG (en haut) et BPG seul à 1,9 ko (0,04 bpp) (en bas)

Institut Polytechnique de Paris
91120 Palaiseau, France

Les deux images ici sont très dégradées et présentent de forts artéfacts de compression (réduction d'environ 1/300 par rapport au PNG initial). Néanmoins, nous pouvons observer que les artéfacts de BPG sont beaucoup plus notables que ceux de BPG+SRGAN, dans la mesure où l'on voit apparaître l'effet de blocs, ainsi que les lignes de prédiction.

Cette fois-ci, contrairement à l'expérience similaire réalisée avec le réseau SRCNN, nous obtenons des premiers résultats très encourageants. De plus, une large marge de progression est possible en réalisant des entrainements adaptés et dédiés à notre problème et en considérant notamment des images fortement compressées en BPG.

Dans la suite, nous introduisons une nouvelle méthode destinée à réaliser à la fois super-résolution et réduction d'artéfacts, en utilisant un seul et même réseau, spécifiquement entrainé pour réaliser ces deux tâches conjointement. Le schéma résumant le pipeline de compression proposé est illustré Figure 5.5 :

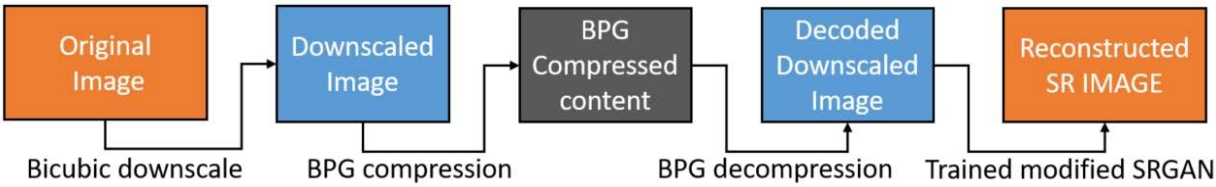


Figure 5.5 : Pipeline proposé alliant compression par codec BPG et SR par SRGAN

5.2 Reconstruction ×4 par SRGAN

La première application de notre pipeline a été testée avec un agrandissement ×4 de l'image, étant donné qu'il s'agissait de la configuration de base du SRGAN, en plus de la RAC. Nous nommerons le procédé de SR ×4 + RAC à travers un même entrainement *Reconstruction ×4*.

La RAC couplée à la SR apparait comme un problème particulièrement complexe, comme mentionné Section 3.3.2. Pour l'aborder au mieux, nous avons changé de corpus d'entrainement afin d'obtenir un grand nombre de données, présentant un panel de qualités et de contenus très varié. Bien que les réseaux à base de GAN ne soient pas les plus sensibles aux problèmes d'*overfitting*, comme notre SRGAN s'avère être très profond, nous préférions bénéficier d'un corpus aussi grand possible (*i.e.* correspondant au maximum que pouvait gérer notre machine).

Nous avons donc créé notre corpus d'entrainement composé de 20000 images extraites de la base d'images MirFlickr25k[Huiskes08]. Nous désignerons ce corpus par Mirflickr20k. Ayant la particularité d'être composé en grande partie de photos amateurs téléchargées sur le réseau social Flickr, cette base d'images offre l'avantage de présenter la diversité recherchée. Concernant la résolution de ces images, elle varie entre 128 et 500 pixels.

Dans le cadre de nos expérimentations, les images n'ont subi aucune transformation, en dehors de la réduction de la résolution et de la compression BPG.

Concernant la version basse-résolution (BR) de notre corpus, la résolution de toutes les images du *dataset* Mirflickr20k a été divisée par 4 et une interpolation bicubique a été appliquée pour gérer les aspects sub-pixeliques. Les images ont ensuite été compressées en BPG avec un facteur de compression q choisi aléatoirement entre 15 et 25.

Un premier entrainement a été réalisé en utilisant la même architecture que celle déjà décrite à la Section 5.1. La seule modification que nous avons apportée a été de réduire la taille des patches HR à 128×128 pixels. Cela permet, d'une part, à les faire correspondre correctement aux dimensions de nos images, et d'autre part, d'avoir une meilleure adaptation par rapport à la taille des *Transform Blocs* utilisés dans la compression BPG. La fonction de pertes et les hyperparamètres sont eux identiques à ceux énoncés dans l'implantation décrite dans la Section 5.1.

Initialement, un entrainement « complet » réalisé sur 2000 *epochs* était prévu. Or nous avons stoppé l'entrainement à 1000 *epochs*, puisque les résultats obtenus présentaient de forts artéfacts, comme illustré Figure 5.6.

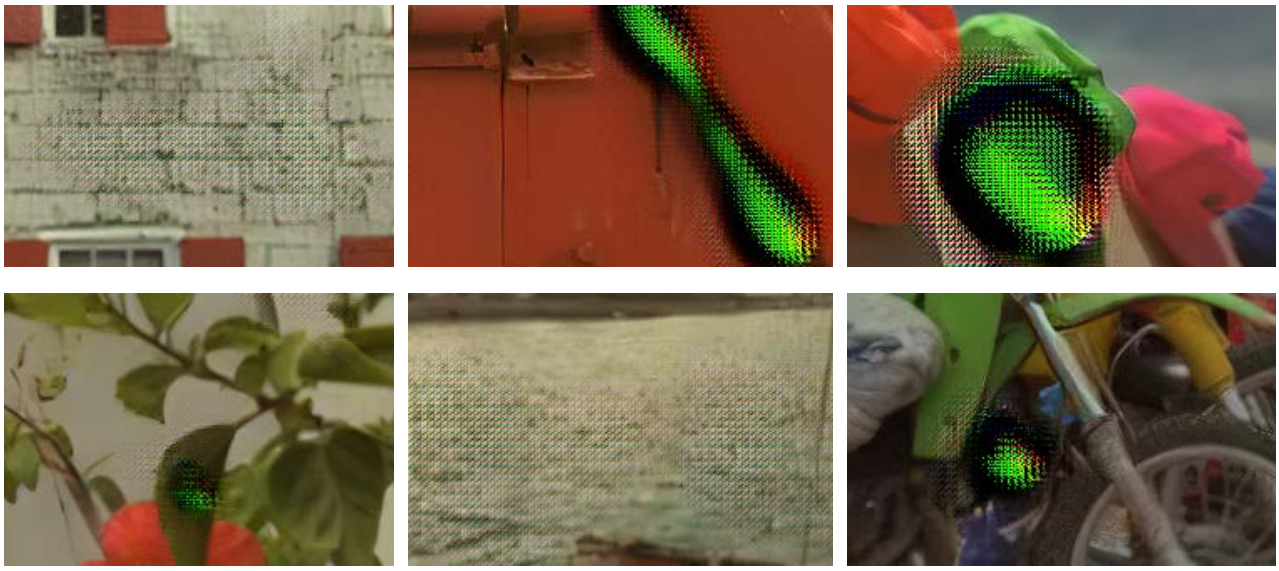


Figure 5.6 : Exemples illustrant les forts artéfacts parasites apparaissant en résultat de notre entrainement

Comme nous pouvons le voir, de forts artéfacts parasites apparaissent sur nos images, et il ne s'agit pas de cas isolés. Après investigations, nous avons pu déterminer la nature du problème, qui vient des couches de normalisation de *batch* (BN – *Batch Normalization*) présentes dans les blocs résiduels de notre réseau. De plus amples détails sont donnés à propos de ce phénomène dans la Section 5.5.1 (ESRGAN). Mentionnons que ce problème a été également identifié dans [Lim17], où des recommandations quant à l'architecture à adopter pour contourner ce problème sont formulées. En suivant ces recommandations, nous avons donc modifié l'architecture de notre générateur comme illustré Figure 5.7.

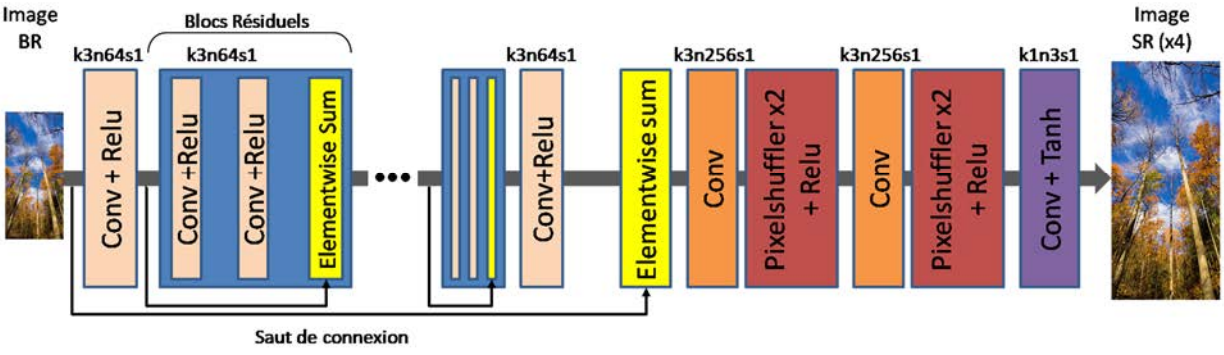


Figure 5.7 : Schéma du générateur du SRGAN modifié pour la Reconstruction ×4

Institut Polytechnique de Paris
91120 Palaiseau, France

La principale différence ici par rapport à l'architecture standard concerne l'élimination de toutes les couches de BN présentes dans le réseau initial.

Un nouvel entrainement a été ensuite réalisé sur 2000 *epochs* avec cette nouvelle architecture et les mêmes paramètres que lors de l'entrainement précédent. Les nouveaux résultats obtenus sont illustrés Figure 5.8.

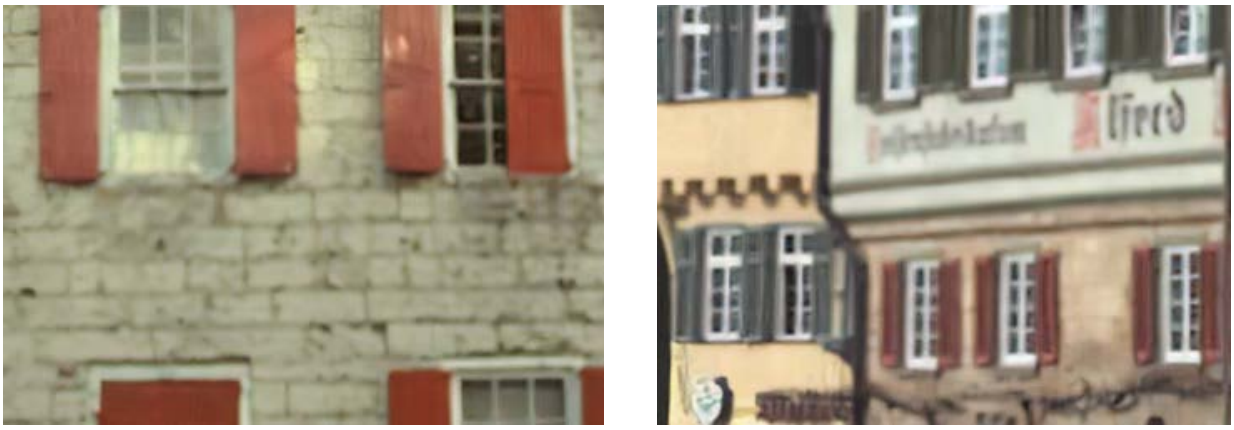


Figure 5.8 : Exemple d'images obtenues par Reconstruction x4 avec notre SRGAN modifié

Les résultats obtenus ne présentent plus les artéfacts gênants que l'on pouvait constater sur les images obtenues avec le SRGAN avec les couches de BN. En revanche, les résultats apparaissent comme assez médiocres et présentant un flou très fort dans certaines régions de l'image, comme ce que l'on peut obtenir avec un flou artistique en photographie. La seule image de notre ensemble d'évaluation relativement épargnée par ce phénomène est celle illustrée Figure 5.9, car dans ce cas l'image originale se trouvait être déjà séparée en deux plans distincts par un flou artistique.

Institut Polytechnique de Paris
91120 Palaiseau, France

a) Image originale

b) Reconstruction ×4 par SRGAN modifié

Figure 5.9 : Image de notre corpus d'évaluation ne présentant pas de flou suite à une Reconstruction ×4 par SRGAN

Dans tous les cas, la Reconstruction ×4 par SRGAN, modifié ou non conduit à des défauts visuels majeurs, qui concerne les éléments de texture, de détails ou encore de texte incrusté dans les images.

Par conséquent, ce modèle ne semble pas adapté, du moins dans l'état étudié, pour offrir une solution viable à des objectifs de compression d'images. Une explication possible pourrait être la perte trop importante d'éléments d'image indispensables à la reconstruction engendrée par le sous-échantillonnage trop important. Afin de vérifier cette hypothèse, dans les sections suivantes nous allégeons les conditions de sous-échantillonnage et cherchons surtout à déterminer un juste équilibre entre paramètres de sous-échantillonnage et de compression.

5.3 Reconstruction ×2 par SRGAN

Afin de conserver un meilleur compris fidélité/débit/qualité visuelle, nous avons investigué une autre piste, qui consiste à partir d'images de taille plus importante mais compressées plus fortement. En effet, si l'image à traiter n'est pas suffisamment grande (inférieure à 800×800 pixels par exemple), le niveau de détails présents dans l'image basse résolution se trouve être trop faible pour obtenir une reconstruction fidèle du contenu.

Nous allons donc considérer dans la suite une super-résolution d'un facteur ×2. Pour cela, nous gardons la même architecture de générateur mais en retirant une couche d'agrandissement de résolution par 2 (Figure 5.10).

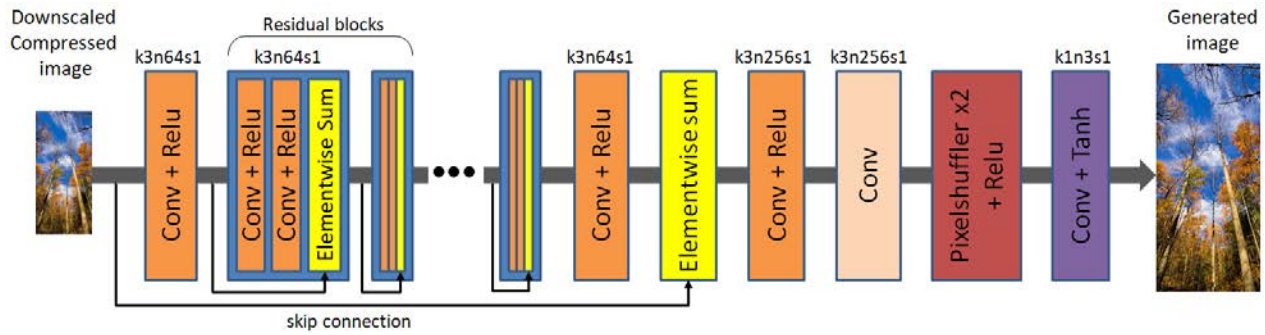


Figure 5.10 : Schéma du générateur du SRGAN modifié pour la Reconstruction ×2

D'autres modifications ont été également nécessaires pour parvenir au modèle final. Tout d'abord, afin de mieux prendre en compte la taille des TB (*Transform Blocks*) du format BPG, la taille des *patchs* en basse résolution a été réduite à 64×64. Cela permet de considérer, dans le meilleur des cas, des blocs entiers.

Enfin, dans l'expérience décrite dans la section précédente, le modèle original ne prenait pas en compte le corpus d'entrainement en basse résolution, les *patchs* basse-résolution étant obtenus uniquement à partir de leur contrepartie HR. Cela peut conduire à un certain déséquilibre, dans la mesure où pour nos expérimentations nous avions besoin que le *dataset* BR soit compressé en BPG et que le *dataset* HR ne le soit pas. Nous avons résolu ce problème, en nous assurant que le modèle puisse se servir séparément dans les images HR et BR adéquates et ainsi de prendre comme entrées des *patchs* BR compressés en BPG et leurs correspondants dans les images HR non-compressées. En outre, nous changeons de manière aléatoire les coordonnées d'extraction des patchs à chaque *epoch* afin de permettre une meilleure généralisation du modèle, tout en gardant la bonne correspondance de coordonnées BR/HR.

Concernant le discriminateur, aucune modification n'a été nécessaire.

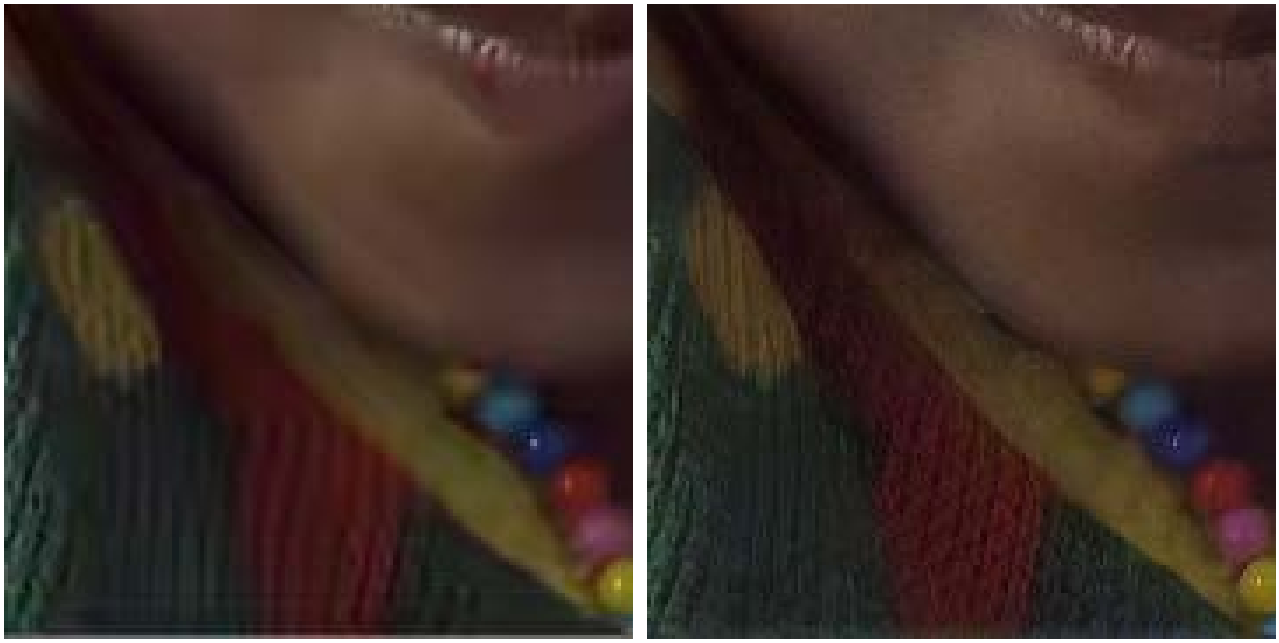
Pour cet entrainement, la résolution de toutes nos images a été réduite en appliquant sous-échantillonnage d'un facteur 2 et interpolation bicubique. Ensuite, les images ont été compressées en BPG avec un facteur de qualité « q » variant aléatoirement entre 25 et 35. Rappelons qu'à chaque *epoch*, un patch différent de celui correspondant à l'*epoch* précédent est extrait aléatoirement de l'image, amenant le nombre « d'images » traitées à bien plus que 20000 pour l'entrainement.

Institut Polytechnique de Paris
91120 Palaiseau, France

L'entrainement a été réalisé sur 2000 *epochs*, avec un taux d'apprentissage de 10^{-4} et un déclin de 0.1 au bout de 1000 *epochs*, une taille de *batchs* de 8 images et en utilisant l'optimisateur ADAM avec un paramètre β_1 de 0.9.

Pour l'évaluation, nous avons décidé de nous concentrer sur les 24 images du corpus d'évaluation de Kodak car, d'une part, il s'agit d'un ensemble d'images très populaire pour des objectifs de *benchmark* au sein de la communauté, et d'autre part il présente une diversité très intéressante en termes de contenu. Nous ajouterons ponctuellement à ce corpus quelques images tirées au hasard d'une partie non utilisée pour l'entraînement de la base MirFlickr afin de mettre en évidence quelques éléments d'analyse précis, non décelables sur les images Kodak. Quelques exemples de résultats obtenus sont illustrés Figure 5.11.

a) 5854 octets b) 5289 octets



a) 7457 octets b) 6812 octets

Figure 5.11 : Comparaison entre compression BPG simple (à gauche) et Reconstruction ×2 (à droite) à débit équivalent. Les détails de la Reconstruction ×2 sont bien plus marqués et agréables à regarder.

Institut Polytechnique de Paris
91120 Palaiseau, France

De manière générale, la Reconstruction ×2 offre de bien meilleurs résultats visuels que sa contrepartie en simple BPG. Les résultats sont plus détaillés, plus texturés et bien plus agréables à regarder. De plus, les artéfacts de compression marqués de BPG n'apparaissent pas ou peu dans la Reconstruction ×2.

A noter également le temps de traitement par image, qui est inférieur à une seconde et ce, même sans la sollicitation de la carte graphique. De plus, il est à noter que nous utilisons une version prototype du réseau, implémentée via Tensorflow et Tensorlayer, non destinée à la production industrielle. Il est donc tout à fait possible d'envisager des optimisations supplémentaires pour obtenir des temps de traitement encore plus faibles.

L'évaluation à la fois objective et subjective de nos résultats sera présentée en détails au Chapitre 6. Néanmoins, il est à noter que ce type de réseau offre des performances relativement pauvres par rapport aux métriques objectives traditionnelles.

En revanche, bien que performante, la Reconstruction ×2 reste susceptible de compromettre certains détails critiques comme des éléments d'écriture si leur taille dans l'image BR n'est pas suffisamment importante. Ce phénomène est illustré Figure 5.12.

a) Compression BPG, q=42 b) Reconstruction ×2, q=30

Figure 5.12 : Comparaison entre BPG simple (à gauche) et Reconstruction ×2 (à droite) à débit égal. La qualité globale de la Reconstruction ×2 est bien meilleure, mais le texte critique se trouve être illisible (*la photo entière n'est pas montrée pour des raisons de respect de la vie privée du possesseur de la carte*)

Pour cela, nous allons explorer une piste plus simple, en nous débarrassant de l'aspect SR de notre *pipeline* pour uniquement réaliser une réduction d'artéfacts de compression et proposer ce que nous appellerons Reconstruction ×1.

5.4 Réduction d'artéfacts de compression par SRGAN : Reconstruction ×1

La Reconstruction ×1 utilise toujours comme base un réseau de neurones de type SRGAN afin de réaliser uniquement une opération de réduction d'artéfacts de compression (RAC) s'affranchissant de tout processus de sous-échantillonnage/super-résolution. Le schéma du générateur SGAN modifié est illustré Figure 5.13.

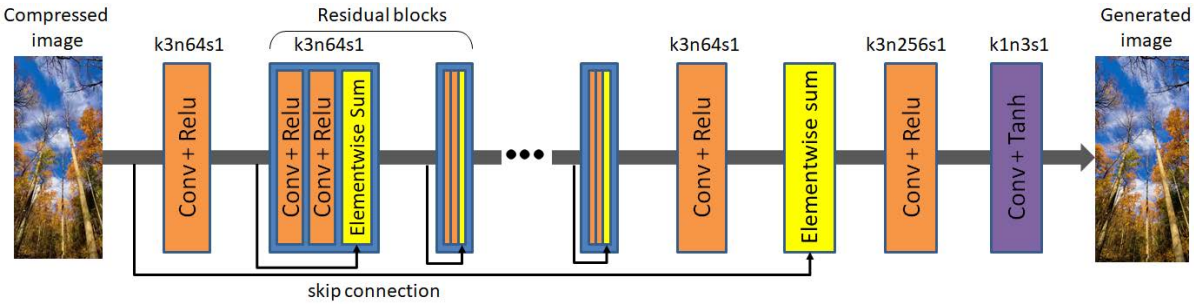


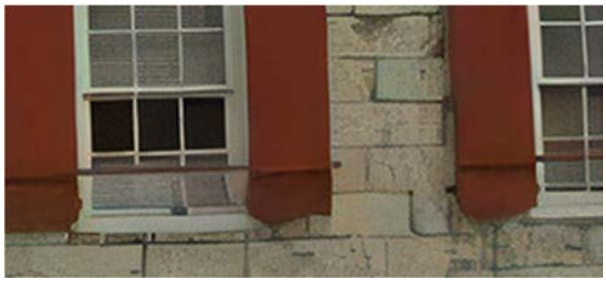
Figure 5.13 : Schéma du générateur du SRGAN modifié pour la Reconstruction ×1

En effet, si nous retirons les couches d'agrandissement de la résolution de notre modèle, nous arrivons à un réseau GAN profond utilisant une fonction de pertes mixte EQM et perceptuelle. Or, ce type de réseau apparait comme tout à fait en mesure de traiter efficacement de problèmes de RAC simples.

Nous avons tout d'abord considéré une première expérience, visant à traiter un seul facteur de compression, afin d'étudier la capacité de notre modèle à traiter des artéfacts de compression complexes et marqués. Pour cela, ainsi qu'à des fins de *benchmark*, nous avons retenu le même corpus d'entrainement de 20000 images utilisé pour l'approche de Reconstruction ×2. Ici les images en « basse qualité », que nous continuerons à nommer BR pour plus de cohérence, ont été compressées en BPG avec un facteur de qualité q » de 43, en résolution originale. Nous nommerons cette expérience Reconstruction ×1_43.

L'entrainement a été réalisé avec les paramètres identiques à ceux appliqués pour la Reconstruction ×2. Les résultats obtenus sont illustrés Figure 5.14.

Image BPG



Reconstruction ×1_43

Image Kodim01, q=43, 9317octets

Image BPG

Reconstruction ×1_43

Image Kodim04, q=43, 3410octets

Figure 5.14 : Comparaison entre BPG seul et Reconstruction ×1_43

Nous pouvons ici tirer des conclusions similaires à celles de la Reconstruction ×2. En effet, les contenus reconstruits apparaissent comme plus détaillés, plus agréables à regarder et présentent des artéfacts de compression plus faibles. On peut en déduire que notre méthode parvient tout à fait à réaliser une réduction/suppression d'artéfacts de compression complexes et marqués, issus d'une forte compression. Néanmoins, dans la mesure où un seul facteur de compression est traité ici (q=43), notre modèle n'est pas encore suffisamment générique.

En particulier, même si un facteur de qualité BPG de 43 offre un bon compromis entre très bas débit et qualité, il apparait nécessaire de traiter différents facteurs de qualité en vue d'une utilisation robuste, générique et grand public de notre méthode.

Nous avons donc testé la même approche mais avec des images cette fois-ci compressées avec des facteurs de qualité choisis aléatoirement entre 35 et 45. Le choix de cette plage de valeurs repose sur le fait qu'en dessous de 35, le contenu est très peu altéré visuellement. Or, l'état de l'art montre que la RAC de contenus « faiblement » compressés donnait déjà de très bons résultats et ne nécessitait pas forcément une investigation plus approfondie.

A contrario, au-delà d'un facteur de compression BPG de 45, le contenu devient excessivement altéré. La perte d'information utile est alors tellement importante (Figure 5.15), qu'une reconstruction satisfaisante en utilisant un modèle « générique » devient très difficile.

q=43 q=45

Figure 5.15: Ecriture sur une image de casquette issue de l'image « Kodim03 ». Nous pouvons clairement lire le mot « Bahamas » avec un facteur de compression q=43 mais cela est très difficile voire impossible avec q=45.

Nous désignerons par Reconstruction x1_gen le modèle réalisant une RAC sur la base d'un entrainement réalisé avec des images compressées avec une plage de valeur de facteur de compression BPG q variée.

Quelques exemples de résultats obtenus avec cette approche sont illustrés Figure 5.16.

a) Image BPG b) Reconstruction x1_gen

Image Kodim03, compression à q=40, 4616 octets

a) Image BPG b) Reconstruction x1_gen

Image Kodim08, compression à q=36, 34193 octets

Figure 5.16: Exemples d'images reconstruites de la base Kodak avec l'approche *Reconstruction x1_gen*

Nous pouvons observer que l'amélioration des résultats par rapport à la compression BPG, bien que visible, est néanmoins plus faible que celle obtenue précédemment avec la Reconstruction ×1_43. En revanche, elle permet de traiter correctement des artéfacts issus de compressions avec des facteurs q multiples. Nous observons également que la Reconstruction ×1_gen, a tendance à ajouter un nouveau type d'artéfacts sur les zones lisses semblables à une « chair de poule ». Cela est dû au fait que le modèle n'est pas capable de distinguer les zones présentant ou non des artéfacts de compression. Ces artéfacts sont caractérisés par de légères tâches claires, denses et régulières. Il s'agit d'une des principales limitations de la méthode proposée. Néanmoins, ces artéfacts, bien que visibles lorsqu'on effectue un zoom sur l'image n'apparaissent quasiment plus à l'œil nu sur l'image globale.

Il convient maintenant de comparer les modèles ×1_43 et ×1_gen entre eux. Pour cela, nous avons appliqué notre modèle entrainé sur des images compressées avec des facteurs de compression q variés. Les résultats obtenus sont illustrés Figure 5.17.

Reconstruction ×1_43

Reconstruction ×1_gen

Image de base : Kodim12, 8231 octets

Reconstruction ×1_43

Reconstruction ×1_gen

Image de base : Kodim20, 10103 octets

Reconstruction ×1_43

Reconstruction ×1_gen

Image de base : Kodim02, BPG= 43, 2362 octets

Figure 5.17 : Comparaison entre Reconstruction ×1_43 et Reconstruction ×1_43

Institut Polytechnique de Paris
91120 Palaiseau, France

Ces figures présentent les mêmes images, traitées avec les approches Reconstruction ×1_43 et Reconstruction ×1_gen. Nous pouvons remarquer que la Reconstruction ×1_43 a tendance à facilement ajouter de nouveaux artéfacts qui peuvent être très gênants et bien plus marqués que pour la Reconstruction ×1_gen.

Ce type d'artéfacts apparait plus fortement au fur et à mesure que l'on s'éloigne du facteur de compression cible q=43. Cela peut s'expliquer par le fait que le modèle essaie de « compenser » les artéfacts de compression BPG en les considérant comme bien plus marqués qu'ils ne le sont vraiment. En effet, même si les artéfacts sont plutôt faibles car issus d'un q moins important (35 par exemple), le modèle les considérera tout de même comme des artéfacts issus d'une compression à q = 43, ayant été entrainé spécifiquement pour cela.

En revanche, quand le facteur q de l'image à reconstruire s'approche la valeur de 43 (soit dans une plage de [41,45]), la reconstruction semble être meilleure que celle obtenue par la Reconstruction ×1_gen car le modèle a été mieux entrainé à traiter de forts artéfacts de compression.

Il apparait également que la Reconstruction x1_43 ne présente pas les artéfacts sous forme de tâches claires régulières comme avec la Reconstruction x1_gen.

Suite à l'analyse de ces résultats, il n'apparait pas de type de reconstruction qui semble clairement outrepasser les autres, car cela dépend en réalité du cas d'application. Néanmoins nous pouvons affirmer qu'il apparait comme possible de réduire efficacement des artéfacts de compression marqués et aussi complexes que ceux issus de formats comme le BPG. Nous pouvons ajouter que les deux approches semblent pertinentes car de bons résultats ont pu être observés à la fois pour une utilisation générique, et pour une utilisation spécifique (*i.e.*, seulement une compression à q=43).

D'un point de vue applicatif nous pouvons donc imaginer que, si l'on possède le temps et les ressources nécessaire, la solution optimale consisterait à entrainer un modèle pour chaque facteur de compression et d'utiliser le modèle correspondant à chaque fois. Ou bien même, pour plus de légèreté, entrainer des modèles pour des plages de valeurs restreintes (comme par exemple 3 valeurs) pour une solution plus économique.

En revanche, dans un cas où l'espace de stockage disponible des points est restreint, un seul entrainement avec une plage de valeur plus grande, à l'image de la Reconstruction ×1_gen, apporte des résultats satisfaisants.

Même si, comme énoncé précédemment, les Reconstructions ×1 et ×2 ne correspondent pas forcément aux mêmes cas d'usage, tentons de les comparer au moins qualitativement. Un exemple de résultats est présenté Figure 5.18.

q=28, Reconstruction x2 q=37, Reconstruction x1_gen q=37, Reconstruction x1_43

Comparaison des Reconstructions x2 et x1 sur l'image Kodim15

q=33, Reconstruction x2 q=41, Reconstruction x1_gen q=41, Reconstruction x1_43

Figure 5.18: Comparaison des Reconstructions x 2 et x1 sur l'image Kodim15

Cette comparaison vient confirmer les observations faites plus haut. Nous retrouvons bien que la Reconstruction ×2 apparait comme très compétitive et conduit à des résultats visuels comparables voire meilleurs que ceux obtenus par Reconstruction ×1, tout en nécessitant deux fois moins de temps de traitement. Il s'agit donc là de la solution à privilégier tant qu'il n'y a pas d'informations critiques/textuelles à conserver.

Concernant la Reconstruction ×1, elle apparait bien comme plus performante en version _43 quand nous nous approchons du facteur q avec lequel le modèle a été entrainé (ici 41). En revanche, pour une un facteur q plus faible (ici 37), la version _gen fournit de meilleurs résultats.

Dans la première partie de ce chapitre, nous avons présenté nos expérimentations conduites sur notre modèle basé sur le réseau SRGAN. Trois types d'expérimentations ont été réalisés, soit les reconstructions ×4, ×2 et ×1. Les résultats obtenus nous permettent de tirer les conclusions suivantes.

Premièrement, il apparait que le type de *pipeline* développé présente une pertinence réelle, notamment lorsqu'il est couplé à un entrainement d'un réseau profond utilisant une perte perceptuelle.

Deuxièmement, nous avons pu statuer que, bien que l'approche présentée apparaisse comme étant efficace, des problèmes de fiabilité apparaissent et la reconstruction n'est pas assurée si une étape de SR est utilisée avec une résolution trop faible des images basse résolution.

Dernièrement, nous pouvons établir que plus le facteur de SR à réaliser est grand, plus le traitement est rapide et plus le gain brut en termes de compression est important. Un compromis entre fidélité, temps d'exécution, gain en débit et résolution d'image de base est alors à considérer, en fonction du cas d'application considéré.

Dans la deuxième partie de ce chapitre, nous allons chercher à améliorer ces résultats afin d'augmenter les taux de compression des contenus à traiter, tout en assurant une qualité optimale. Pour cela, nous nous appuyons sur un nouveau réseau, présenté comme le successeur du SRGAN, et appelé *Enhanced SRGAN* (ESRGAN).

5.5 ESRGAN et compression d'image

5.5.1 Du SRGAN au ESRGAN

Deux ans après l'introduction du réseau SRGAN, une version améliorée de celui-ci, nommée Enhanced SRGAN (ESRGAN) [Wang18b] a été proposée dans le cadre du *Challenge* international PIRM2018-SR [Blau18]. Le nouveau réseau ESRGAN a remporté alors la première place de ce *challenge* dans la catégorie évaluant l'index perceptuel étant ici calculé en fonction du score de Ma [Ma17] et du NIQE [Mittal12]. L'ESRGAN offre, à travers plusieurs optimisations, de bien meilleurs résultats que son prédécesseur SRGAN. De plus, son entraînement est rendu plus aisé.

Les éléments d'optimisation proposés par ESRGAN s'articulent autour de quatre points clefs.

1. Le premier concerne la modification de l'architecture du réseau. La principale réalisation a été de modifier les blocs « de base ». En effet, les blocs de « base » du SRGAN étaient les blocs résiduels, contenant notamment des couches de normalisation de *batchs*. Or, il a été démontré dans [Lim17] que ces couches n'étaient pas adaptées aux opérations de SR avec réseaux profonds. Comme nous l'avons précédemment illustré, les couches de normalisation de *batchs* ont tendance à introduire des forts artéfacts indésirables (*cf.* Figure 5.6) au cours de l'entrainement.
De plus, toujours d'après [Lim17] il apparait que ces couches n'améliorent par les résultats, tout en nécessitant un temps de calcul plus élevé. Plus précisément, c'est lorsque que la distribution statistique des données d'entrainement présente des variations relativement importantes que ces couches introduisent des artéfacts gênants (*cf.* Figure 5.6) et limitent la capacité de généralisation du réseau.
Pour pallier cet inconvénient, une nouvelle structure de blocs de base, s'affranchissant de toute étape de normalisation de *batchs* est proposée. Ce nouveau type de bloc, illustré Figure 5.19 est appelé *Residual in Residual Dense Block* (RRDB).

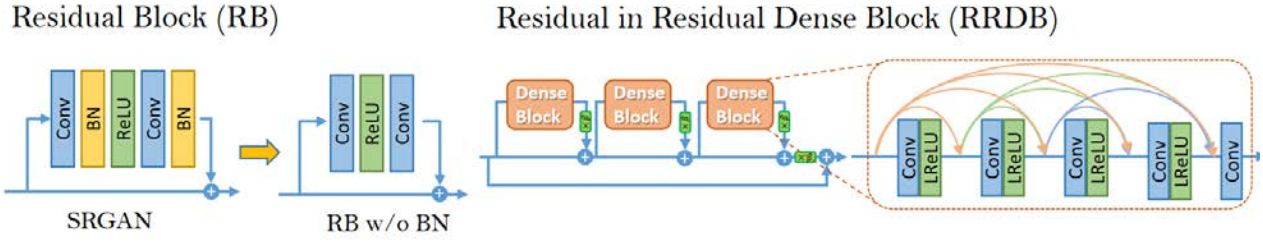


Figure 5.19 : Illustration du passage de bloc résiduel à *Residual in Residual Dense Block*

Institut Polytechnique de Paris
91120 Palaiseau, France

La structure résultante est naturellement plus profonde et plus complexe que celle utilisée dans le SRGAN. Ce choix s'appuie sur l'observation plus générale statuant qu'un réseau avec plus de couches et de connexions amène souvent à de meilleurs résultats, dans la mesure où le corpus d'entrainement est suffisamment riche et adapté aux objectifs ciblés.

2. Des techniques facilitant l'entrainement de réseaux très profonds ont également été utilisées comme la mise à l'échelle (*scaling*) résiduelle comme présenté dans [Lim17], [Szegedy17] ou l'emploi d'une initialisation des paramètres de variance plus réduite.

3. Ensuite, une modification importante a été apportée au niveau du discriminateur, en s'appuyant sur le principe de GAN Relativiste (Figure 5.20) [Jolicoeur-Martineau18]. Ici, le discriminateur ne détermine pas, comme dans le cas d'un réseau GAN classique, la probabilité qu'une image en entrée soit réelle ou fausse mais estime la probabilité qu'une image réelle *x* soit plus réaliste qu'une fausse image *y*.

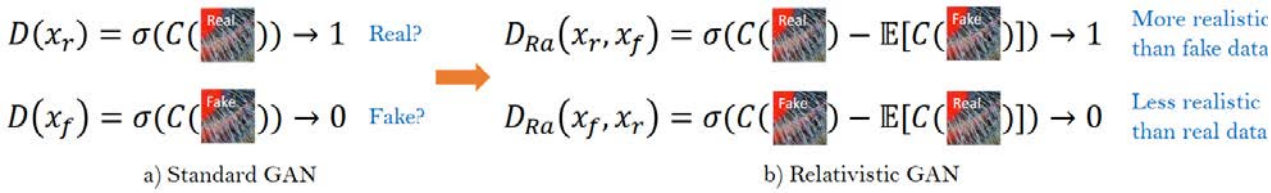


Figure 5.20 : Illustration du principe de réseau GAN relativiste

En conséquence, comme les résultats du discriminateur sont utilisés dans le calcul de la fonction de pertes, cette composante bénéficie alors à la fois des informations des gradients des données réelles et générées, donnant ainsi plus de richesse à la fonction de pertes. D'après les auteurs, ces modifications aident à l'obtention de contours plus précis et de textures plus détaillées.

4. La fonction de pertes, bien que s'agissant toujours d'une fonction de pertes perceptuelles, a également été modifiée. La nouvelle fonction de pertes est définie comme décrit dans l'équation (5.2) :

$$L_G = L_{percept} + \lambda L^{Ra_G} + \eta L_1 \quad (5.2)$$

Avec

$L_{percept}$= perte VGG

L^{Ra_G} = perte du discriminateur du GAN relativiste

L_1 = perte de contenu évaluant la norme L_1entre l'image prédite et la vérité terrain

λ, η = coefficients permettant de réguler les différentes pertes

Cette fois-ci, la composante de perte de contenu inclue également, comme dans le cas du réseau SRGAN, à la fois les pertes VGG et des pertes orientées pixel. Néanmoins, pour ces dernières, la fonction EQM est remplacée par une fonction L_1.

Pour la perte VGG, le réseau VGG19, à différentes profondeurs est utilisé (soit VGG19-22 et VGG19-54) comme dans le papier du SRGAN.

Une différence majeure concernant la perte VGG est ici soulignée. Dans le cas SRGAN, la perte était calculée par rapport aux paramètres du réseau VGG après leur activation. Or, à ces niveaux de profondeur, il apparait qu'un très grand nombre de paramètres sont mis à 0 après activation par ReLU. Ce phénomène, illustré Figure 5.21sur l'image Baboon, peut être très important. Ici, au niveau de la couche VGG19-54, seulement 11.17% des paramètres sont activés.

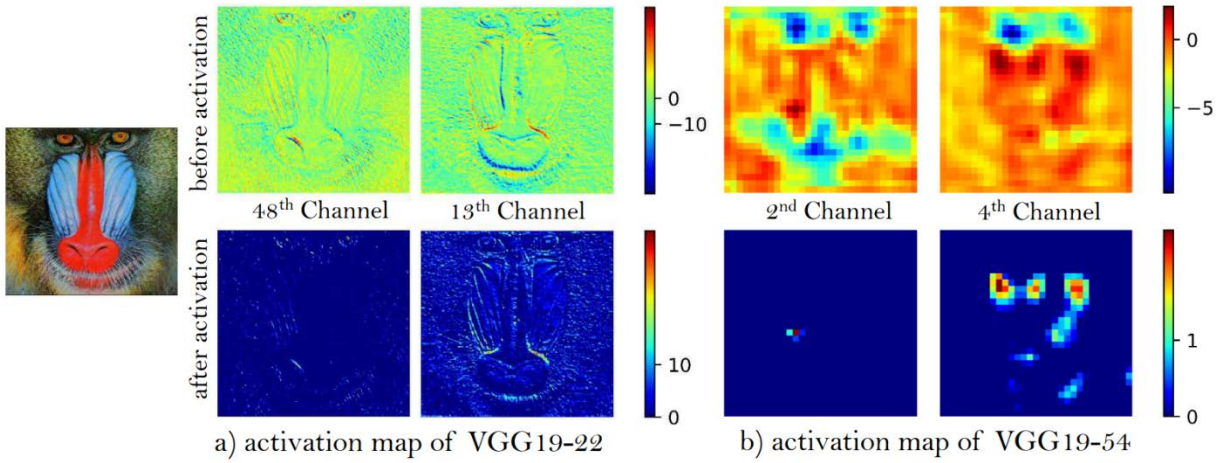


Figure 5.21: Illustration des paramètres activés par Relu dans le réseau VGG

Comme peu de neurones sont activés, une quantité d'information très réduite est alors traitée par la fonction de pertes VGG. La solution proposée consiste alors à calculer la fonction de pertes non plus après, mais avant l'activation RELU, pour préserver au maximum les informations disponibles.

Finalement, afin de bien estimer le gain apporté par chacune des modifications apportées par rapport au SRGAN, un comparatif visuel entre chaque étape est proposé (Figure 5.22).

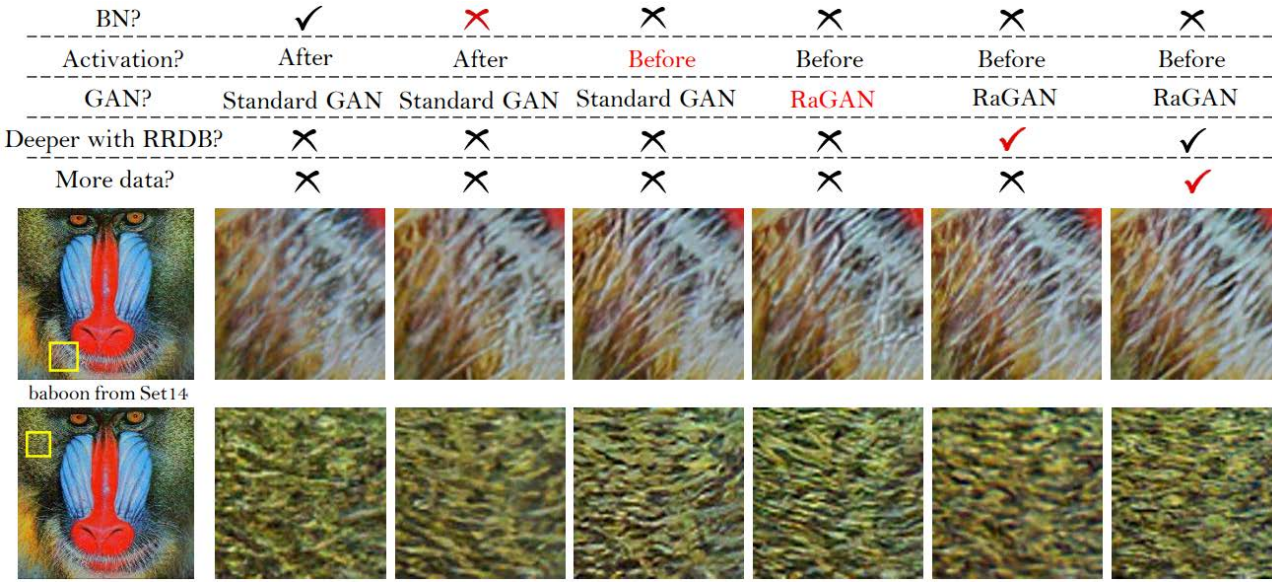


Figure 5.22 : Illustration du gain apporté par chaque étape du passage de SRGAN à ESRGAN [Wang18b]

Institut Polytechnique de Paris
91120 Palaiseau, France

Dans la mesure où les résultats obtenus par le ESRGAN sont meilleurs en tous points à ceux obtenus par son prédécesseur, nous avons pris le parti d'appliquer directement ce nouveau réseau à notre méthode plutôt que de continuer à améliorer nous-même notre version du SRGAN. Ce choix porte également sur l'implantation proposée du ESRGAN, cette fois-ci codée avec le *framework* PyTorch, qui s'avère être bien plus souple et aisée à manipuler que celle en TensorFlow du SRGAN.

Les résultats obtenus sont détaillés dans les sections suivantes.

5.5.2 Utilisation du ESRGAN : aspects pratiques

La manipulation et l'utilisation du ESRGAN pose quelques différences par rapport à celle du SRGAN.

La majeure différence réside dans le corpus d'entrainement à utiliser. En effet, dans son fonctionnement, le SRGAN prenait à chaque *epoch* des images et isolait des patches aléatoires et de taille prédéfinie. De cette manière, le réseau SRGAN supporte des images de résolutions différentes, à partir du moment où leurs dimensions étaient supérieures à la taille du *patch* considéré. Cette facilité nous a notamment permis d'utiliser le corpus « Mirflickr 20k ».

Au contraire, le réseau ESRGAN, dans son implantation initiale, ne supporte pas des images à résolutions différentes. La taille des images se trouve, dans ce cas, limitée à 480×480 pixels pour les *patchs* en haute résolution. Pour cette raison, nous avons suivi les recommandations des auteurs et utilisé les bases d'images DIV2K et Flickr2K pour nos expérimentations.

Ces deux corpus sont composés de respectivement 800 et 1200 images, présentant des contenus variés, en très haute résolution (4k environ). Pour utiliser ces images, il est donc nécessaire de les subdiviser en sous-patches de taille 480×480 pixels. Pour des raisons techniques, nous avons dû nous limiter à un corpus composé de 1100 images dont 800 de Div2k et 300 de Flick2k.

Ainsi, nous avons obtenu un corpus d'entrainement HR composé de 44000 sous-images de 480×480 pixels issues des 1100 citées précédemment. Nous nommerons ce corpus « DiFli1100 ».

D'après les auteurs, l'utilisation d'un modèle pré-entrainé avec une fonction de pertes de type PSNR permet au réseau d'obtenir de meilleures performances. Dans cette optique, un modèle pré-entrainé dans ces conditions pour de la SR ×4 est donc proposé par les auteurs.

Dans toutes les expérimentations concernant le ESRGAN que nous allons traiter dans les sections suivantes, la structure du générateur et du discriminateur seront exactement les mêmes que celles présentées dans le papier original, aux couches d'agrandissement, situées en fin de réseau, près.

Le nombre de blocs de type RRDB présents dans le générateur sera par défaut de 23.

Dans les sections suivantes, la taille des *batchs* utilisés lors des entrainements va varier de 8 à 16. Il est à noter que dans le cadre de nos travaux, le seul impact de la taille du *batch* concerne uniquement le temps d'entrainement qui en résulte.

5.5.3 Reconstruction ×4 par ESRGAN

Notre première expérience a été réalisée pour effectuer une Reconstruction ×4 car il s'agissait des paramètres de base du réseau et qu'un modèle pré-entrainé, orienté PSNR, pour de la SR ×4 était proposé.

Le corpus d'entrainement BR a été réalisé dans les mêmes conditions que pour la Reconstruction ×4 avec le réseau SRGAN. Ainsi, la résolution de toutes les images HR a été divisée par 4 par interpolation bicubique et compressée en BPG avec un facteur de compression q choisi aléatoirement entre 15 et 25.

Les paramètres utilisés ici sont : une taille de *batchs* de 8, une taille de blocs HR de 128×128, un taux d'apprentissage pour le générateur et le discriminateur de 10^{-4} et l'optimisateur ADAM avec un paramètre β_1 de 0.9. L'entrainement a été réalisé sur 5×10^5 itérations, ce qui représente 90 *epochs* sur l'ensemble de notre corpus.

La fonction de pertes utilisée est définie comme décrit dans l'équation (5.3), basée sur l'équation (5.2) :

$$L_G = L_{percept} + 5.10^{-3} L^{Ra_G} + 1.10^{-2} L_1 \,. \tag{5.3}$$

Nous pouvons noter que seulement 90 *epochs* ont été réalisées pour l'entrainement du réseau ESRGAN. En effet, l'objectif de cet entrainement est principalement de tester le réseau et d'avoir un premier aperçu des résultats obtenus. Concernant le temps d'entrainement, il est 7 fois plus réduit que celui réalisé avec le SRGAN pour 2000 *epochs* (avec une taille de *batchs* de 8 sur le corpus Mirflickr20k).

Institut Polytechnique de Paris
91120 Palaiseau, France

Quelques premiers exemples de résultats obtenus sont illustrés Figure 5.23.

Figure 5.23 : Illustration des résultats obtenus par Reconstruction ×4 avec ESRGAN

Nous pouvons observer que la Reconstruction ×4 offre des résultats intéressants, notamment au niveau des régions d'image hautement texturées, comme dans le cas des forêts, rivières, montagnes et fleurs. En revanche, dès qu'il s'agit d'éléments plus petits et ne présentant pas de texture particulière comme les visages et les écritures, le résultat devient très dégradé, voire inexploitable.

Une première comparaison des résultats obtenus par les réseaux ESRGAN et SRGAN est illustrée Figure 5.24.

Institut Polytechnique de Paris
91120 Palaiseau, France

SRGAN x4 modifié ESRGAN x4

Figure 5.24 : Comparaisons des résultats de Reconstruction ×4 par SRGAN modifié et ESRGAN

Les résultats obtenus par ESRGAN apparaissent comme bien plus nets et plus agréables à regarder que ceux obtenus par son prédécesseur. De plus, les problèmes de flous intempestifs ont disparu avec ce nouveau modèle, rendant ce dernier envisageable dans le cadre de problématiques de compression, contrairement au SRGAN ×4.

La Reconstruction ×4 par ESRGAN, bien qu'offrant des résultats bien plus intéressants que celle obtenue par SRGAN, ne parvient malheureusement à pallier son défaut majeur, qui concerne le manque de fidélité du contenu reconstruit. Les causes s'avèrent être les mêmes que précédemment, les détails fins n'existant quasiment plus dans la version basse résolution et compressée de l'image BR. Dans ce cas, le réseau, comme pour le cas de la compression générative à très bas débit, ne peut alors qu'estimer une reconstruction probable et visuellement plaisante, mais sans la moindre garantie de restituer la texture, le motif ou bien le caractère écrit.

Notre expérience nous a montré qu'augmenter le nombre d'*epochs* est inutile et ne parvient pas à corriger cette limitation.

Pour répondre au besoin de fidélité des images reconstruites, nous nous sommes donc orientés, comme dans le cas du réseau SRGAN, vers une Reconstruction ×2, décrite dans la section suivante.

Institut Polytechnique de Paris
91120 Palaiseau, France

5.5.4 Reconstruction ×2 par ESRGAN

Comme dans l'expérimentation précédente, nous avons tout d'abord réalisé un premier entrainement avec les paramètres par défaut du réseau ESRGAN, afin d'avoir un premier aperçu des résultats que nous pouvons obtenir.

Notre premier entrainement a donc été réalisé également sur 500 000 itérations avec une taille de batch de 8, ce qui représente 90 *epochs* sur le corpus retenu. Les hyperparamètres et la fonction de pertes utilisés sont les mêmes que ceux cités dans la section précédente.

Concernant les images BR, la résolution des images du *dataset* HR a été divisée par 2 par interpolation bicubique, et ces images ont ensuite été compressées en BPG avec un facteur de compression q choisi aléatoirement entre 25 et 35 pour chaque image, comme nous l'avions fait dans le cas de la Reconstruction ×2 par SRGAN.

Quelques exemples de résultats obtenus sont illustrés Figure 5.25 (ESRGAN *versus* BPG seul) et Figure 5.26 (ESRGAN *versus* SRGAN).

Reconstruction ×2 à 90 epochs — BPG seul

Figure 5.25 : Comparaison entre Reconstruction ×2 par ESRGAN à 90 *epochs* et compression BPG seule à débit équivalent

ESRGAN ×2 à 90 *epochs* — SRGAN ×2 modifié à 2000 *epochs*

Figure 5.26 : Comparaison à 0,33 bpp entre Reconstruction ×2 (à 90 *epochs*) par ESRGAN et par SRGAN modifié à (à 2000 *epochs*)

Institut Polytechnique de Paris
91120 Palaiseau, France

Ici, nous pouvons constater que la reconstruction obtenue par ESRGAN apparait comme bien plus détaillée et agréable à regarder, et ce, malgré un entrainement 7 fois plus court par rapport à son équivalent SRGAN. Ces premières observations confirmant la supériorité du réseau ESRGAN, nous avons poursuivi nos recherches pour le cas de la Reconstruction ×2 avec ESRGAN.

A des fins d'évaluation que nous détaillerons au Chapitre 6, nous avons ainsi effectué un deuxième entrainement, dans les mêmes conditions que celles que nous venons de décrire, mais avec une fonction de pertes différente, qui est dans ce cas une EQM pure, sans composante perceptuelle. Quelques exemples de résultats comparatifs sont illustrés Figure 5.27

Perte EQM pure

Perte perceptuelle mixte

Perte EQM pure

Perte perceptuelle mixte

Figure 5.27 : Comparaison entre Reconstruction ×2 par ESRGAN en utilisant une fonction de pertes EQM pure contre une fonction de pertes perceptuelle mixte

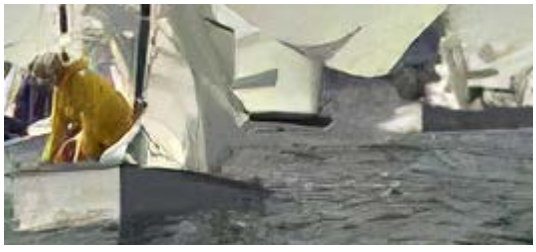
Ces résultats nous montrent que l'utilisation d'une fonction de pertes de type EQM limite les artéfacts liés à un en entrainement réalisé sur une fonction de pertes perceptuelle. En contrepartie, cela conduit à un flou excessif et un manque de détails. Ainsi, le résultat obtenu via la fonction de pertes perceptuelle demeure bien plus intéressant d'un point de vue visuel.

Afin d'étudier l'influence du nombre d'*epochs* d'entrainement sur les résultats de la reconstruction, nous avons réalisé un entrainement bien plus long, sur 1000000 d'itérations et avec une taille de *batchs* de 16. Sur le corpus considéré, cela correspond à un nombre 360 *epochs*. Les résultats obtenus sur une même image avec réseaux respectivement entrainés avec 90 et 360 *epochs*, sont illustrés Figure 5.28. Pour une meilleure visualisation, les images sont présentées ici à différents niveaux de zoom.

90 *epochs*, zoom × 740%

360 *epochs*, zoom × 740%

90 époch, zoom ×1



360 époch, zoom ×1

Figure 5.28 : Illustration des différences entre reconstruction d'une même image avec réseaux entrainés à 90 et 360 epochs à différents niveaux de zoom

Institut Polytechnique de Paris
91120 Palaiseau, France

Les résultats obtenus apparaissent comme strictement équivalents à 90 et à 360 *epochs*, et un très gros niveau de zoom est nécessaire pour réellement distinguer les différences. Même si l'image issue des 360 *epochs* présente des arêtes très légèrement plus marquées, les différences restent quasiment invisibles à l'œil nu. Nous concluons qu'il n'est pas nécessaire d'augmenter le nombre d'*epochs* au-delà de 90.

L'approche de Reconstruction ×2 par ESRGAN apparait comme bien plus performante que son homologue par SRGAN, et ce selon tous les critères de qualité visuelle ciblés. De plus, elle permet de palier aux problèmes majeurs relevés dans la Reconstruction ×4 par ESRGAN, et apparait comme bien plus fidèle par rapport au contenu original, en respectant les textures et en nécessitant un temps de traitement réduit.

Cela est illustré également Figure 5.29, pour une image présentant des zones de texte.

Reconstruction ×2 Reconstruction ×4

Figure 5.29 : Illustration de la différence de rendu de détails écrits par Reconstruction ×2 et ×4 par ESRGAN.

Ici, le texte est clairement dégradé par la version ×4 alors qu'il est clairement lisible dans la version ×2.

De par ses caractéristiques et avantages offerts, la Reconstruction ×2 s'impose donc naturellement comme la solution à privilégier.

Toutefois, dans le cas où des informations critiques doivent être transmises, comme les éléments d'une carte d'identité, elle n'atteint pas encore le niveau de fidélité et de fiabilité attendu. Ce problème est illustré Figure 5.30, où l'on constate que les éléments d'information de la carte d'identité n'ont pas pu être reconstitués avec succès.

Le besoin principal exprimé par l'entreprise partenaire Be-Bound, consistait en la transmission de cartes d'identité à très bas débit, pour les applications décrites dans l'introduction. C'est pour cette raison que, tout au long de ce manuscrit et du développement de cette thèse, nous avons mis l'accent sur le côté « fidélité » des méthodes.

Figure 5.30 : Exemple d'un cas où la Reconstruction x2 par ESRGAN n'est pas adaptée

Nous allons donc nous pencher, comme nous l'avions fait avec notre SRGAN modifié, sur la Reconstruction ×1 par ESRGAN, pour investiguer son niveau de fidélité de restitution de contenus. Cette approche est décrite dans la section suivante.

5.5.5 Reconstruction ×1 par ESRGAN

Pour les expérimentations qui vont suivre, la taille des patches HR a été modifiée de 128×128 à 64×64, dans l'optique, d'une part, de mieux les faire correspondre à la taille de *Transform Blocs* de la compression BPG et, d'une autre part, pour limiter le temps de traitement des images.

Comme première expérience nous avons entrainé notre réseau pour réaliser une Reconstruction ×1 générique par ESRGAN.

5.5.5.1 Reconstruction ×1_gen par ESRGAN

Notre première expérience s'inscrit dans la lignée de celles réalisées avec les autres versions du ESRGAN présentées. Nous avons donc choisi d'entrainer une version générique de notre Reconstruction ×1 par ESRGAN sur notre corpus « DiFli1100 » (cf. Section 5.5.2). Les images gardent cette fois leur résolution originale et sont compressées en BPG avec un facteur de compression q choisi aléatoirement entre 35 et 45, en suivant la même procédure mises en place pour l'approche SRGAN ×1 modifiée (*cf.* Section 5.4). Nous appellerons cette méthode Reconstruction ×1_gen par ESRGAN.

L'entrainement a été ici réalisé sur 2000000 d'itérations avec une taille de *batchs* de 8, ce qui correspond à un nombre de 360 *epochs* sur le corpus considéré. Quelques premiers exemples de résultats obtenus, en comparaison avec une compression BPG seule, sont présentés Figure 5.31, pour des images relativement peu texturées, et Figure 5.32, pour des images hautement texturées.

Institut Polytechnique de Paris
91120 Palaiseau, France

BPG seul, q=38

Reconstruction ×1_gen par ESRGAN

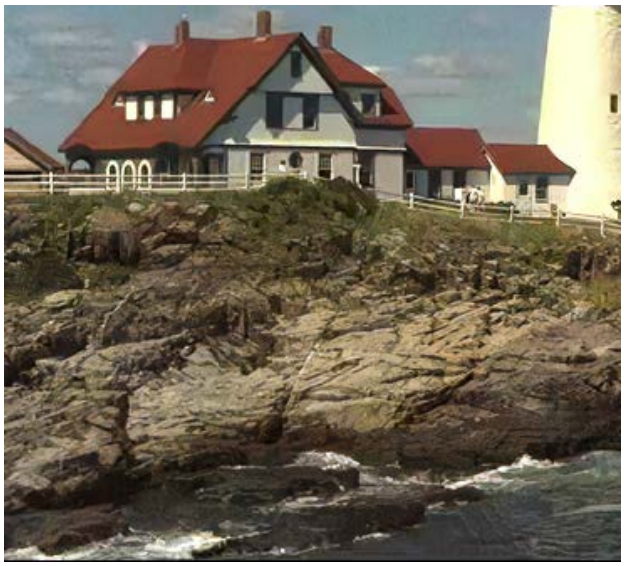
BPG seul, q=44

Reconstruction ×1_gen par ESRGAN

Figure 5.31 : Comparaison entre BPG seul et Reconstruction ×1_gen par ESRGAN sur des zones peu texturées

Institut Polytechnique de Paris
91120 Palaiseau, France

BPG seul, q=38



Reconstruction ×1_gen par ESRGAN

BPG seul, q=43

Reconstruction ×1_gen par ESRGAN

Figure 5.32 : Comparaison entre BPG seul et Reconstruction ×1_gen par ESRGAN sur des zones hautement texturées

Ces résultats nous montrent que la Reconstruction ×1_gen fonctionne correctement à la fois sur des contenus texturés et non-texturés et à des taux de compressions variant au sein de la plage utilisée pour l'entrainement, soit pour un q entre 35 et 45.

En outre, la Reconstruction ×1_gen fournit une nette amélioration par rapport au format BPG pour les (très) bas débits (entre 0,1 et 1 bpp).

Cette amélioration était déjà présente pour le cas équivalent (Reconstruction ×1_gen) réalisée avec le réseau SRGAN. Comparons donc les résultats issus de ces deux modèles (Figure 5.33).

Institut Polytechnique de Paris
91120 Palaiseau, France

Par ESRGAN, q=36

Par SRGAN, q=36

Par ESRGAN, q=40

Par SRGAN, q=40

Figure 5.33 : Comparaison entre les Reconstruction ×1_gen obtenues par ESRGAN et SRGAN à partir de la même image compressée en BPG

Les résultats obtenus par les deux modèles SRGAN et ESRGAN apparaissent comme quasi-équivalents. Toutefois, nous pouvons identifier certaines différences, peu perceptibles mais présentes.

Ainsi, la reconstruction par ESRGAN permet une meilleure fidélité quant à la colorimétrie (même si cela est en réalité quasi imperceptible) des résultats, avec une restitution des détails et de textures plus précises. De plus, l'approche ESRGAN ne fait pas apparaitre les artéfacts de type « chair de poule » propres à la Reconstruction ×1_gen par SRGAN. En revanche, comme nous pouvons le voir sous le mot « Bahamas » de la Figure 5.33, la solution par SRGAN semble elle corriger plus fortement les artéfacts issus de la compression BPG sur les zones lisses. Ce phénomène est également illustré Figure 5.34. Ici, la reconstruction générée par l'approche de Reconstruction ×1_gen par SRGAN est moins fidèle mais plus agréable visuellement.

En résumé, la solution par ESRGAN offre une meilleure fidélité mais potentiellement au prix d'un résultat visuel moins agréable sur les zones lisses comme également illustré Figure 5.34.

Figure 5.34 : Exemple de Reconstruction ×1_gen par SRGAN moins fidèle mais plus agréable visuellement

Pour une meilleure visualisation des différences que nous avons citées, le lecteur est invité à se rapporter à l'Annexe 2. Images complètes de celles utilisée Figure 5.34 où nous présentons les images en taille réelle.

Enfin, il est à souligner que la solution par ESRGAN, avec l'implantation utilisée, est bien plus rapide à entrainer et utiliser (environ 3× plus rapide) et surtout plus souple et aisée de manipulation.

Comme pour la Reconstruction ×2 par ESRGAN, nous avons cherché à améliorer notre modèle. Pour cela nous avons réalisé un entrainement en utilisant une fonction de pertes L_1 pure.

Cette expérience a été réalisée en visant trois objectifs différents. Le premier est de pouvoir observer le comportement de notre modèle sans la composante perceptuelle de la fonction de pertes. Le deuxième est de se servir de ce modèle comme pré-entrainement, comme cela est recommandé dans le papier original du ESRGAN [Wang18b]. Enfin, le troisième objectif vise notamment les aspects d'évaluation nécessaires pour réaliser une comparaison objective des résultats. Nous reviendrons sur ce dernier aspect plus tard dans le manuscrit, au Chapitre 6.

Cet entrainement a été réalisé sur un nombre de1000000 d'itérations, avec une taille de *batchs* de 16, soit 360 *epochs* sur le corpus d'entrainement considéré.

Institut Polytechnique de Paris
91120 Palaiseau, France

La Figure 5.35 illustre les résultats obtenus, en comparaison avec une compression BPG seule.

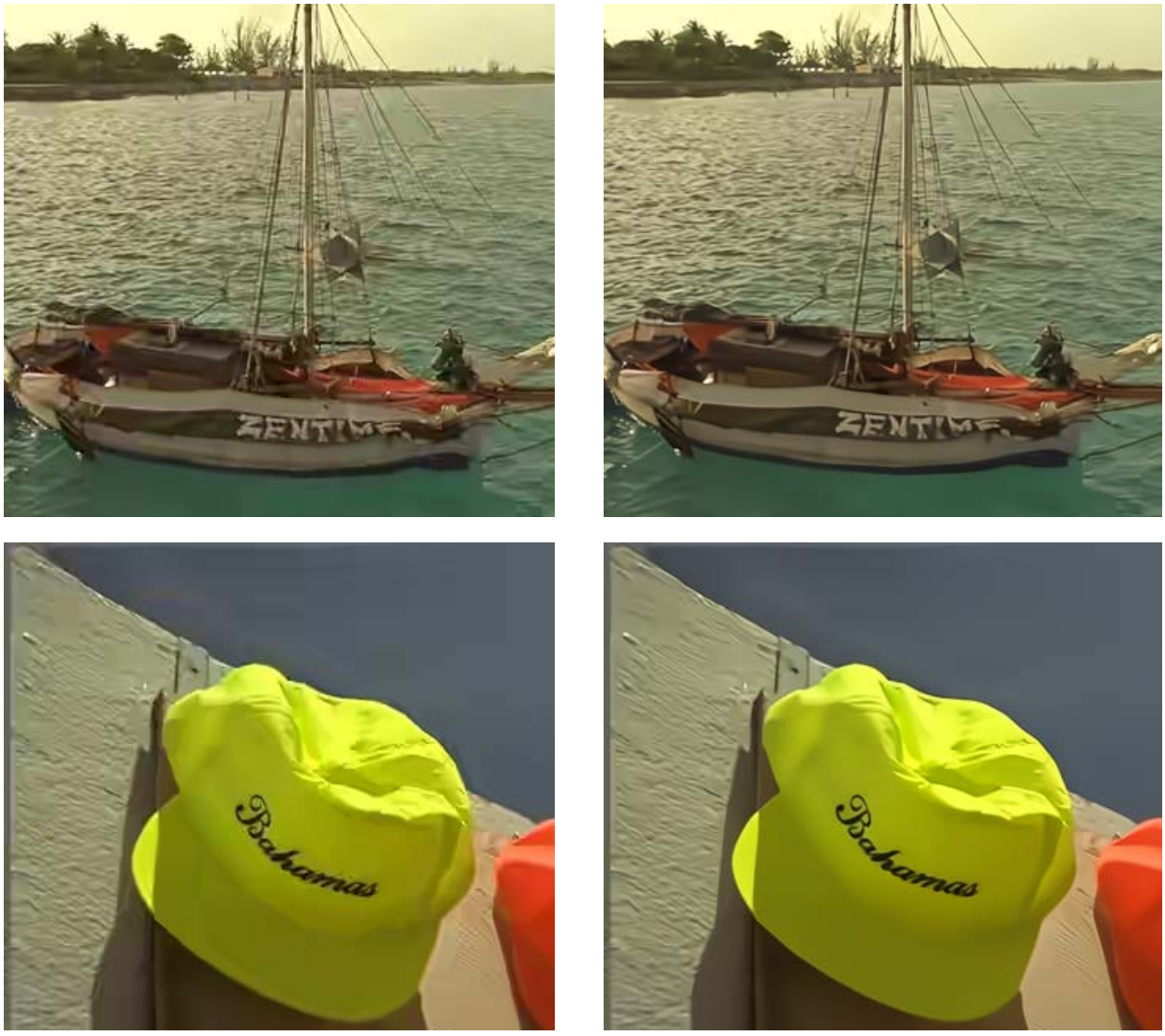


Image BPG | Reconstruction ×1_gen avec perte L_1 pure

Figure 5.35 : Comparaison entre compression BPG seule et Reconstruction ×1_gen avec perte L_1 pure sur des images présentant des niveaux de textures divers

Les résultats de la reconstruction s'avèrent satisfaisants, et montrent, de manière générale, une amélioration indéniable par rapport au BPG. Les images en taille réelle sont également disponibles en Annexe 3. Images tailles réelles illustrant les différences de qualité entre SRGAN ×1_gen (en haut) et ESRGAN×1_gen (en bas).

Bien que la Reconstruction ×1_gen avec perte L_1 semble peiner à reconstruire les textures (Figure 5.36), elle offre des résultats particulièrement intéressants pour les zones lisses. Ce résultat est d'autant plus intéressant dans la mesure où c'est sur ces parties que les reconstructions à base de fonctions de perte perceptuelles offrent les moins bons résultats, comme illustré dans la comparaison présentée Figure 5.36.

Institut Polytechnique de Paris
91120 Palaiseau, France

Avec perte perceptuelle | Avec perte L_1 pure

Figure 5.36 : Comparaison entre Reconstruction x1_gen avec pertes perceptuelle et L_1 pure

Ces comparaisons nous permettent de clairement voir les forces et faiblesses de ces deux modèles pour des problèmes de réduction d'artéfacts de compression.

Afin de tenter d'allier les résultats obtenus par ces deux modèles, nous avons considéré un nouvel entrainement qui utilise une fonction de pertes perceptuelle, appliquée sur un réseau pré-entrainé avec une perte L_1 pure. Comme pour l'expérience similaire avec la Reconstruction ×2 par ESRGAN (*cf.* Section 5.5.4), les résultats restent strictement équivalents à ceux obtenus sans pré-entrainement orienté PSNR.

La Reconstruction ×1_gen par ESRGAN offre bel et bien une amélioration par rapport à celle obtenue par SRGAN. Elle permet une bien meilleure reconstruction des textures. De plus, en considérant une fonction de pertes L_1 pure, elle conduit à une reconstruction des structures s'affranchissant des artéfacts générés par l'utilisation d'une fonction de pertes perceptuelle.

Trouver un bon compromis pour la pondération de ces deux fonctions de perte semble être une piste majeure pour optimiser nos résultats. Malheureusement, nous n'avons pas à ce jour eu le temps de déterminer cet équilibre, mais cela apparait comme un très bon point de départ pour des recherches futures.

Une autre solution envisageable serait de séparer les structures et textures d'une d'image puis de les traiter séparément. Ce principe sera abordé au Chapitre 7.

Comme le réseau ESRGAN a pu démontrer de meilleures capacités de généralisation que son prédécesseur SRGAN, en offrant notamment une meilleure Reconstruction ×1_gen, nous avons tenu à vérifier les capacités de ce nouveau réseau à effectuer une Reconstruction ×1 pour un facteur de compression unique, comme nous l'avions fait avec le SRGAN. Pour tester cela, nous avons réalisé une expérience de Reconstruction ×1_43 par ESRGAN, décrite dans la section suivante.

5.5.5.2 Reconstruction ×1_43 par ESRGAN

Dans cette expérimentation, notre objectif est de tester la capacité du ESRGAN à traiter des problèmes de RAC pour un seul et unique facteur de compression et d'observer son comportement dans ces conditions.

Pour cela, l'ensemble des images du corpus BR ont été compressées en BPG avec un facteur de compression q=43. Bien entendu, la résolution des images n'a pas été réduite ici.

L'entrainement a été réalisé sur 2000000 d'itérations avec une taille de *batchs* de 8, ce qui conduit à un nombre de 360 *epochs* sur la totalité du corpus considéré.

La Figure 5.37 illustre quelques exemples de résultats obtenus.

Figure 5.37 : Image compressée BPG avec q=43 (à gauche) et sa Reconstruction ×1_43 par ESRGAN (à droite)

Nous observons ici que le travail de RAC est effectué dans la mesure où les artéfacts caractéristiques de la compression BPG n'apparaissent quasiment plus. Néanmoins, les artéfacts générés par la reconstruction sont marqués et désagréables à regarder. Cela est d'autant plus problématique dans la mesure où les résultats obtenus pour la Reconstruction ×1_43 par le réseau précédent SRGAN sont beaucoup moins exposés à ce phénomène, comme illustré Figure 5.38 .

Institut Polytechnique de Paris
91120 Palaiseau, France

Figure 5.38 : Comparaison entre les Reconstruction ×1_43 par ESRGAN (à gauche) et par SRGAN (à droite)

Nous pouvons remarquer que la Reconstruction 1_43 par SRGAN est la meilleure. Néanmoins, il paraît curieux qu'il s'agisse pour l'instant du seul cas où le SRGAN semble apporter une meilleure solution à un problème que son successeur. Tentons donc d'investiguer un peu plus sur les raisons de ces résultats relativement étonnants.

Tout d'abord, il semble que d'un côté, la solution par SRGAN ait tendance à lisser fortement les artéfacts de BPG afin de les atténuer, alors que le ESRGAN va chercher à les compenser en rajoutant ses propres *patterns* de reconstruction. A première vue, si l'on suit les bases du *machine learning*, ce type de comportement pourrait être dû à un problème de *overfitting*. Toutefois, il est connu que les réseaux GAN sont en général peu soumis à ce problème [Goodfellow16]. En outre, le corpus d'entrainement utilisé est suffisamment large et varié en termes de contenus.

Pour investiguer néanmoins de manière plus approfondie cette hypothèse, nous avons étudié l'évolution des résultats obtenus au fil des *epochs*. Cela est illustré Figure 5.39.

Institut Polytechnique de Paris
91120 Palaiseau, France

10 *epochs* 20 *epochs* 100 *epochs*

Figure 5.39 : Evolution des résultats de la Reconstruction x1_43 par ESRGAN à différentes *epochs*

Ces résultats nous montrent qu'à 10 *epochs* le modèle n'est pas finalisé, puisqu'à seulement 20 *epochs* les artéfacts gênants commencent déjà à apparaitre. Cela écarte un peu plus l'hypothèse de l'*overfitting*.

Nous avons par la suite cherché à vérifier si ces artéfacts n'étaient pas dus à une trop forte dominance de la composante perceptuelle de la fonction de pertes. Pour cela, nous avons entrainé notre modèle avec une fonction de pertes L_1 pure. Les résultats obtenus sont illustrés Figure 5.40.

Figure 5.40 : Résultats obtenus pour la Reconstruction x1_43 par ESRGAN avec perte L_1

Institut Polytechnique de Paris
91120 Palaiseau, France

Nous pouvons observer ici les mêmes tendances que nous avions constatées pour la Reconstruction ×1_gen par ESRGAN. En effet, nous obtenons de très bons résultats sur les zones lisses/structurés. Sur ces régions, les artéfacts de *blocking* et de *ringing* de la compression BPG sont quasiment supprimés, comme illustré davantage Figure 5.41.

Image BPG avec q=43 | Reconstruction ×1_43 par ESRGAN avec perte L_1

Figure 5.41 : Illustration de la suppression des artéfacts de *blocking* et *ringing* du BPG par la Reconstruction ×1

Néanmoins, bien que beaucoup d'artéfacts se trouvent être bien supprimés par cette reconstruction, les artéfacts de « flou » engendrés par BPG ne sont que très peu atténués, ce qui rend les zones de textures très peu agréables à regarder malgré la reconstruction (Figure 5.42).

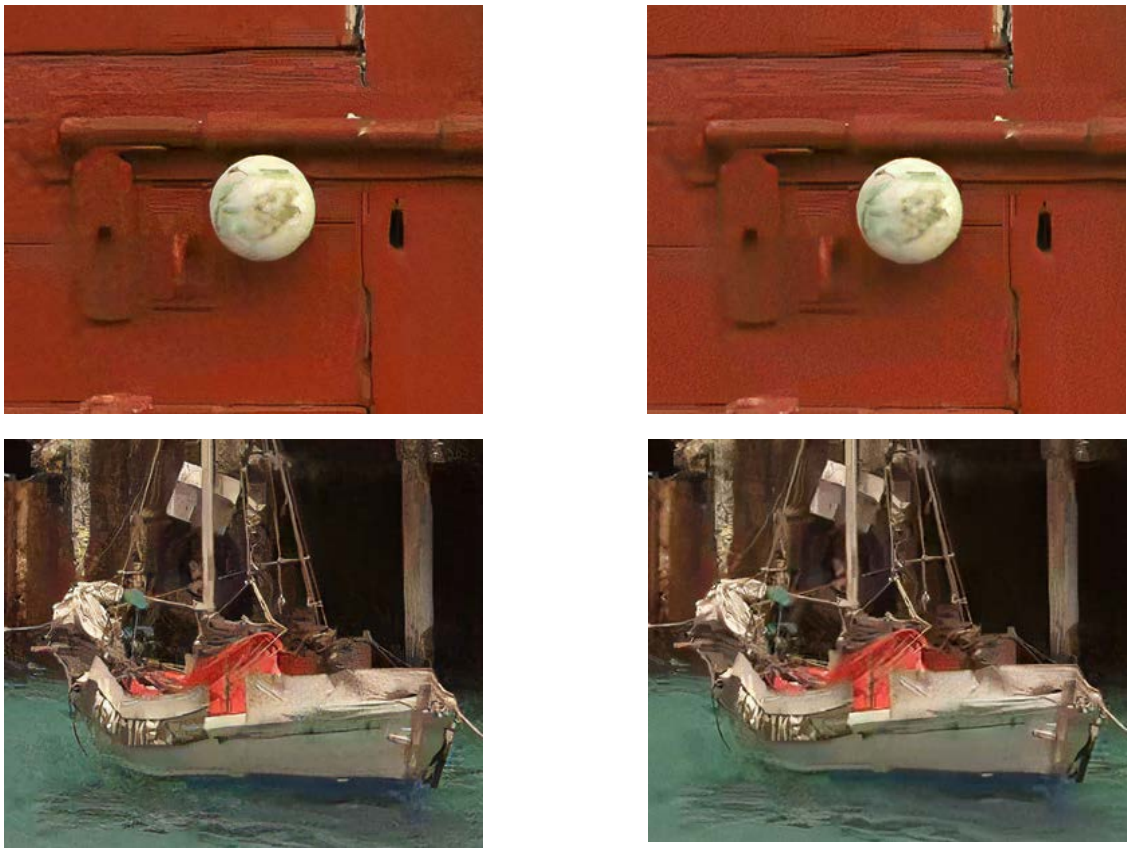
Image BPG avec q=43 | Reconstruction ×1_43 par ESRGAN avec perte L_1

Figure 5.42 : Illustration du manque d'efficacité de la Reconstruction ×1_43 par ESRGAN avec perte L_1 sur des zones texturées

Nous nous retrouvons ainsi avec les mêmes conclusions que pour la Reconstruction ×1_gen où il semble nécessaire de trouver un compromis au niveau de la pondération entre la composante de perte perceptuelle et la composante L_1. Toutefois, dans le cas de l'approche de Reconstruction ×1_43 les artéfacts générés par l'entrainement avec perte perceptuelle se trouvent être bien plus gênants que pour la version ×1_gen.

Institut Polytechnique de Paris
91120 Palaiseau, France

Afin de déterminer s'il est pertinent de pousser le concept de Reconstruction ×1_43 par ESRGAN, comparons là, pour des images compressées en BPG à q=43, à sa Reconstruction ×1_gen (Figure 5.43).



Reconstruction ×1_43 par ESRGAN | Reconstruction ×1_gen par ESRGAN

Figure 5.43 : Comparaison entre les Reconstruction ×1_gen et _43 par ESRGAN à q=43

Les différences entre les deux reconstructions sont très mineures et donc très peu perceptibles à l'œil nu. Nous pouvons en déduire que le ESRGAN avec une fonction de pertes perceptuelle permet une assez bonne capacité de généralisation (contrairement au SRGAN modifié) au point de se permettre d'utiliser la version générique de la Reconstruction x1 dans tous les cas de compression à très bas débit par BPG.

Ainsi, suite à ces observations nous avons décidé de stopper les expérimentations concernant la Reconstruction ×1_43 par ESRGAN au profit de la version ×1_gen.

Institut Polytechnique de Paris
91120 Palaiseau, France

5.5.6 Conclusion

Pour conclure sur nos expérimentations, il apparait que les reconstructions réalisées par le réseau ESRGAN sont plus performantes et ce en tous points à celles obtenues par SRGAN. Dans la mesure où l'approche ESRGAN est une version optimisée du SRGAN, cela n'est pas tout à fait surprenant. En outre, malgré le fait que le corpus d'images utilisé pour entrainer le réseau ESRGAN présente une moins grande variété de contenus et de qualités que celui utilisé pour le SRGAN, ESRGAN arrive à montrer une meilleure capacité de reconstruction et de généralisation, tout en nécessitant un temps d'entrainement et de traitement bien inférieur par rapport au SRGAN.

Notons aussi que toutes nos expérimentations ont été réalisées avec, dans l'ensemble, les paramètres par défaut proposés et avec un seul corpus d'entrainement. Une marge d'amélioration importante nous semble donc possible, en affinant notamment les paramètres, hyperparamètres et corpus d'entrainement.

Concernant les applications, il convient de déterminer quel type de reconstruction correspond à quel type d'image à traiter. D'entrée, nous pouvons exclure les reconstructions en SR ×4, dans la mesure où elles n'apportent pas une qualité suffisante par rapport à la perte de fidélité engendrée.

La Reconstruction ×2 quant à elle, offre de performances bien plus intéressantes. En effet, même si la fidélité de la reconstruction pour les détails n'est pas encore suffisante, elle apporte une très bonne qualité de reconstruction pour les structures et textures et ce pour un temps de traitement réduit. De cette manière nous pouvons statuer qu'il s'agit de la reconstruction à recommander par défaut dès que l'on souhaite transmettre un contenu ne présentant pas d'information critique.

La Reconstruction ×1 quant à elle a su démontrer son efficacité et sa fiabilité. Bien que les capacités de compression soient légèrement inférieures à celles de la Reconstruction ×2, elle reste très performante, tout en offrant les meilleures performances en termes de fidélité des contenus reconstruits par rapport aux Reconstructions ×2 et ×4. Elle offre ainsi la piste la plus prometteuse pour utilisation dans le cas des contenus critiques, où il est indispensable de préserver l'information présente dans les images. Le prix à payer est lié en revanche à la complexité de calcul plus élevée, les *patchs* en basse résolution étant ici de la même taille que ceux en résolution initiale.

Pour l'heure, ces conclusions s'appuient uniquement sur une évaluation qualitative, purement visuelle des résultats obtenus. A l'évidence, ce seul critère ne présente aucunement une validation fiable de nos résultats. Nous consacrerons donc le prochain chapitre à une évaluation beaucoup plus approfondie de nos résultats, en prenant en compte des critères à la fois objectifs et subjectifs, tout en détaillant l'importance de ces deux termes.

Chapitre 6. Évaluation expérimentale

Résumé. Nous proposons dans ce chapitre une évaluation à la fois objective et subjective de nos résultats. Tout d'abord, nous introduisons trois grandes notions de réalisme définissant la qualité d'une image. Ces trois notions sont le **réalisme physique**, le **photoréalisme** et le **réalisme fonctionnel**.

Nous présentons ensuite une première évaluation de nos résultats à partir de métriques objectives. Les résultats objectifs obtenus démontrent le manque de pertinence de ces métriques dans le cadre de nos travaux. Ainsi, des images reconstruites qui présentent à l'évidence une qualité visuelle supérieure sont significativement pénalisées par les scores PSNR et SSIM obtenus. Ce problème s'avère être caractéristique aux approches de reconstruction par réseaux GAN, qui fournissent des images globalement plausibles et visuellement agréables mais sans respecter des critères de fidélité pixel à pixel.

Pour valider nos approches, nous mettons alors en œuvre un protocole d'évaluation subjective, répondant à la norme internationale UIT-R BT.500-13, mieux adapté à la bonne évaluation de nos résultats vis-à-vis de leur utilisation réelle.

Ces évaluations subjectives effectuées sur trois modèles différents, incluant réseaux SRGAN et ESRGAN, avec différents types d'entrainements et de fonctions de perte. Les résultats obtenus, présentés et discutés en détail confirment globalement la supériorité, en termes de scores MOS, des approches de reconstruction par réseaux GAN par rapport à une simple compression BPG. L'impact des paramètres et conditions d'entrainement sur les performances est également analysé.

Mots clés **:** photoréalisme, évaluation objective, évaluation subjective, PSNR, Mean Opinion Score

Institut Polytechnique de Paris
91120 Palaiseau, France

Dans le chapitre précédent, nous avons fait état des différents résultats obtenus lors de nos expérimentations, pour l'ensemble des modèles retenus, tout en proposant une comparaison visuelle qualitative de nos résultats, qui nous ont permis de tirer de premières conclusions et d'identifier les approches les plus prometteuses. Toutefois, aucune évaluation expérimentale protocolaire n'a été donnée lors de la description de nos expérimentations.

Dans ce chapitre, nous allons notamment proposer une évaluation expérimentale rigoureuse de nos résultats.

En premier lieu, nous proposons une discussion portant sur les différentes approches d'évaluation de la qualité d'image, qui reste aujourd'hui un sujet difficile et hautement controversé.

Traditionnellement, et comme dans la plupart des domaines comme la vision par ordinateur ou l'apprentissage statistique, l'utilisation de métriques mathématiques est privilégiée pour évaluer objectivement les résultats.

Par exemple, dans le domaine de la reconnaissance d'objets, la métrique la plus adaptée semble être de déterminer le pourcentage d'objets correctement reconnus (tout en prenant en compte les faux positifs) par rapport à une vérité terrain, supposée disponible.

Pour mesurer la qualité d'une image traitée/compressée, l'opération est moins évidente. Cela est dû à l'impossibilité de retranscrire de manière mathématique le système de vision/perception visuelle humain, ainsi qu'aux aspects subjectifs et affectifs propres aux jugements émis par les humains.

Une solution simple consiste à mesurer la distance pixel par pixel entre une image traitée et l'image originale. On peut alors imaginer que si les pixels traités sont proches de ceux de l'image originale, le résultat sera satisfaisant et donc peu altéré d'un point de vue perceptif. C'est le principe même des scores PSNR ou des distances EQM, introduits au Chapitre 3.

Le score PSNR permet une évaluation très simple et rapide des résultats obtenus, en caractérisant un certain niveau de qualité d'une image et demeure encore aujourd'hui une des métriques les plus adoptées dans l'état de l'art.

Malgré cela, en pratique, cette métrique reste assez peu corrélée avec la perception humaine, comme démontré dans [Chikkerur11]. La différence entre les résultats traduits par des métriques comme le PSNR par rapport à la réalité de la perception visuelle humaine est également détaillée dans [Ferwerda03], où la qualité d'une image est déclinée en trois grandes familles distinctes que sont le **réalisme physique**, le **photoréalisme** et le **réalisme fonctionnel**.

Le **réalisme physique** cible la fidélité pixel à pixel des valeurs d'une image et ce, tout en considérant uniquement le point de vue initial de la scène représentée et en négligeant tout aspect relatif à la vision humaine. Initialement, cette définition a plutôt été établie dans une optique d'évaluation des systèmes de visualisation de contenus visuels comme des écrans ou projecteurs. Toutefois, dans un contexte purement de vision par ordinateur, avec des vérités terrain elles-mêmes numériques, le concept de réalisme physique peut se décliner par une similitude pixel par pixel entre l'image créée et sa vérité terrain. Il est ainsi équivalent aux métriques de type PSNR ou EQM.

Institut Polytechnique de Paris
91120 Palaiseau, France

L'idée de **photoréalisme** renvoie au principe de création d'une image qui n'est pas distinguable d'une photographie représentant la même image. Cela pose notamment des soucis de définition dans la mesure où la lumière dégagée, et donc le signal, est très différent si l'on regarde une image sur un écran rétroéclairé. Dans notre cas, la notion de photoréalisme se traduit par la notion de ressemblance visuelle entre image traité et vérité terrain. Idéalement, l'image créée de doit pas être distinguable par l'œil humain de sa vérité terrain.

Enfin, le dernier volet concerne la notion de **réalisme fonctionnel**. Dans ce cadre, le principe est de cibler l'information désirée dans un contenu visuel et de ne prendre en compte dans l'évaluation que la restitution de cette information. Pour illustrer ces aspects, prenons l'exemple de la photo d'un formulaire, de qualité relativement faible, illustrée Figure 6.1.

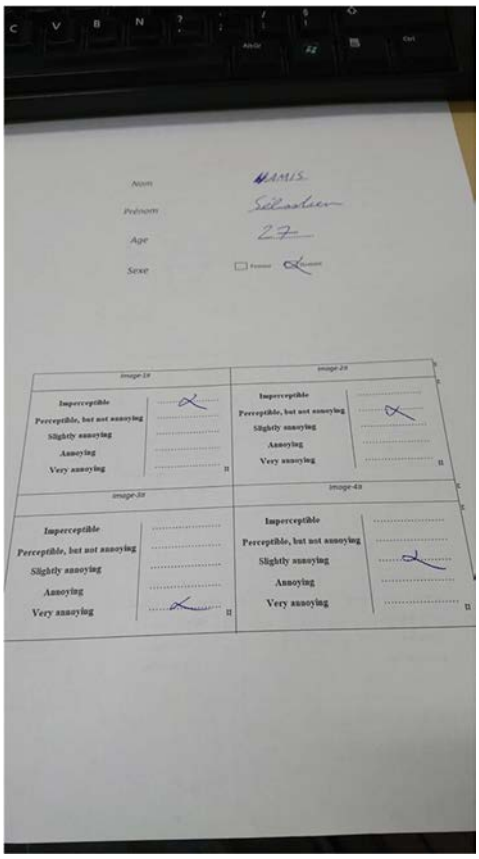
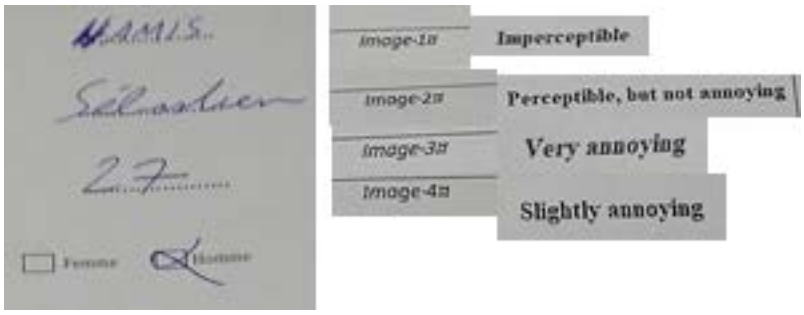


Figure 6.1 : Photo de faible qualité d'un formulaire

Dans ce cas, nous pouvons statuer que seule l'information fonctionnelle, représentant les éléments écrits présents sur les formulaires, est nécessaire et doit être restituée avec fidélité. Le réalisme fonctionnel serait alors atteint par des images simplifiées, qui contiennent intégralement les informations écrites dans le formulaire. Quelques exemples d'images qui satisfont ce critère de réalisme fonctionnel sont illustrés Figure 6.2.

Institut Polytechnique de Paris
91120 Palaiseau, France



HAMIS	Image 1 : Imperceptible
Sébastien	Image 2 : Perceptible but not annoying
27	Image 3 : Very annoying
Homme	Image 4 : Slightly annoying

Figure 6.2 : Images atteignant un réalisme fonctionnel de la photo de formulaire

Cette notion de réalisme fonctionnel peut s'appliquer de manière plus générale à d'autres éléments d'images, comme des visages ou plus généralement d'objets d'intérêt.

Compte-tenu des définitions des trois grands types de réalismes présentés plus haut, nous pouvons statuer que le réalisme que nous avons cherché à atteindre dans l'ensemble des travaux présentés jusqu'à présent concerne principalement les notions de photoréalisme (pour obtenir des reconstructions à partir d'images compressées maximisant la ressemblance perceptuelle avec l'image d'origine) et de réalisme fonctionnel (notamment pour la préservation des éléments de texte incrustés dans une image).

Dans la suite, c'est principalement dans une optique de photoréalisme que nous allons conduire nos évaluations expérimentales.

Pour ces évaluations, nous continuerons à utiliser le corpus d'évaluation Kodak [Kodak]. Les 24 images de test du corpus Kodak sont illustrées Figure 6.3.

Institut Polytechnique de Paris
91120 Palaiseau, France

Figure 6.3 : Illustration des 24 images d'évaluation du corpus Kodak

Tout d'abord, analysons les résultats obtenus en termes de métrique PSNR, présentés Figure 6.4. Le Tableau 6-1 : résume les techniques retenues pour évaluation comparée, avec différents paramètres associées et conditions d'entrainement.

Institut Polytechnique de Paris
91120 Palaiseau, France

Technique	Super-résolution	Corpus entrainement	Fonction de pertes	Paramètres
ESRGAN_×4	Oui, _×4	DiFli1100, 90 *epochs*	Hybride (L_1 + perceptuel VGG	Compression BPG des images BR avec q∈ $[15, 25]$
ESRGAN_×2_p	Oui, _×2	DiFli1100, 360 *epochs*	Hybride (L_1 + perceptuel VGG)	Compression BPG des images BR avec q∈ $[25, 35]$
ESRGAN_×2_l1	Oui, _×2	DiFli1100, 360 *epochs*	L_1	Compression BPG des images BR avec q∈ $[25, 35]$
SRGAN_×1_43	Non	Mirflickr20k 2000 *epochs*	Hybride (L_1 + perceptuel VGG)	Compression BPG des images BR avec $q = 43$
ESRGAN_×1_43_10RRDB	Non	DiFli1100, 360 *epochs*	Hybride (L_1 + perceptuel VGG	Compression BPG des images BR avec $q = 43$
ESRGAN_×1_43_l1	Non	DiFli1100, 360 *epochs*	L_1	Compression BPG des images BR avec $q = 43$
ESRGAN_×1_gen	Non	DiFli1100, 360 *epochs*	Hybride (L_1 + perceptuel VGG	Compression BPG des images BR avec q∈ $[35, 45]$
ESRGAN_×1_11	Non	DiFli1100, 360 *epochs*	L_1	Compression BPG des images BR avec q∈ $[35, 45]$

Tableau 6-1 : Techniques retenues pour évaluation comparée

Dans les deux évaluations suivantes (PSNR et SSIM), les champs « BPG ×1_43 », « BPG ×1_rand » et « BPG ×2 » correspondent à une compression BPG seule des images du corpus d'évaluation.

« BPG ×1_43 » et « BPG ×1_rand » représentent directement les images BPG avant les reconstructions évaluées correspondantes (respectivement (E)SRGAN ×1_43_xx et ESRGAN ×1_gen et _l1).

Institut Polytechnique de Paris
91120 Palaiseau, France

« BPG ×2 » représente la compression BPG seule des images du corpus d'évaluation en résolution initiale (et non sous-échantillonnées par deux). Les images « BPG ×2 » ont un débit équivalent à celui de leurs contreparties sous-échantillonnées et compressées (plus légèrement) en BPG ; ces contreparties sous-échantillonnées étant les images avant leur traitement par ESRGAN_×2_p et ESRGAN_×2_l1.

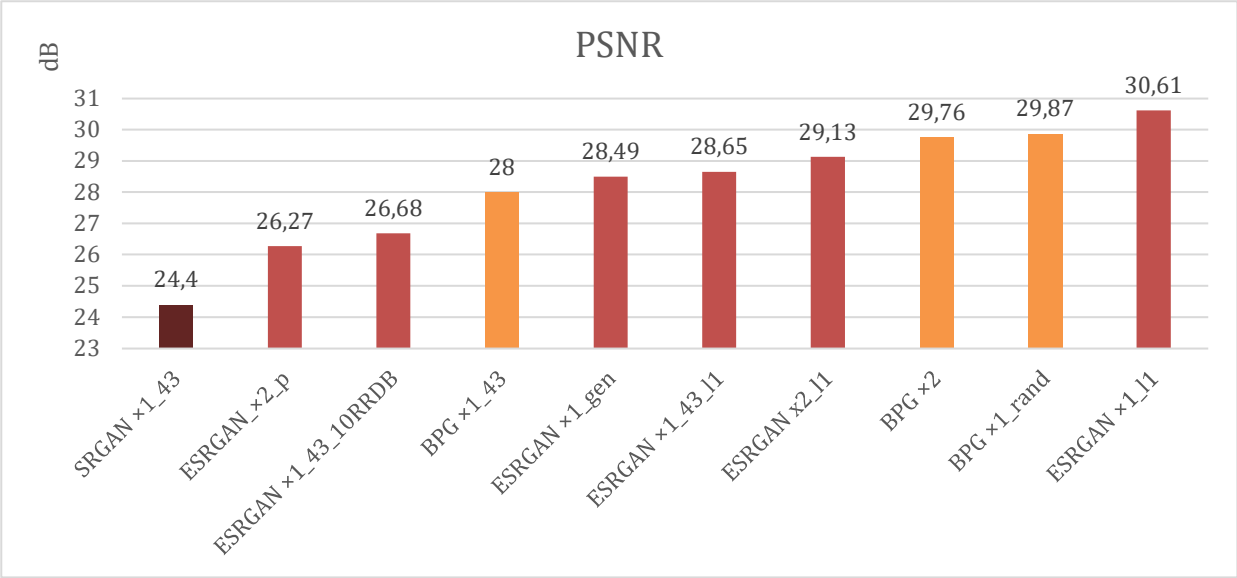


Figure 6.4 : Scores PSNR moyens sur l'ensemble du corpus d'évaluation

Les scores PSNR obtenus par les différentes reconstructions ESRGAN sont inférieurs à ceux correspondant à une compression BPG seule. Au premier abord, ce résultat semble étonnant. D'après les résultats des différentes expérimentations présentées tout au long du Chapitre 5, Comment est-il possible que les reconstructions soient visuellement meilleures que les images correspondantes compressées en BPG ?

Une première explication prend en compte les définitions des 3 types de réalisme énoncées précédemment. Ainsi, la métrique PSNR correspond bien plus à une déclinaison de réalisme physique qu'à du photoréalisme. Or, c'est principalement dans un objectif de photoréalisme que nos modèles ont été entrainés.

Un deuxième élément d'explication est lié à la conception même du format BPG. L'étape d'optimisation débit-distorsion (*cf.* Section 2.4) vise notamment à sélectionner, parmi l'intégralité des combinaisons taille de blocs/modes de prédiction possibles, celle qui maximise le score PSNR.

Nous pouvons tirer des conclusions similaires pour les modèles entrainés avec une fonction de pertes L_1. Dans ce cas, le score PSNR des reconstructions obtenues augmentent significativement.

En effet un réseau de neurones va optimiser la fonction de pertes considérée. Choisir une fonction de pertes correspondant aux critères fondamentaux d'une métrique comme la L_1 ou L_2 pour le PSNR, conduit implicitement à l'optimisation de la métrique correspondante, ce qui est confirmé par les résultats obtenus.

L'incapacité de la métrique PSNR de retranscrire la perception de qualité propre à la vision humaine a déjà été mise en lumière dans plusieurs travaux de l'état de l'art [Chikkerur11]. C'est d'ailleurs une

des raisons pour laquelle la métrique d'évaluation SSIM [Zang04] a été introduite, afin de prendre en compte les éléments structurants de l'image, auxquels la vision humaine semble être plus sensible.

Nous avons donc également réalisé une évaluation de nos résultats en considérant la métrique SSIM. Les résultats obtenus sont résumés Figure 6.5.

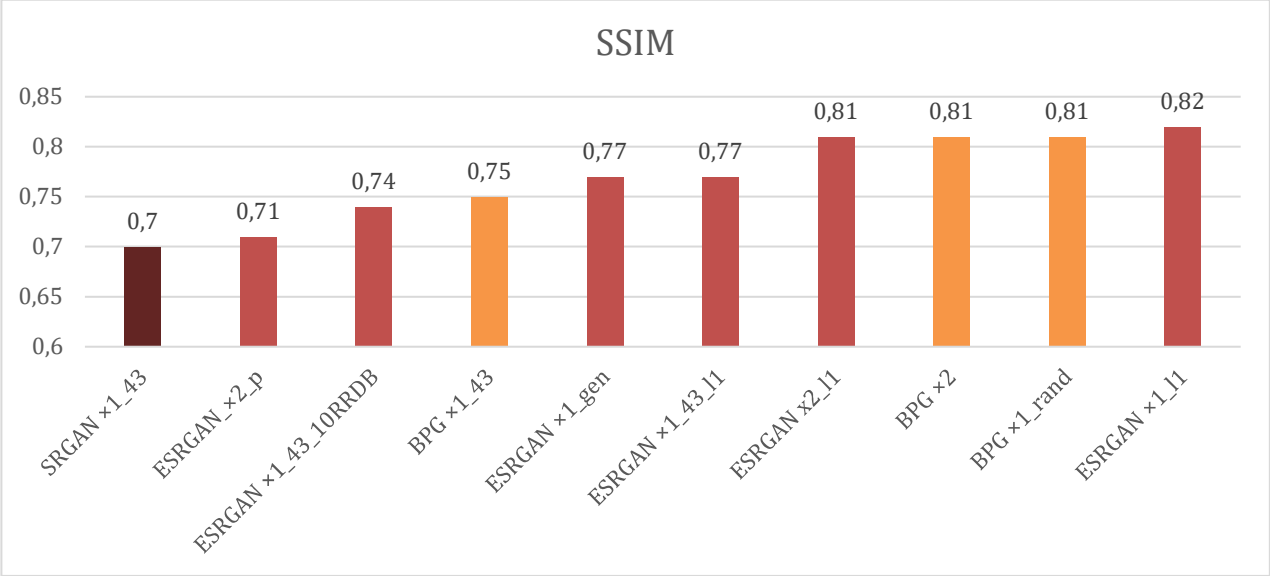


Figure 6.5 : Scores SSIM moyens sur l'ensemble du corpus d'évaluation

Nous pouvons constater une corrélation forte entre les valeurs SSIM obtenues et les scores PSNR présentés précédemment. Cette corrélation entre PSNR et SSIM est en fait tout à fait normale, dans la mesure où ces deux métriques se focalisent localement sur les valeurs des pixels, sans prendre en compte la globalité de l'image.

De cette manière, en cherchant à optimiser une image sur les valeurs des pixels uniquement, soit dans le domaine spatial initial, on peut facilement se retrouver avec de bons résultats en termes de métriques ne regardant que cette composante. Or, bien que le SSIM soit reconnu comme plus proche de la perception humaine que le PSNR, il en reste toutefois éloigné, d'après l'étude présentée dans [Chikkerur11].

La question qui se pose alors est comment prendre en compte un critère de qualité adapté à la perception humaine. La difficulté principale vient du fait que celle-ci ne semble pas être évaluable numériquement.

L'état de l'art fait apparaître une multitude de métriques objectives visant à approcher au maximum la qualité perceptuelle des images comme par exemple NIMA[Talebi18], DIIVINE[Moorthy11], M²S[Zhai05] ou même le score de Ma[Ma17] le et NIQE[Mittal12] cités plus tôt. Dans nos travaux nous ne retenons que les PSNR et SSIM car il s'agit encore aujourd'hui des mesures les plus utilisés dans la littérature.

Enfin, lors des différents challenges internationaux organisés pour évaluer les méthodes de compression ou de super-résolution, il incombe aux organisateurs de challenges de choisir parmi toutes les métriques existantes celles qui semblent le mieux correspondre à l'application visée et aux participants d'adapter leurs travaux à ce choix.

C’est également le cas du réseau ESRGAN, qui a été proposé lors du challenge PIRM-SR [Blau18]. Les auteurs spécifient explicitement que le modèle présenté pour le challenge diffère du modèle présenté dans le papier notamment par sa fonction de pertes. Cette dernière a été notamment modifiée pour le challenge afin de mieux s’adapter à la métrique sélectionnée qui était un d’index perceptuel, fonction de deux autres métriques que sont le NIQE et le score de Ma.

Comme les métriques objectives n’arrivent pas à fournir un indice de qualité fiable et fidèle à la perception humaine, dans nos travaux nous avons considéré une approche alternative, qui consiste en une évaluation subjective de qualité. Le principe est ici de demander à un panel d’évaluateurs humains de déterminer la qualité d’une image, pour dresser ensuite une analyse statistique des résultats.

Ce principe a le mérite de mettre l’humain au centre du processus d’évaluation. Néanmoins, pour s’assurer de la pertinence et de la crédibilité des résultats, il est indispensable de suivre des protocoles d’évaluation rigoureux, soumis à des règles strictes. Pour cela, des normes internationales [Union12] donnant des recommandations précises ont été spécifiées. Ces normes spécifient notamment les conditions dans lesquelles les évaluations doivent être réalisées. Cela concerne le nombre d’évaluateurs impliqués dans le processus, leur diversité en termes d’âge et de domaine d’activité, leur acuité visuelle, mais aussi les conditions d’éclairage dans la pièce dans lesquelles l’évaluation est réalisée ainsi que l’éloignement de l’évaluateur par rapport au support de visualisation des images.

Grâce notamment à ces normes, l’évaluation subjective se positionne aujourd’hui comme la métrique la plus fiable existante quant à l’évaluation de la qualité perceptuelle de contenus visuels [Wang20].

Notons toutefois que cette approche présente un certain nombre d’inconvénients et d’imprécisions. Concernant les inconvénients, il se trouve que cette méthode d’évaluation est très chronophage par rapport aux méthodes objectives qui permettent de calculer les métriques de façon automatique. C’est notamment une des raisons pour lesquels ce mode d’évaluation est encore peu adopté et que les métriques objectives continuent d’être grandement majoritaires dans la littérature.

Quant aux éventuelles imprécisions, elles sont principalement dues à des problèmes d’échelle de notation, qui peuvent être à la fois discrètes et continues. Le plus souvent, une échelle discrète est graduée de 1 à 5 et/ou associée voire remplacée par des étiquettes estampillées *Excellent*, *Bon*, *Satisfaisant*, *Médiocre*, *Mauvais.*

Les échelles continues prennent de plus souvent des valeurs entre 1 et 100 et sont aussi parfois associées à des étiquettes marquant les paliers d’évaluation.

Le problème principal avec ces valeurs et étiquettes vient du fait qu’elles peuvent autant aider la personne réalisant la notation que fausser son jugement. Des recherches et thèses [Bensaid Ghaly18] étudient ces problématiques et visent à optimiser ce genre d’évaluations et d’échelles.

Néanmoins, comme énoncé plus haut, ce système d’évaluation semble à ce jour la méthode la plus fiable pour déterminer la qualité d’une image si un panel d’évaluateurs suffisant est convoqué et qu’un nombre d’images à évaluer suffisamment important. En général, le nombre minimum d’évaluateurs est fixé à 15 d’après les recommandations de la norme UIT-R BT.500-13 [Union12] mais le nombre d’échantillons à évaluer n’est pas spécifié.

Plus précisément, bien que l'UIT demande toujours que le nombre de sujets impliqués dans les évaluations soit supérieur à 15 (par exemple, UIT-R BT.500-13), les études expérimentales font état d’une grande variabilité de ce paramètre. Il apparait donc comme compliqué de démontrer l'influence théorique de la taille du panel d'observateurs.

Institut Polytechnique de Paris
91120 Palaiseau, France

Nous avons donc décidé de valider une partie de nos résultats en utilisant un protocole d'évaluation subjective. Nous avons pour cela choisi de suivre la norme UIT-R BT.500-13.

La norme UIT-R BT.500-13 spécifie de nombreux protocoles d'évaluations. Classiquement, pour évaluer la qualité d'images compressées/traitées il est demandé aux utilisateurs de donner une note, suivant les échelles décrites précédemment, à des contenus visuels qui leurs sont présentés. Parmi ces contenus, il y a à la fois des images traitées et non traitées.

L'inconvénient dans ce genre de méthode et qu'elle correspond assez peu à notre cas d'application. En effet, dans la mesure où nous travaillons sur des contenus fortement compressés, la qualité en restera plutôt faible aux yeux de l'évaluateur et ce, dans tous les cas. Cela est d'autant plus vrai qu'il est fortement déconseillé de convier des personnes étant expertes dans le domaine de l'imagerie pour participer à ce genre d'expérimentation.

Pour pallier à cet inconvénient, nous avons comparé uniquement l'image compressée en BPG par rapport à sa reconstruction (à des débits équivalents), et non la qualité générale de nos résultats (donc pas de comparaison avec les images initiales).

Pour cela nous avons adopté la méthode intitulée « *Stimulus comparison with adjectival categorical judgment* ». Cette méthode consiste à présenter deux images à un évaluateur et de lui demander laquelle il préfère entre les deux, et à quel point. Les critères d'appréciation ne sont ici pas renseignés. Pour cela, l'échelle illustrée Figure 6.6 est présentée à l'évaluateur.

–3	Much worse
–2	Worse
–1	Slightly worse
0	The same
+1	Slightly better
+2	Better
+3	Much better

Figure 6.6 : Echelle de notation utilisée pour la méthode « *Stimulus comparison with adjectival categorical judgment* »

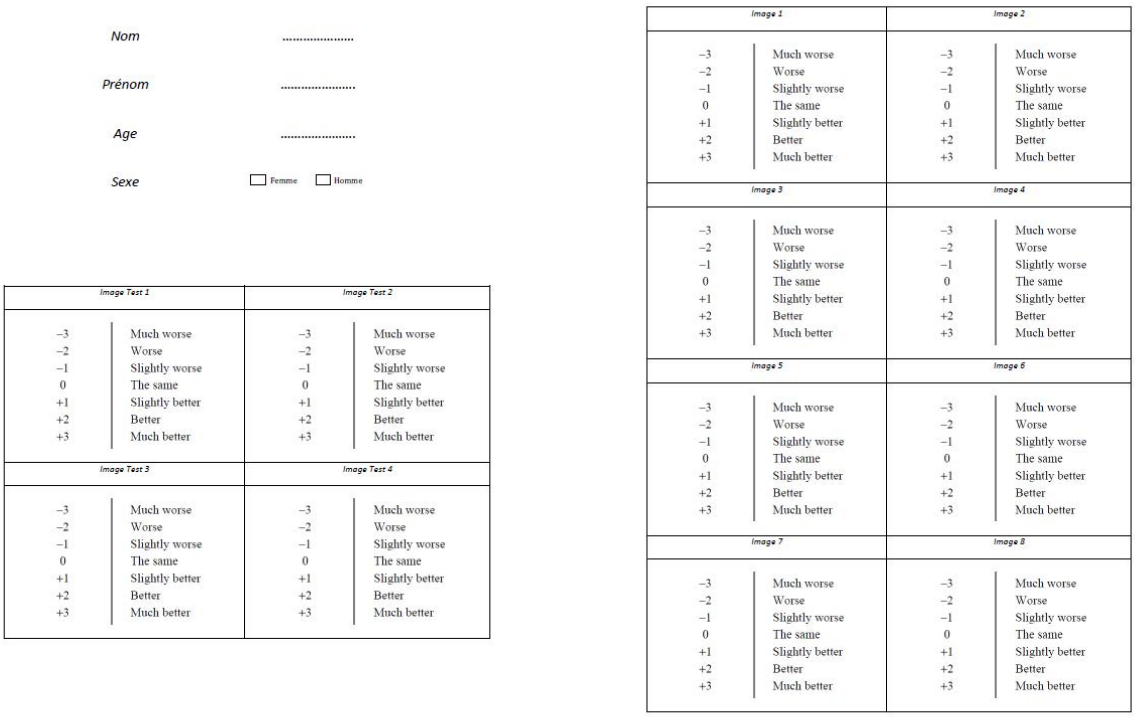
Concernant les images à évaluer, elles sont présentées à l'évaluateur côte à côte, comme illustré Figure 6.7.

Figure 6.7 : Exemple d'images présentées pour notation aux évaluateurs

Pour chaque image, l'évaluateur est libre de manipuler/zoomer l'image à sa guise et durant le temps qu'il souhaite avant d'attribuer une note.

Pour l'ensemble des évaluations, il a été demandé de noter l'image de droite par rapport à l'image de gauche. Nous noterons que la position de l'image compressée en BPG et de l'image reconstruite est déterminée aléatoirement pour chaque image.

Ainsi dans le cas de l'image ci-dessus, si l'on considère l'image de droite comme « *Slightly Worse* » que l'image de gauche, l'appréciation « *Worse* » sera donnée, associée à son score de -2. Le formulaire permettant l'évaluation est présenté Figure 6.8.



Nom .....

Prénom .....

Age .....

Sexe ☐ Femme ☐ Homme

Image Test 1		Image Test 2	
−3	Much worse	−3	Much worse
−2	Worse	−2	Worse
−1	Slightly worse	−1	Slightly worse
0	The same	0	The same
+1	Slightly better	+1	Slightly better
+2	Better	+2	Better
+3	Much better	+3	Much better
Image Test 3		Image Test 4	
−3	Much worse	−3	Much worse
−2	Worse	−2	Worse
−1	Slightly worse	−1	Slightly worse
0	The same	0	The same
+1	Slightly better	+1	Slightly better
+2	Better	+2	Better
+3	Much better	+3	Much better

Image 1		Image 2	
−3	Much worse	−3	Much worse
−2	Worse	−2	Worse
−1	Slightly worse	−1	Slightly worse
0	The same	0	The same
+1	Slightly better	+1	Slightly better
+2	Better	+2	Better
+3	Much better	+3	Much better
Image 3		Image 4	
−3	Much worse	−3	Much worse
−2	Worse	−2	Worse
−1	Slightly worse	−1	Slightly worse
0	The same	0	The same
+1	Slightly better	+1	Slightly better
+2	Better	+2	Better
+3	Much better	+3	Much better
Image 5		Image 6	
−3	Much worse	−3	Much worse
−2	Worse	−2	Worse
−1	Slightly worse	−1	Slightly worse
0	The same	0	The same
+1	Slightly better	+1	Slightly better
+2	Better	+2	Better
+3	Much better	+3	Much better
Image 7		Image 8	
−3	Much worse	−3	Much worse
−2	Worse	−2	Worse
−1	Slightly worse	−1	Slightly worse
0	The same	0	The same
+1	Slightly better	+1	Slightly better
+2	Better	+2	Better
+3	Much better	+3	Much better

Figure 6.8 : Deux première pages du formulaire fournis aux évaluateurs pour le protocole d'évaluation subjective, les autres pages sont similaires à la deuxième (à droite) afin d'aller jusqu'à l'image 24

Pour globaliser les scores d'évaluation obtenus, nous avons adopté la métrique MOS (*Mean Opinion Score*), définie pour chaque image évaluée comme la moyenne des notes sur l'ensemble des évaluateurs. Nous avons également calculé l'écart-type des scores MOS afin d'avoir une analyse plus précise.

Les images dites de test sont un ensemble de 4 images (Figure 6.9) étrangères au corpus d'évaluation, visant à « apprendre » à l'évaluateur à réaliser l'évaluation. Ces images ont ainsi un rôle de calibration afin que l'évaluateur puisse s'approprier sa propre échelle d'appréciation de la qualité visuelle. Bien entendu, les notes de ces 4 images sont exclues des résultats finaux.

Figure 6.9 : Ensemble de 4 images test utilisées pour calibrer l'évaluation

Enfin, notons que, dans un souci d'intégrité, aucune interaction entre les évaluateurs et nous n'a eu lieu pendant l'évaluation des 24 images du corpus d'évaluation Kodak.

Présentons à présent les résultats de l'évaluation subjective proposée. Comme l'évaluation subjective est très chronophage, en raison de l'investissement massif en ressources humaines, nous avons choisi pour notre évaluation seule les approches les plus pertinentes (*cf.* Chapitre 5). Ainsi, les méthodes retenues pour cette évaluation subjective sont les suivantes : Reconstruction ×1_43 par SRGAN, Reconstruction ×2 par ESRGAN et Reconstruction ×1_gen par ESRGAN avec perte L_1 pure.

Institut Polytechnique de Paris
91120 Palaiseau, France

6.1 Évaluation de la Reconstruction ×1_43 par SRGAN

Notre première évaluation expérimentale vise à évaluer les résultats issus de la Reconstruction ×1_43 par SRGAN. Ici, il a été demandé aux évaluateurs de comparer les résultats de la reconstruction à sa contrepartie compressée en BPG. Bien entendu, l'évaluateur ne savait pas quelle image correspondait à quel traitement. Cette évaluation a été réalisée dans le cadre de la publication de notre papier proposant cette technique dans la conférence IEEE ICCE Berlin 2019 [Hamis19].

Ici, 15 personnes âgées de 19 à 48 ans ont été conviées à évaluer les 24 images de validation, amenant le nombre d'images évaluées à 360 (15 évaluateurs par 24 images). Les scores MOS obtenus, ainsi que les variances associées sont présentés Figure 6.10.

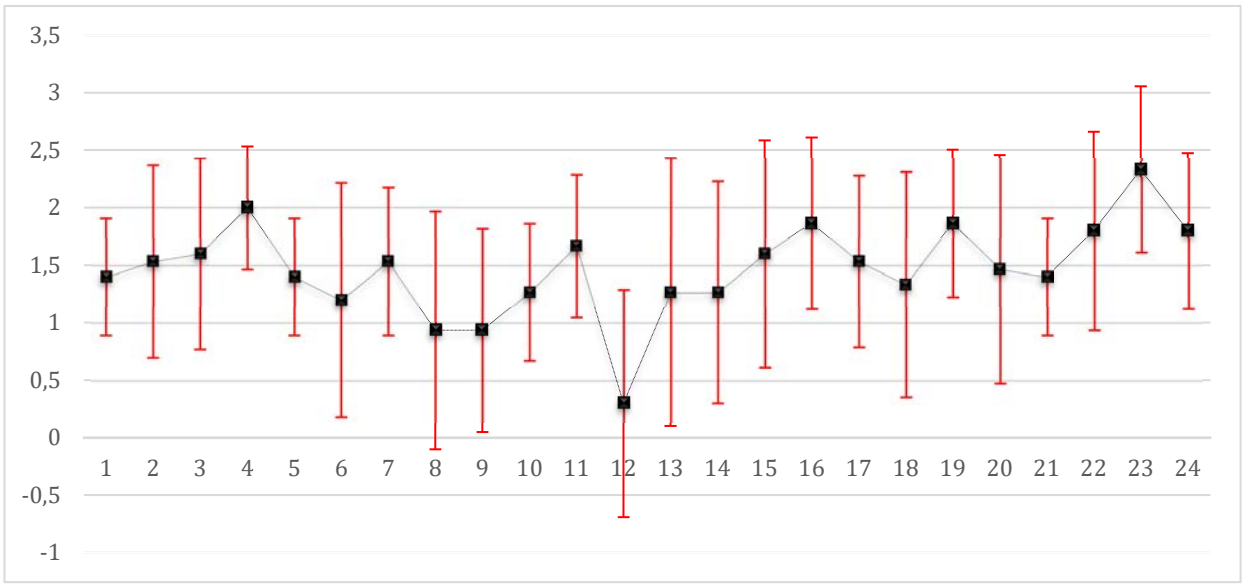


Figure 6.10 : *Mean Opinion Score* et Ecart Type obtenus par 15 évaluateurs sur les 24 images du corpus d'évaluation Kodak lors de l'évaluation subjective de la Reconstruction x1_43 par SRGAN

La moyenne globale des scores MOS est ici de 1,47, pour un écart type moyen est de 0,79. Pour l'ensemble des images de test, les scores MOS sont positifs, ce qui signifie que les reconstructions SRGAN ont été plébiscités.

Ainsi, la conclusion que nous pouvons tirer de ces résultats est que, de manière indiscutable, notre reconstruction est meilleure que la compression BPG par rapport à la perception humaine. Ce résultat s'oppose à ceux de l'évaluation objective, corroborant donc ce qui est rapporté dans la littérature concernant l'évaluation de modèles utilisant une perte perceptuelle.

Nous pouvons également observer une certaine homogénéité dans les résultats, à l'exception de deux cas « extrêmes », correspondant aux images avec respectivement le plus bas et le plus haut score.

L'image qui conduit au plus bas score (0,3) est l'image 12, représentant deux personnes marchant sur une plage (Figure 6.11).

Image BPG, q=43

Reconstruction x1_43 par SRGAN

Figure 6.11 : Image ayant reçu le plus bas score (0,3) lors de l'évaluation subjective dans ses deux versions

Institut Polytechnique de Paris
91120 Palaiseau, France

Pour cette image, le score de 0 a été assigné 4 fois, le score de 1 a été assigné 8 fois et des scores négatifs de -2 et -1 ont été assignés chacun 2 fois. C'est particulièrement le nombre de 1 et de 0 qui est notable et qui explique le faible score moyen de cette image.

Ce résultat s'explique par le fait que la partie saillante de l'image se trouve être le couple de personnes marchant sur la plage, et non l'arrière-plan. Or, les régions d'image relativement petites qui correspondent à ces deux personnages se trouvent être, de base, fortement altérées par la compression BPG, au point d'avoir des visages totalement inexploitables. Cela ne peut pas être corrigé par la Reconstruction SRGAN x1_43, qui ne dispose pas en entrée d'une quantité suffisante d'information, contrairement à l'ensemble de l'arrière-plan, qui est rendu correctement. De cette manière, les évaluateurs ont associé une note très faible à cette image en se concentrant, logiquement, sur la partie saillante.

Institut Polytechnique de Paris
91120 Palaiseau, France

A contrario, analysons à présent l'image 23, qui a obtenu le meilleur score (2,33), illustrée Figure 6.12

Image BPG, q=43

Reconstruction ×1_43 par SRGAN

Figure 6.12 : Image ayant reçu le plus haut score lors de l'évaluation subjective de la Reconstruction ×1_43 par SRGAN

Institut Polytechnique de Paris
91120 Palaiseau, France

Nous pouvons observer que pour cette image notre modèle a su prouver toute sa pertinence. Ici, la plupart des détails du premier plan, et donc de la partie saillante, ont pu être reconstruits et il en va de même pour l'arrière-plan, ce qui conduit à un résultat global particulièrement satisfaisant.

Ce résultat peut notamment être expliqué par la qualité initiale de la photo qui propose une très bonne séparation entre le premier et l'arrière-plan, ce dernier étant bien flouté. Le fait que des couleurs vives soit présentes, ainsi que le détail des plumes sont également des facteurs facilitant la reconstruction. Cette image nous permet également de souligner la puissance de notre méthode de reconstruction pour la réduction des artéfacts de compression BPG autour des arêtes, ainsi que pour la suppression les artéfacts de *blocking*.

Concernant l'évaluation globale, nous pouvons ajouter qu'en plus des 15 personnes retenues pour le protocole, nous avons également demandé à deux personnes atteintes de daltonisme de réaliser le test. Les résultats obtenus de ces deux personnes confirment ceux des 15 autres en suivant la même tendance, les scores obtenus étant de 1,21 et 1,42. Cela montre que le sentiment d'amélioration de notre modèle par rapport au BPG ne semble pas varier en fonction de la simple perception des couleurs de l'utilisateur, ce qui démontre une fois de plus la robustesse de notre modèle.

Finalement, pour conclure sur l'évaluation globale de la Reconstruction ×1_43 par SRGAN en employant un protocole d'évaluation subjective, notons que parmi les 360 échantillons évalués, la compression BPG seule a été préférée dans seulement 12 cas (soit dans 3,4% des cas) et considérée comme équivalente à notre reconstruction seulement 15 fois (soit dans 4,2% des cas). Cela nous indique que dans 96,6% des cas, notre reconstruction a été considérée comme meilleure ou équivalent à la compression BPG seule, prouvant ainsi la pertinence de notre modèle.

Institut Polytechnique de Paris
91120 Palaiseau, France

6.2 Évaluation de la Reconstruction ×2 par ESRGAN

Une deuxième évaluation subjective a été réalisée afin de valider notre Reconstruction ×2 par ESRGAN. L'objectif était ici d'étudier l'impact d'artéfacts sensiblement différents de ceux obtenus par Reconstruction ×1_43 par SRGAN. Cette évaluation a été réalisée dans le cadre de l'analyse de nos résultats sur la méthode alliant super-résolution et réduction d'artéfacts via un (E)SRGAN à des fins de compression à très bas-débit, publiée dans le IEEE Consumer Electronics Magazine en 2020 [Hamis20].

Ici, 20 personnes âgées de 22 à 55 ans, différentes des 15 personnes sollicitées lors de l'évaluation précédente, ont été conviées à évaluer les 24 images de validation, amenant le nombre d'images évaluées à 480. Les scores MOS obtenus, ainsi que leurs variances respectives sont présentés Figure 6.13.

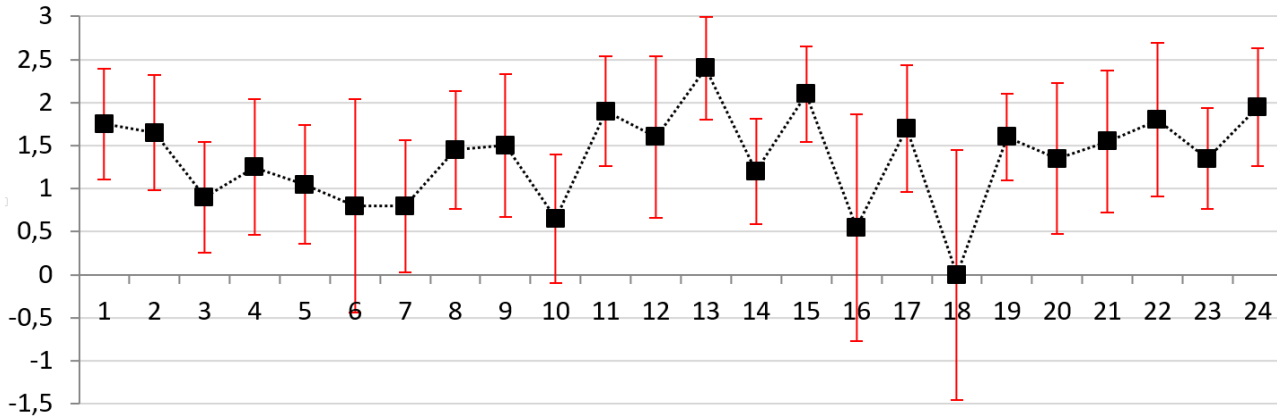


Figure 6.13 : Mean Opinion Score et Ecart Type obtenus sur 20 évaluateurs sur les 24 images du corpus d'évaluation Kodak lors de l'évaluation subjective de la Reconstruction x2 par ESRGAN

La moyenne des scores MOS sur l'ensemble des images de test est ici de 1,37, pour un écart type moyen est de 0,79.

Ici encore, cette évaluation démontre que notre modèle permet une très nette amélioration de la compression BPG. Nous nous retrouvons ainsi avec des scores similaires à ceux de l'évaluation précédente même si le score global moyen est très légèrement plus faible. Néanmoins, malgré la similitude des scores au regard de la moyenne globale, nous pouvons constater que la répartition des scores par images est bien plus hétérogène que lors de l'évaluation du modèle précédent.

Cela rejoint nos observations précédentes quant à l'apparition de nouveaux artéfacts liés à la reconstruction en utilisant une fonction perceptuelle relativement forte. En effet, ce type d'artéfacts résultant de l'utilisation d'une fonction de pertes perceptuelle à la faculté d'améliorer globalement la qualité de l'image, notamment au niveau des textures. Toutefois, dans certains cas, ces types d'artéfacts apparaissent comme trop marqués et, surtout, à un mauvais endroit.

Ce phénomène apparaît notamment pour le cas de l'image numéro 18 de notre corpus de test, qui a ici obtenu un score nul (Figure 6.14).

Figure 6.14 : Image ayant reçu un score nul lors de notre évaluation subjective de la Reconstruction ×2 par ESRGAN

Cette image, bien qu'ayant été préférée dans 10 cas sur 20 et considérée comme égale dans 2 n'a pas obtenu de score positif car, dans les faits, toutes les évaluations positives l'ont été avec un score de 1 (correspondant à « légèrement meilleure »), alors que les évaluations négatives ont plusieurs fois atteint -3 et -2. En effet, si l'on considère l'image dans sa globalité, elle apparait effectivement meilleure pour la version reconstruite. Les textures sont beaucoup plus détaillées, les artéfacts de flous ont été correctement remplacés par de la texture et les artéfacts de *blocking* ont disparu.

Néanmoins un problème majeur apparait dans notre reconstruction, au niveau notamment du visage de la personne en premier plan, qui représente la partie saillante de l'image. Pour mieux illustrer ce phénomène, la Figure 6.15 présente un zoom sur la partie du visage.

Institut Polytechnique de Paris
91120 Palaiseau, France

Image BPG, q=40

Reconstruction ×2 par ESRGAN

Figure 6.15 : Zoom sur la partie « controversée » (le visage) de l'image la moins bien notée

Nous pouvons observer que le visage a été très mal reconstruit par notre reconstruction, au point de devenir particulièrement désagréable à regarder. Cela est dû principalement au caractère relativement peu texturé de cet élément. La compression BPG a naturellement tendance à lisser et est donc capable de fournir un résultat correct. Au contraire, la reconstruction ESRGAN cherche à déterminer des éléments de texture et ajoute donc artificiellement des détails qui dans ce cas sont inutiles et même nuisibles.

Concernant l'image ayant reçu le meilleur score (2,4), illustrée Figure 6.16, nous nous retrouvons dans un cas totalement différent où la quasi-totalité des éléments, même saillants sont hautement texturés.

Institut Polytechnique de Paris
91120 Palaiseau, France

Image BPG, q=41

Reconstruction ×2 par ESRGAN

Figure 6.16 : Image ayant reçu le meilleur score lors de notre évaluation subjective de la Reconstruction ×2 par ESRGAN

Institut Polytechnique de Paris
91120 Palaiseau, France

Ce bon résultat s'explique par le fait que, dans la mesure où la plupart du contenu est constitué de textures marquées, ces dernières ont été naturellement fortement floutées par la compression BPG. Notre modèle étant particulièrement adapté à la reconstruction de textures, il en résulte forcément un résultat bien plus intéressant et agréable visuellement qu'avec la compression BPG seule.

Pour cette évaluation de la Reconstruction ×2 par ESRGAN, parmi les 480 échantillons évalués, l'image BPG seule a été préférée dans seulement 19 cas, et une note équivalente a été attribuée 38 fois (soit dans 7,9% des cas). Cela signifie que dans 96% des cas notre méthode a été plus appréciée ou considérée équivalente au BPG, prouvant encore une fois la pertinence de notre approche.

Ces résultats, bien que très encourageants apparaissent comme moins bons que ceux obtenus pour la Reconstruction ×1_43 par SRGAN. Cela est expliqué notamment par le fait que la Reconstruction ×2 étudiée lors de nos recherches ne prend pas en compte qu'un seul facteur de compression mais une plage d'une dizaine de facteurs. Il s'agit d'une tâche bien plus complexe que la seule RAC d'un seul facteur de compression, expliquant ainsi les résultats légèrement moins bons malgré l'utilisation d'un réseau de neurones plus développé. Il pourrait en conséquent être intéressant dans de futures recherches d'étudier la Reconstruction ×2 pour un unique facteur de compression à l'instar de la Reconstruction ×1_43.

6.3 Évaluation de la Reconstruction ×1_gen par ESRGAN avec perte L_1 pure

Une dernière évaluation subjective a été menée afin d'évaluer les résultats de la Reconstruction ×1_gen avec une fonction de pertes L_1 pure. Lors de la présentation des résultats expérimentaux (Section 5.5.5.1), nous avons établi que ce type de reconstruction permettait une très bonne reconstruction des structures mais n'avait que très peu d'impact sur les textures.

Pour cette évaluation, 15 personnes âgées de 24 à 49 ans, ont été conviées à évaluer les 24 images de validation, amenant le nombre d'images évaluées à 360. Les scores MOS obtenu, ainsi que leurs variances respectives, sont présentés Figure 6.17.

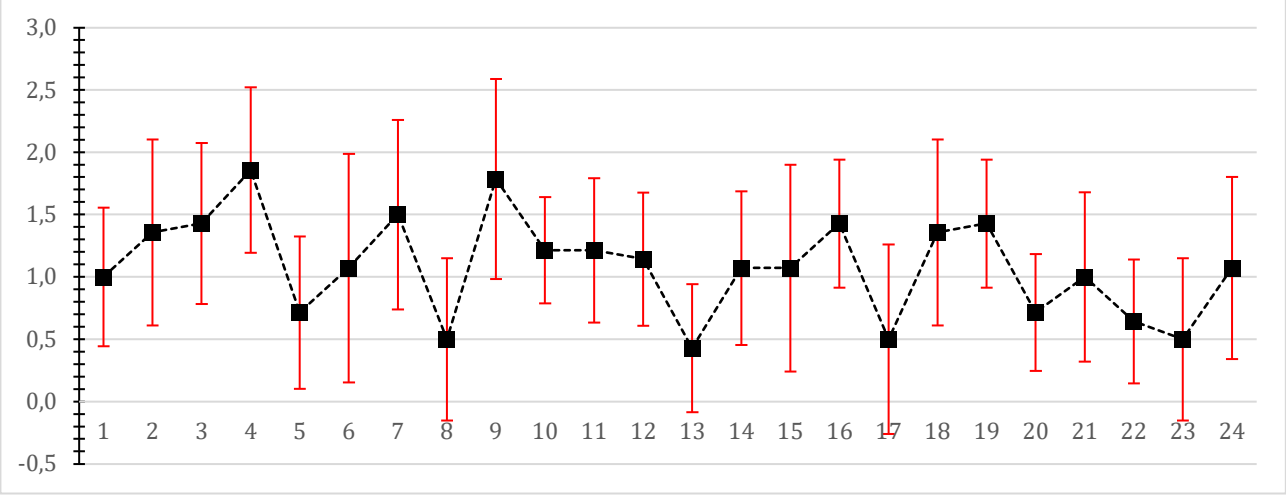


Figure 6.17 : *Mean Opinion Score* et Ecart Type obtenus sur 15 évaluateurs sur les 24 images du corpus d'évaluation Kodak lors de l'évaluation subjective de la Reconstruction ×1 par ESRGAN avec perte L_1 pure.

La moyenne globale des scores est de 1,08, pour un écart type moyen de 0,64.

Institut Polytechnique de Paris
91120 Palaiseau, France

Nous nous retrouvons donc ici avec à la fois une moyenne et un écart type moyen sensiblement plus faibles que lors de l'évaluation des modèles ayant été entrainés avec une fonction de pertes perceptuelle.

Cela s'explique par le fait que notre modèle n'a permis dans ce cas que la réduction d'artéfacts spécifiques de BPG, sans réaliser de réelle « reconstruction » des contenus dégradés lors de la compression. En conséquence, nous nous retrouvons avec des résultats supérieurs à BPG, de manière bien plus systématique qu'avec une perte perceptuelle, mais avec des scores moins probants.

Toutefois, sur les 360 résultats évalués, l'image BPG n'a été préférée que dans seulement 3 cas, ce qui signifie que notre modèle L_1 a été préféré ou considéré comme égal au BPG seul dans 99,16% des cas, ce qui est bien plus élevé que pour nos deux autres modèles.

En revanche, si l'on regarde le nombre de fois où les deux images ont été considérées comme équivalentes, nous nous retrouvons avec 67 cas (soit 18,6%), ce qui est significativement plus élevé que lors de l'évaluation des modèles perceptuels.

Si l'on regarde l'image ayant eu le score le plus faible, nous retombons sans surprise sur l'image ayant eu le meilleur score lors de l'évaluation de la Reconstruction ×2 par ESRGAN (Figure 6.18).

Institut Polytechnique de Paris
91120 Palaiseau, France

Image BPG, q=43

Image reconstruite

Figure 6.18 : Image ayant reçu le moins bon score lors de notre évaluation subjective de la Reconstruction ×1_gen par ESRGAN avec perte L_1 pure

Institut Polytechnique de Paris
91120 Palaiseau, France

Les raisons de ce mauvais score sont les mêmes que celles expliquant le bon score de cette image lors de l'évaluation de la Reconstruction ×2 par ESRGAN. En effet, cette image étant principalement composée de textures, que la reconstruction à base de perte L_1 n'est pas en mesure de reconstruire, il est tout à fait normal que les différences entre le modèle BPG et sa reconstruction ne soient que peu notables.

A l'inverse, l'image ayant reçu le meilleur score est une image ne présente que très peu de textures, y compris sur la partie saillante (Figure 6.19).

Image reconstruite Image BPG

Figure 6.19 : Image ayant reçu le meilleur score lors de notre évaluation subjective de la Reconstruction ×1_gen par ESRGAN avec perte L_1 pure

Nous pouvons clairement voir dans cette image la suppression des artéfacts de *blocking*, de *ringing* ainsi que la suppression des lignes de prédiction. Néanmoins, nous pouvons observer que la partie texturée (au niveau des cheveux), n'a aucunement été reconstruite. Dans les faits, cela n'a aucunement gêné les évaluateurs, dans la mesure où le visage est ici l'élément saillant de l'image.

Cela conclut la série d'évaluations subjectives de nos différents modèles de reconstruction. Nous ne proposons que trois évaluations différentes car, comme évoqué précédemment, ce type de protocole se révèle être assez chronophage et nécessite de mobiliser des panels conséquents d'évaluateurs.

Institut Polytechnique de Paris
91120 Palaiseau, France

Notre choix s'est porté sur ces trois modèles en particuliers car ils constituent un échantillon représentatif en termes de résultats et surtout en termes d'artéfacts de reconstruction. La Reconstruction ×1_43 par SRGAN nous montre les résultats pour un facteur de compression fixe et fait était des capacités obtenues par un SRGAN. La Reconstruction ×2 par ESRGAN représente les meilleurs résultats que nous pouvons obtenir sur des images génériques et permet de bien insister sur les artéfacts de reconstruction de textures. Enfin, la Reconstruction ×1 par ESRGAN avec perte L_1 pure permet de mettre en lumière la puissance de cet outil pour reconstruire les zones lisses d'une image mais également de réaliser sa faiblesse quant à la reconstruction de textures.

Dans les trois cas que nous avons étudiés, nous pouvons souligner le très haut taux de préférence de nos modèles par rapport à une compression BPG exclusive. Nous pouvons également insister sur le fait que ces tendances restent les mêmes, quels que soient les résultats traduits par les métriques de qualité objectives que sont les SSIM et PNSR.

En conclusion, nous avons, par le biais de ces évaluations, démontré la pertinence de nos modèles et de notre méthode, tout en ayant mis en lumière les limitations associées. Les performances obtenues renforcent l'hypothèse que nous avons énoncée précédemment : il apparait comme nécessaire, afin d'obtenir un modèle bien plus efficace, de pré-segmenter notre image en fonction de zones de textures et des structures associées, pour les appliquer de manière adaptative sur les différentes régions d'image.

Il s'agirait là de notre principal axe à explorer pour des travaux futurs visant à l'amélioration de notre méthode.

Dans l'optique d'expérimenter cet axe d'amélioration, nous avons décidé de nous pencher dans le prochain chapitre sur un aspect particulier de nos cas d'usage de transmission d'images à bas très débit. Il s'agit notamment des les images contenant des contenus critiques comme du texte.

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 7. Application à la problématique des images avec texte

Résumé. Durant la quasi-totalité de ce manuscrit nous avons consacré nos recherches à la compression d'images génériques en considérant les images seulement dans leur globalité. Nous avons plusieurs fois évoqué la problématique de conservation de détails critiques comme du texte mais nous n'avons pour le moment pas dédié de travaux spécifiquement à cette tâche.

L'objectif de ce chapitre est donc de se concentrer plus finement sur la transmission de contenus images comme des photos de formulaires textuels ou de cartes d'identité, notamment afin de répondre aux problématiques qu'a pu nous soumettre notre entreprise partenaire, Be-Bound.

La contribution principale détaillée dans ce chapitre concerne notamment la mise en place d'une approche de compression d'image adaptative. Le principe consiste à segmenter l'image en deux parties, une première correspondant aux zones de texte et la seconde représentant l'arrière-plan. Ces deux parties sont ensuite compressées séparément, avec des techniques dédiées. L'objectif ici est de préserver au maximum l'information critique qui, dans ce cas, correspond aux éléments textuels. Les images reconstruites sont ensuite recomposées au niveau du décodeur. Pour éliminer les artéfacts de composition, une approche de reconstruction ESRGAN est proposée.

Les résultats obtenus sont présentés et discuté sur un corpus de test varié, incluant des images à la fois naturelles, de cartes d'identité ou encore présentant massivement du texte.

Mots clés **:** compression adaptative, segmentation, détection de texte, recomposition d'image

Institut Polytechnique de Paris
91120 Palaiseau, France

Bien que nous ayons évoqué à plusieurs reprises la problématique de fidélité de reconstruction par rapport à des contenus critiques, comme les séries de chiffres ou de lettres qui apparaissent sur une photo de carte d'identité, nous n'avons pour le moment pas étudié en profondeur la question. Jusqu'à présent, l'approche qui paraît la plus prometteuse est celle de la Reconstruction ×1, introduite spécifiquement pour conserver au maximum ce type de détails.

Un deuxième axe qui nous paraît pertinent concerne les reconstructions qui s'appuient sur des modèles entrainés avec fonction de pertes totalement orientées pixels (L_1 ou L_2), qui s'affranchissent de toute composante perceptuelle. Nos premiers résultats présentés dans la Section 5.5.5.1 ont montré que ce type de modèle permettait une très bonne reconstruction des structures d'une image. Or, un texte dans une image s'apparente bien plus à de la structure qu'à de la texture. De plus, dans le cadre de cette application, il ne s'agit pas de privilégier les aspects esthétiques globaux de la reconstruction, mais de préserver au maximum l'information utile contenue dans l'image.

Observons également que les différentes approches proposées présentent des avantages et limitations et qu'aucune d'entre elles ne semble fournir une solution globale au problème considéré.

Dans ce chapitre, nous allons explorer la piste d'amélioration évoquée dans le chapitre précédent, qui concerne une approche adaptative, par rapport aux différentes régions d'image. Le principe consiste à segmenter l'image en zones d'intérêts, puis de compresser/reconstruire séparément ces dernières en fonction des besoins avec des méthodes adéquates à chaque type de contenu.

7.1 Solution de compression adaptative proposée

Plus précisément, nous nous sommes concentrés sur un cas d'usage relativement simple, où seulement deux zones d'intérêt sont considérées, correspondant aux régions incluant du texte et à l'arrière-plan. Ces deux zones distinctes seront alors séparées, puis traitées différemment.

Pour la partie textuelle, le critère fondamental à prendre en compte est sa lisibilité au niveau des images décodées. Jusqu'à présent, les approche de compression purement BPG ainsi que les reconstructions ×1 avec fonctions de pertes orientées pixels sont les candidats les plus prometteurs. Les paramètres contrôlant le niveau de compression (comme le facteur q de la compression BPG) doivent ici être réglés finement. Pour le fond, ne contenant pas d'information utile, nous pouvons appliquer des compressions plus fortes, afin de réduire au maximum le débit. En particulier, nous pouvons considérer ici des techniques de super-résolution qui permettent de réduire drastiquement la quantité de l'information à coder. Ce processus est illustré Figure 7.1. L'image du fond est ici sous-échantillonnée d'un facteur 2, en vue d'une Reconstruction ×2.

Institut Polytechnique de Paris
91120 Palaiseau, France

Figure 7.1 : Illustration de la décomposition d'une image (de 480×480 pixels) en parties texte et arrière-plan, qui vont être traitées et compressées différemment : la partie textuelle est compressée en BPG (q=47, débit 2082 octets octets) en résolution initiale. Le fond est sous-échantillonné d'un facteur 2 et compressé également en BPG, à un taux plus important (q=37, débit 3273 octets). Nous avons ainsi une somme de 5355 octets pour l'image totale soit 0,19 bpp.

La séparation de la partie texte et de l'arrière-plan a été réalisée dans cet exemple par segmentation automatique. Pour cela, nous avons adopté le réseau EAST (*Efficient and Accurate Scene Text Detector*), introduit dans [Zhou17]. EAST est un détecteur de texte réalisé via un réseau de neurones purement convolutionnel, dont l'architecture est illustrée Figure 7.2 Nous pouvons remarquer que la segmentation EAST n'est pas parfaite, mais ce n'est pas un aspect essentiel pour nos travaux. Ce qui nous intéresse est uniquement d'étudier en quelle mesure une approche par segmentation peut conduire à des résultats de compression viables.

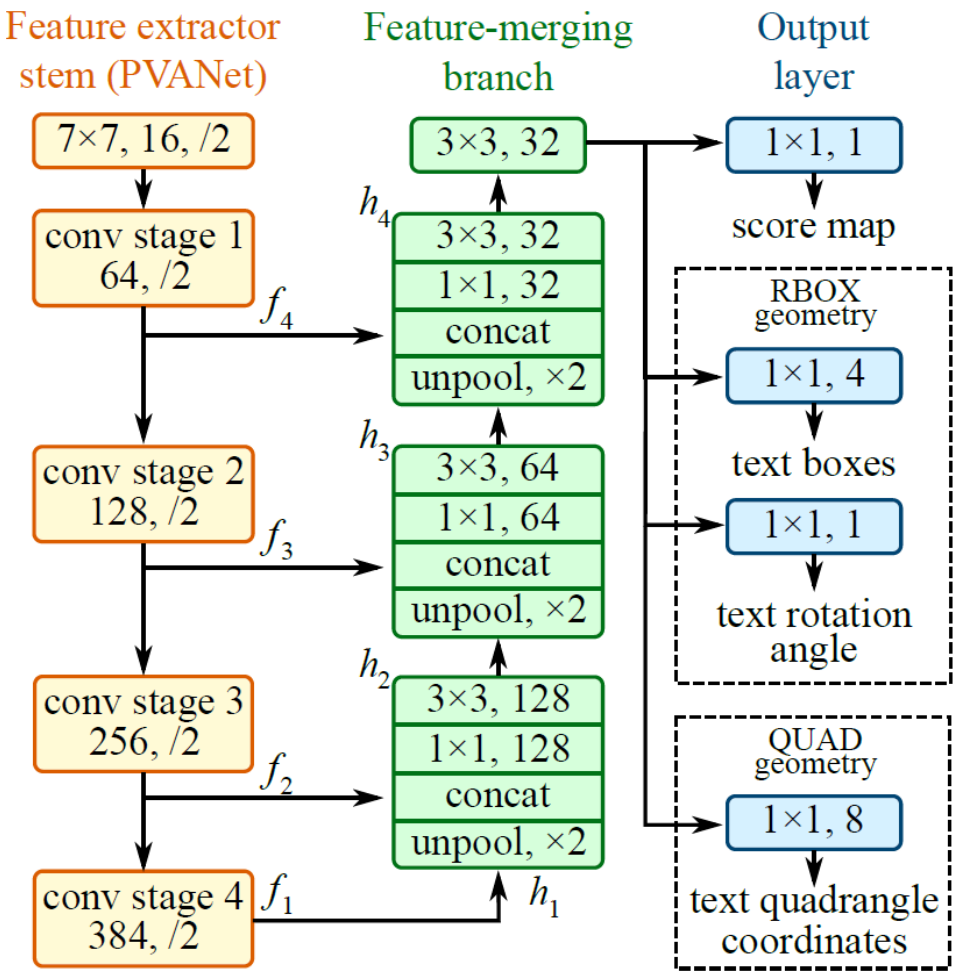


Figure 7.2 : Architecture du détecteur de texte EAST [Zhou17]

Pour nos expérimentations, nous avons utilisé la version de EAST disponible au GitHub suivant : https://github.com/argman/EAST.

Nous retrouvons donc avec deux images différentes compressées en BPG, à deux résolutions différentes et avec des facteurs de compression adaptés à chacun. Elles vont correspondre à deux flux/fichiers à transmettre séparément au décodeur, qui aura alors la tâche de réaliser le décodage/la reconstruction de chaque flux et surtout la recomposition de l'image globale.

Un premier objectif est d'assurer que la somme des débits obtenus pour les deux parties, texte et fond, soit bien inférieure à celle que nous aurions obtenue avec une seule compression globale, réalisée sur l'ensemble de l'image, et ce, à qualité équivalente.

Ce résultat devient possible en compressant bien plus fortement les contenus séparés dans la mesure où, d'une part, ils ne présentent pas le même degré de criticité d'information, et, d'autre part, la compression s'adapte mieux à des contenus homogènes.

Dans ce premier exemple, nous avons considéré une Reconstruction ×2 pour l'arrière-plan et une Reconstruction ×1 avec une fonction de pertes L_1 pour l'image contenant la partie de texte segmenté. Les résultats obtenus sont illustrés Figure 7.3.

Reconstruction ×1 avec perte L_1 du texte segmenté

Reconstruction ×2 de l'arrière-plan

Figure 7.3 : Illustration des reconstructions des deux images contenant le texte segementé et l'arrière-plan

Une fois reconstruites, ces deux images sont ensuite recomposées comme elles l'étaient dans l'image originale. Dans ce cadre, les zones de texte décodées sont tout simplement superposées sur l'image de l'arrière-plan. Pour cela, il est nécessaire de transmettre au décodeur une information auxiliaire, représentant les coordonnées des boîtes englobantes des zones textuelles. Le résultat obtenu pour notre exemple est illustré Figure 7.4.

Institut Polytechnique de Paris
91120 Palaiseau, France

Figure 7.4 : Image recomposée (zoom ×3.5) à partir des deux reconstructions (texte et arrière-plan)

Après recomposition des deux parties reconstruites, nous pouvons observer l'apparition d'artéfacts très marqués et désagréables autour des zones segmentées. Cela est dû au fait que les deux images aient été traitées différemment et séparément. Par conséquent, elles présentent de fortes discontinuités au niveau des « jonctions » des deux images, faisant clairement apparaître sur l'image reconstruite les boîtes englobantes des zones textuelles. Ce phénomène est d'autant plus gênant que, dans un cas d'usage pratique, comme celui qui concerne la transmission de cartes d'identité, cela donne l'impression d'une image faussée et donc inutilisable. Il est donc indispensable de proposer une solution adaptée, pouvant traiter et éliminer ce type d'artéfacts.

Pour corriger ces effets de bords, nous proposons encore une fois d'utiliser une approche de reconstruction par ESRGAN, dédiée spécifiquement et exclusivement à la suppression de ces nouveaux artéfacts. Le principe consiste à entraîner un réseau ESRGAN prenant en entrée les recompositions brutes obtenues précédemment et en sortie l'image à coder initiale.

Pour mettre en place cette stratégie, nous avons entrainé une Reconstruction ×1 par ESRGAN avec perte L_1 pure. Pour cet entrainement, nous avons créé une base de données contenant 843 images contenant une partie conséquente de texte (Figure 7.5). Ces images ont ensuite été subdivisés en 4285 patches de taille 480×480 pixels, constituant ainsi notre corpus d'entrainement.

Institut Polytechnique de Paris
91120 Palaiseau, France

Figure 7.5 : Exemples d'images de notre corpus d'entrainement.

L'entrainement a ici été réalisé sur 1500 *epochs*.

Le résultat obtenu pour l'exemple présenté précédemment est illustré Figure 7.6.

Figure 7.6 : Résultat de la reconstruction du contenu recomposé (zoom ×3,5)

Nous pouvons observer que les artéfacts de recomposition ont très sensiblement disparus, conduisant maintenant à un contenu unifié, agréable à regarder et exploitable.

De manière plus générique, la chaîne de traitement de la méthode proposée est illustrée Figure 7.7.

Institut Polytechnique de Paris
91120 Palaiseau, France

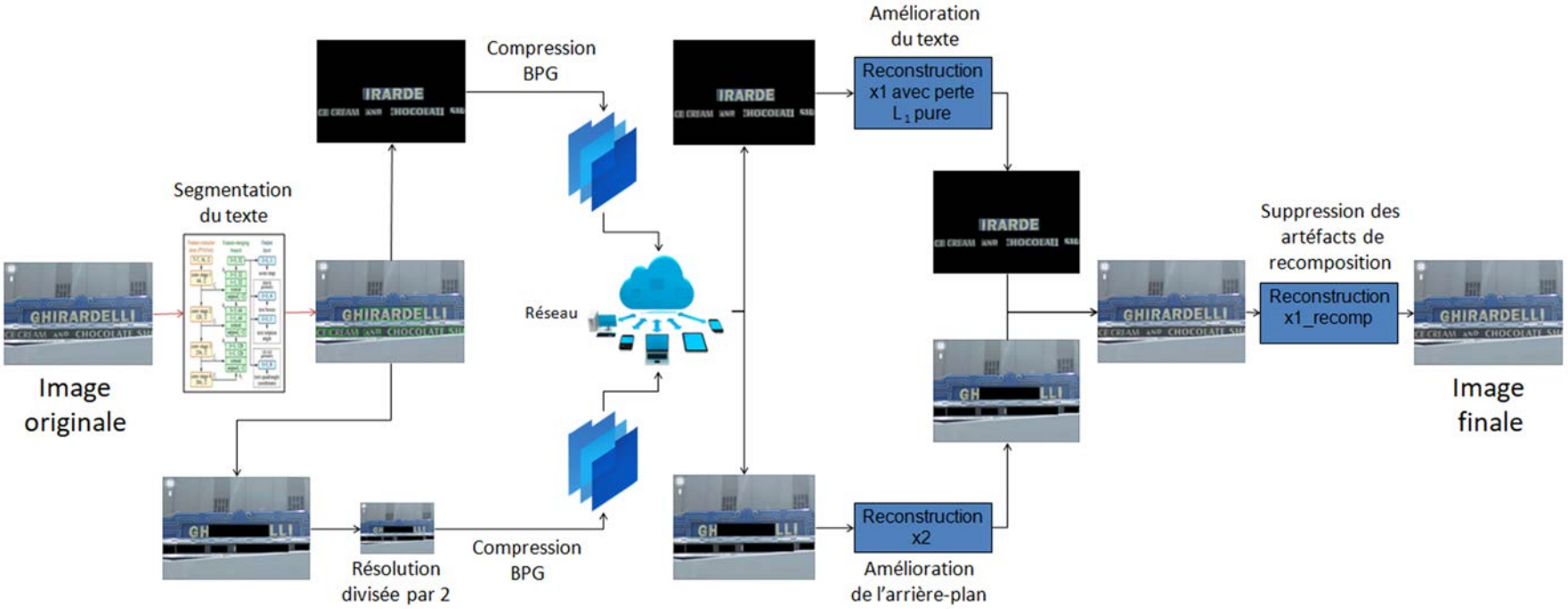


Figure 7.7 : Pipeline complet de notre solution utilisant la segmentation pour les images avec texte

Institut Polytechnique de Paris
91120 Palaiseau, France

7.2 Validation expérimentale

Pour valider le modèle proposé, nous avons sélectionné comme corpus de test, un panel de 16 images de différentes types, contenant du texte et incluant à la fois des contenus naturels (Figure 7.8) et des photos de cartes d'identité.

Pour les 6 images correspondant à des images naturelles les résultats sont très bons et les artéfacts de recomposition sont quasiment imperceptibles, comme illustré.

Figure 7.8 : Résultats de reconstruction adaptative sur des images naturelles

En revanche, notre méthode ne semble toujours pas permettre de faire face au cas où le texte est préalablement rendu illisible par la compression. De même, malgré les résultats encourageants, de trop forts taux de compression semblent également perturber la réduction d'artéfacts de recomposition, comme illustré sur la première et dernière figure présentée.

Nous avons également testé 5 images de type cartes d'identité (Figure 7.9).

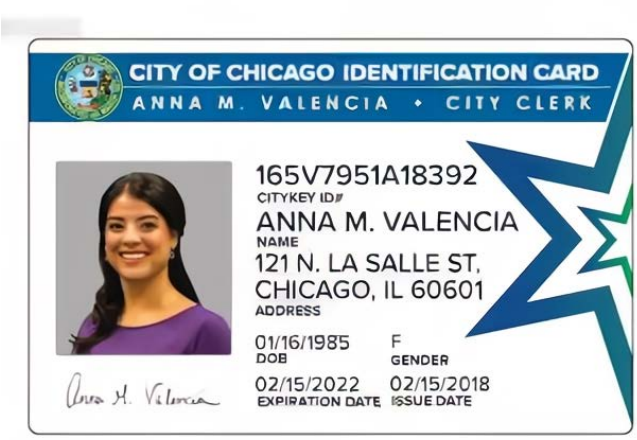


Figure 7.9 : Reconstruction d'images de cartes d'identité.

Institut Polytechnique de Paris
91120 Palaiseau, France

Il est relativement difficile de tirer ici des conclusions, notamment dans le cadre d'une éventuelle application industrielle. En effet, pour certaines cartes, les artéfacts de recompositions sont totalement effacés, alors que pour d'autres, bien que significativement réduits, ils demeurent néanmoins visibles.

Une piste possible d'amélioration serait d'élargir légèrement les boîtes englobantes et d'appliquer également une procédure de *padding* à l'intérieur des zones de texte éliminées sur l'image d'arrière-plan. Cela pourrait minimiser, d'entrée de jeu, les artéfacts de compression qui apparaissent naturellement dans ces zones, en limitant les discontinuités.

Nous pouvons néanmoins remarquer que dans tous les cas le texte critique est clairement conservé et les photos d'identités restent entièrement reconnaissables.

Enfin, présentons les résultats obtenus sur des photos contenant une quantité très importante de texte comme des articles de journaux, livres ou écran d'informations (*screen content* [Zhu14]) :

Institut Polytechnique de Paris
91120 Palaiseau, France

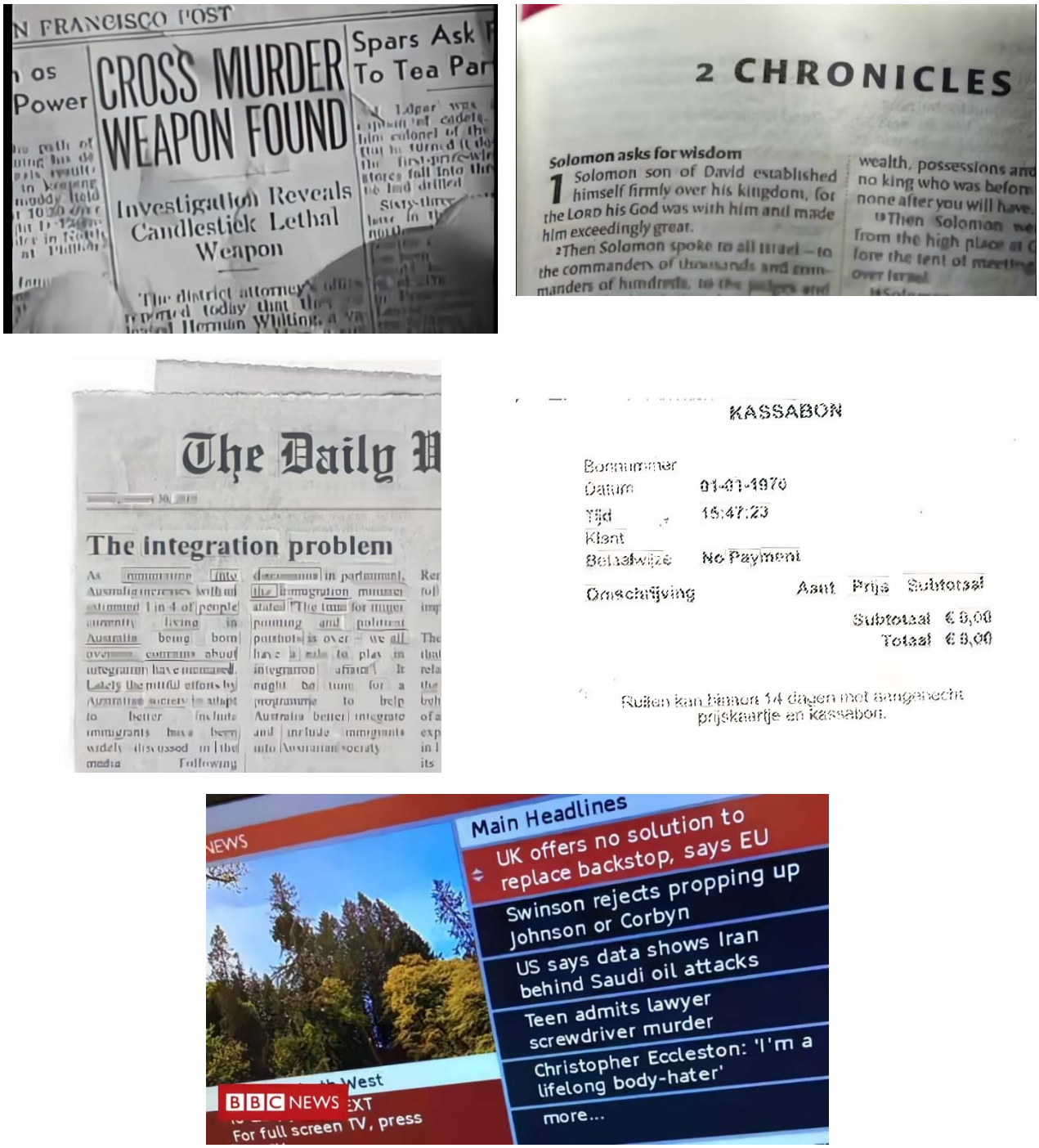


Figure 7.10 : Reconstruction d'images faisant intervenir une quantité importante de texte.

Ces résultats nous permettent de tirer globalement des conclusions similaires à celles mises en évidence pour les types d'images précédents. Globalement, l'approche conduit dans l'ensemble à des résultats satisfaisants, mais des améliorations supplémentaires sont toutefois nécessaires.

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 8. Conclusion et perspectives

Les recherches présentées dans ce manuscrit de thèse se sont focalisées sur la problématique de compression d'image, avec un objectif précis, guidé par une application industrielle : proposer une solution pour rendre exploitable la transmission et le rendu des images sur téléphone mobile, à très bas débit et si possible sur des réseaux 2G (par paquets de SMS).

Pour atteindre cet objectif, nous avons réalisé en premier lieu un état de l'art des codecs d'images actuels et de leurs potentielles évolutions (Chapitre 2). L'analyse de l'état de l'art a montré que les normes et formats de compression existants ne sont pas entièrement adaptés à nos objectifs. Pour cette raison, nous avons décidé d'orienter nos développements vers un nouvel axe de recherche qui concerne les méthodologies émergentes de *machine* et *deep learning,* aujourd'hui en plein essor. Un état de l'art de ces techniques est proposé au Chapitre 3.

Les études préliminaires que nous avons réalisées, nous ont conduit notamment à considérer une approche consistant à mettre en place des méthodes d'amélioration de la qualité de contenus déjà fortement compressés par des codecs génériques, puis de raffiner notre modèle pour l'adapter à des cas d'usage particuliers. Cela permet de coupler la facilité d'utilisation, la généricité et légèreté des codecs existants à un post-traitement spécifique d'amélioration de qualité et rend possible une utilisation large de nos solutions, puisqu'elles ne requièrent pas de modifier totalement la chaîne de compression/décompression des codecs existants. De plus, notre méthode reste applicable indépendamment du codec utilisé.

Suivant cette logique, nous avons donc considéré une série de réseaux de neurones initialement dédiés aux objectifs de super-résolution, que nous avons modifié et entraînés spécifiquement à des fins d'amélioration visuelle d'images très fortement compressées. Ils concernent des réseaux de type GAN (*Generative Adversial Networks*), qui ont montré ces dernières années des résultats et performances spectaculaires pour des objectifs génériques de transformation/reconstruction d'images. Les différentes techniques de l'état de l'art par réseaux GAN sont présentées au Chapitre 4.

Plus précisément, les approches retenues dans le cadre de nos développements concernent les réseaux SRGAN et ESRGAN, qui s'imposent aujourd'hui comme état de l'art dans le domaine de la super-résolution. Nos réseaux agissent alors comme des filtres de reconstruction, placé en bout de la chaîne de compression/décompression. Ils peuvent alors très aisément être ajoutés (ou désactivés) à un schéma de compression/décompression déjà existant.

Les modèles proposés, présentés en détails au Chapitre 5 se divisent en deux grandes catégories. Dans la première, les modèles gardent la résolution de l'image intacte et agissent uniquement pour atténuer les artéfacts issus de la compression initiale. Dans la deuxième, le modèle effectue conjointement super-résolution et atténuation des artéfacts de compression. Dans ce deuxième cas, nous proposons un schéma de compression alternatif où la réduction de résolution de l'image initiale est opérée avant la compression par codec, la résolution initiale étant alors restituée en phase de décompression.

Dans le cadre de l'évaluation de nos résultats (Chapitre 6), nous proposons une discussion sur les différentes techniques d'évaluation de contenus visuels compressés, notamment quand l'utilisation finale de ces contenus se résume à une visualisation simple. Plus précisément nous nous intéressons à l'évaluation des images traitées par des réseaux de neurones, et démontrons que le système

d'évaluation le plus pertinent, dans notre cas, se trouve être de type subjectif, ne répondant pas aux métriques objectives traditionnelles.

Ainsi, un protocole d'évaluation subjective est proposé et mis en œuvre pour évaluer les plus prometteurs modèles retenus. L'analyse comparée des résultats obtenus est présentée en détails au Chapitre 6. Ces évaluations démontrent que nos modèles sont suffisamment robustes et trouvent toute leur pertinence pour des cas d'utilisation variés.

Les modèles et la méthodologie proposés permettent d'améliorer des contenus très compressés génériques. Nous avons alors tenté d'affiner nos travaux, afin de répondre à une des problématiques majeures qui nous a été proposée par notre entreprise partenaire Be-Bound : le traitement d'images à très-bas débit présentant du contenu critique comme du texte sur des cartes d'identité par exemple. Ces développements sont présentés au Chapitre 7.

Nous avons premièrement pu déterminer que pour cette application, il était nécessaire d'utiliser comme base nos réseaux ne réalisant pas de super-résolution, puisque cette dernière rend d'entrée de jeu les contenus critiques inexploitables. Nous avons ensuite étudié un schéma de compression sélective, où les contenus critiques seraient compressés moins fortement que le reste de l'image. Nous montrons également ici comment les réseaux ESRGAN peuvent être utilisés pour des objectifs de composition sans rupture des régions d'images différemment compressées.

Bien que les premiers résultats obtenus soient très encourageants, des développements restent à faire pour les rendre exploitables dans le cadre d'une application industrielle.

Ce dernier point représente aussi notre première piste de développements futurs. En effet, nous pensons qu'en considérant une meilleure segmentation du texte sur une image pour développer une automatisation du schéma de compression sélective présenté précédemment, il serait tout à fait envisageable de pouvoir transmettre des contenus tels que des cartes d'identité sur des paquets d'une dizaine de SMS, sans risque d'erreur de contenu.

Une deuxième piste de développement concerne l'optimisation computationnelle de ces réseaux, notamment pour le traitement d'images de basse résolution. Dans ce cadre, les pistes de recherche à considérer concernent l'optimisation de l'architecture des réseaux, des paramètres d'entraînement ainsi qu'une meilleure définition de la fonction de pertes. L'utilisation de *datasets* d'entraînement plus adaptés et plus conséquents semble aussi être une piste de développement prometteuse.

Lors de nos recherches, nous avons considéré le format BPG, qui présente les artéfacts de compression les plus complexes parmi les différents codecs d'images existants. Mais *a priori*, la même méthodologie peut tout aussi bien s'appliquer, et même donner de meilleurs résultats, pour des codecs moins complexes. Il serait donc intéressant d'entraîner de nouveaux modèles sur d'autres codecs afin de valider complètement cette hypothèse. De manière plus générale, il serait intéressant d'investiguer la possibilité d'élaborer un unique modèle permettant de traiter tout type d'artefact de compression issu d'un codec générique.

Institut Polytechnique de Paris
91120 Palaiseau, France

Dans le cadre de notre collaboration industrielle avec Be-Bound, nous avons également été amenés à considérer des images possédant une grande richesse de contenus ayant une résolution élevée. L'objectif derrière ces expériences n'était plus ici de traiter des images à très bas débit mais de compressée au maximum une image « artistique » sans que les déformations soient vraiment notables.

Nous avons pour cela sélectionné notre Reconstruction ×2 par ESRGAN et ce, pour plusieurs raisons.

La première raison est tout simplement liée au fait qu'il s'agit de la reconstruction ayant donné les meilleurs résultats sur des images génériques, notamment par rapport aux textures. Ensuite, dans la mesure où la plupart des opérations réalisées par notre réseau de neurones est réalisée sur les patches en basse-résolution, nous bénéficions d'un temps de traitement bien plus réduit en pratiquant une Reconstruction ×2 plutôt que ×1.

Nous avons pu constater que, même si la méthodologie de base avait été développée dans l'optique de traiter des images à basse résolution, notre réseau peut s'appliquer sans aucune modification à des images de (très) haute résolution. Quelques exemples de résultats sont présentés Figure 8.1.

Institut Polytechnique de Paris
91120 Palaiseau, France

a) Image RAW originale, 3942×2612 pixels, 20 789 247 octets, 16 bpp

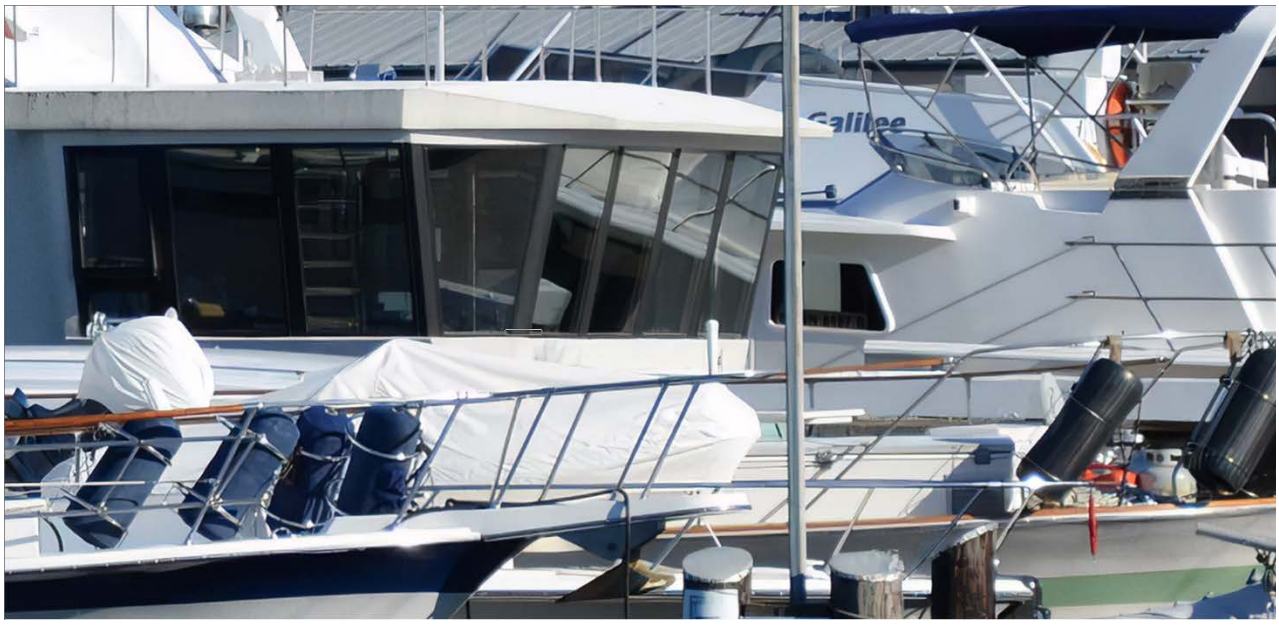
b) Reconstruction ×2, BPG q=25, 410 817 octets, 0,32 bpp

Figure 8.1 : Illustration des résultats de notre Reconstruction ×2 sur une image de très haute résolution et à fort niveau de détails

Institut Polytechnique de Paris
91120 Palaiseau, France

Il est important de noter que nous retrouvons un niveau de qualité quasi identique même en réalisant un zoom important comme illustré Figure 8.2.

a) Image RAW originale (zoom)



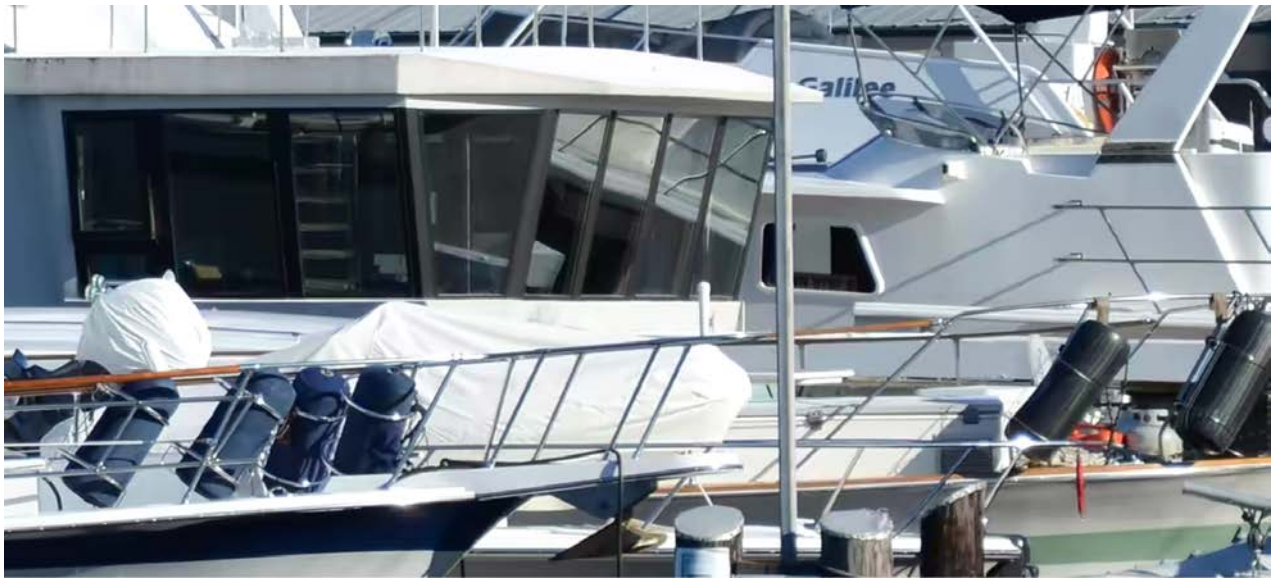
b) Reconstruction ×2 (zoom)

Figure 8.2 : Comparaison du niveau de détail obtenu même en réalisant un zoom conséquent

Comparons maintenant la différence entre notre Reconstruction ×2 et une compression BPG simple.

Si l'on compare sans zoom, nous ne voyons non plus aucune différence. De plus, si nous prenons le même zoom que présenté Figure 8.2, nous sommes forcés de constater que l'amélioration du notre Reconstruction ×2, bien que sensible, reste minime, comme illustré Figure 8.3.

Institut Polytechnique de Paris
91120 Palaiseau, France



a) Image BPG (zoom), q=34, 399 053 octets, 0,31 bpp

b) Reconstruction ×2 (zoom), BPG q=25, 410 817 octets, 0,32 bpp

Figure 8.3 : Comparaison entre BPG seul et Reconstruction ×2 sur un zoom peu texturé

Institut Polytechnique de Paris
91120 Palaiseau, France

Regardons en revanche un autre zoom de la même image présentant un niveau de textures plus important (Figure 8.4).

a) Image BPG (zoom)

b) Reconstruction ×2 (zoom)

Figure 8.4 : Comparaison entre BPG seul et Reconstruction ×2 sur un zoom très texturé

Nous pouvons constater ici que la différence entre les deux techniques est bien plus sensible sur une zone très texturée.

Notons néanmoins que dans tous les cas, même si les améliorations constatées ne sont pas majeures, notre méthode en reste supérieure au BPG seul et ce, même sur du contenu de très haute résolution. Le BPG étant l'état de l'art des codecs images, cela renforce donc la pertinence et la robustesse de notre technique.

Institut Polytechnique de Paris
91120 Palaiseau, France

Nous pouvons donc constater que notre réseau fonctionne particulièrement bien, dans le cas d'images génériques de très haute résolution et semble même être plus efficace qu'avec des images en basse-résolution. Cela pourrait s'expliquer par le fait que d'une part, le ESRGAN a de base été créé pour ce type de contenus et d'autre part que le réseau possède un nombre bien plus important d'informations à partir desquelles il peut réaliser ses prédictions sur une image en très haute résolution.

Dans tous les cas, nous pouvons ainsi imaginer une nouvelle application industrielle de nos travaux derniers. Nous rejoignons ainsi l'idée de base que nous avons évoquée depuis le début, en orientant nos recherches vers la création d'un module se plaçant en bout de toute chaîne de décompression, utilisable par tous sans modification de codec nécessaire.

Pour finir, une dernière piste de travaux futurs concerne l'application des méthodologies considérées sur des contenus vidéo, en assurant une certaine continuité temporelle entre les *frames*.

Institut Polytechnique de Paris
91120 Palaiseau, France

Liste des publications

HAMIS, Sébastien, ZAHARIA, Titus, et ROUSSEAU, Olivier. Image compression at very low bitrate based on deep learned super-resolution. In: *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*. IEEE, 2019. p. 128-133.

HAMIS, Sébastien, ZAHARIA, Titus, et ROUSSEAU, Olivier. Artifacts reduction for very low bitrate image compression with generative adversarial networks. In: *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*. IEEE, 2019. p. 76-81.
(Special Merit Paper Award)

HAMIS, Sébastien, ZAHARIA, Titus, et ROUSSEAU, Olivier. Optimizing image compression with deep super-resolution techniques. *IEEE Consumer Electronics Magazine*, 2020.

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 9. Références

[BeBound] « Be-Bound » [En ligne]. Available: https://be-bound.com/fr/.

[Ballé16] Ballé, J., Laparra, V., & Simoncelli, E. P. (2016, December). End-to-end optimization of nonlinear transform codes for perceptual quality. In 2016 Picture Coding Symposium (PCS) (pp. 1-5). IEEE.

[Hinton94] Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length and Helmholtz free energy. In Advances in neural information processing systems (pp. 3-10).

[Goodfellow14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[Santurkar18] Santurkar, S., Budden, D., & Shavit, N. (2018, June). Generative compression. In 2018 Picture Coding Symposium (PCS) (pp. 258-262). IEEE.

[Wallace91] Wallace, G. K. (1992). The JPEG still picture compression standard. IEEE transactions on consumer electronics, 38(1), xviii-xxxiv.

[Capon59] Capon, J. (1959). A probabilistic model for run-length coding of pictures. IRE Transactions on Information Theory, 5(4), 157-163.

[JPEG2000] «JPEG2000 Standard,» [En ligne]. Available: https://jpeg.org/jpeg2000/

[Heil89] Heil, C. E., & Walnut, D. F. (1989). Continuous and discrete wavelet transforms. SIAM review, 31(4), 628-666.

[Yu04] Yu, J. (2004, June). Advantages of uniform scalar dead-zone quantization in image coding system. In 2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No. 04EX914) (Vol. 2, pp. 805-808). IEEE.

[Baarir11] Baarir, Z. E. (2011). Entropy Encoding EBCOT (Embedded Block Coding with Optimized Truncation) In JPEG2000. International Journal of Computer Science Issues (IJCSI), 8(4), 531.

[WebP] «WebP,» [En ligne]. Available: https://developers.google.com/speed/webp/.

[Bankoski11] Bankoski, J., Wilkins, P., & Xu, Y. (2011). VP8 data format and decoding guide. RFC, 6386.

[Marpe03] Marpe, D., Schwarz, H., & Wiegand, T. (2003). Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard. IEEE Transactions on circuits and systems for video technology, 13(7), 620-636.

[VP8] « The VP8 Video Codec » [En ligne]. Available: https://www.slideshare.net/pfleidi/the-vp8-video-codec.

[BPG] Bellard, F. (2015). BPG Image format. URL https://bellard. org/bpg.

[Sullivan12] Sullivan, G. J., Ohm, J. R., Han, W. J., & 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.

Institut Polytechnique de Paris
91120 Palaiseau, France

[x265] « x265 HEVC Encoder / H.265 Video Codec,» [En ligne]. Available: http://x265.org/.

[Samet84] Samet, H. (1984). The quadtree and related hierarchical data structures. ACM Computing Surveys (CSUR), 16(2), 187-260.

[Sullivan98] Sullivan, G. J., & Wiegand, T. (1998). Rate-distortion optimization for video compression. IEEE signal processing magazine, 15(6), 74-90

[Zhu14] Zhu, W., Ding, W., Xu, J., Shi, Y., & Yin, B. (2014). Screen content coding based on HEVC framework. IEEE Transactions on Multimedia, 16(5), 1316-1326.

[Chen18] Chen, Y., Murherjee, D., Han, J., Grange, A., Xu, Y., Liu, Z., ... & Chiang, C. H. (2018, June). An overview of core coding tools in the AV1 video codec. In: 2018 Picture Coding Symposium (PCS) (pp. 41-45). IEEE.

[Massimo17] P. Massimo, «AOM - AV1, How does it work? » Jul 2017. [En ligne].

[Mukherjee15] Mukherjee, D., Han, J., Bankoski, J., Bultje, R., Grange, A., Koleszar, J., ... & Xu, Y. (2015). A technical overview of vp9—the latest open-source video codec. SMPTE Motion Imaging Journal, 124(1), 44-54.

[Kodak] « Kodak Lossless True Color Image Suite » [En ligne]. Available: http://r0k.us/graphics/kodak/.

[Zang04] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.

[Wang09] Wang, Z., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at signal fidelity measures. IEEE signal processing magazine, 26(1), 98-117.

[Lin11] Lin, W., & Kuo, C. C. J. (2011). Perceptual visual quality metrics: A survey. Journal of visual communication and image representation, 22(4), 297-312.

[Chikkerur11] Chikkerur, S., Sundaram, V., Reisslein, M., Karam, L. J. (2011). Objective video quality assessment methods: A classification, review, and performance comparison. IEEE Trans. on broadcasting, 57(2), 165-182

[Harikrishnan17] Harikrishnan, N. B., Menon, V. V., Nair, M. S., & Narayanan, G. (2017, February). Comparative evaluation of image compression techniques. In 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET) (pp. 1-4). IEEE.

[WyohKnott] WyohKnott, « Image formats comparison » [En ligne]. Available: https://wyohknott.github.io/image-formats-comparison/report.html.

[Huynh-Thu10] Huynh-Thu, Q., Garcia, M. N., Speranza, F., Corriveau, P., & Raake, A. (2010). Study of rating scales for subjective quality assessment of high-definition video. IEEE Transactions on Broadcasting, 57(1), 1-14.

[AV1Demo] « AV1 Still Demo,» [En ligne]. Available: https://people.xiph.org/~tdaede/av1stilldemo/.

[Kibeya14] Kibeya, H., Belghith, F., Ayed, M. A. B., & Masmoudi, N. (2014, November). A fast CU partitionning algorithm based on early detection of zero block quantified transform coefficients for HEVC standard. In International Image Processing, Applications and Systems Conference (pp. 1-5). IEEE.

[Lei16] Lei, J., Li, D., Pan, Z., Sun, Z., Kwong, S., & Hou, C. (2016). Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding. IEEE Transactions on Broadcasting, 63(1), 48-58.

[Fong15] Fong, C. K., Han, Q., & Cham, W. K. (2015). Recursive integer cosine transform for HEVC and future video coding standards. IEEE Transactions on Circuits and Systems for Video Technology, 27(2), 326-336.

[Fracastoro16] Fracastoro, G., Fosson, S. M., & Magli, E. (2016). Steerable discrete cosine transform. IEEE Transactions on Image Processing, 26(1), 303-314.

[Selesnick11] Selesnick, I. W., & Guleryuz, O. G. (2011, September). A diagonally-oriented DCT-like 2D block transform. In Wavelets and Sparsity XIV (Vol. 8138, p. 81381R). International Society for Optics and Photonics.

[Zhang16] Zhang, L., Xiu, X., Chen, J., Karczewicz, M., He, Y., Ye, Y., ... & Kim, W. S. (2016). Adaptive color-space transform in HEVC screen content coding. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 6(4), 446-459.

[Li18] Li, J., Li, B., Xu, J., Xiong, R., & Gao, W. (2018). Fully connected network-based intra prediction for image coding. IEEE Transactions on Image Processing, 27(7), 3236-3247.

[Wilson14] Wilson, K., & Snavely, N. (2014, September). Robust global translations with 1dsfm. In European Conference on Computer Vision (pp. 61-75). Springer, Cham.

[Birman20] Birman, R., Segal, Y., Hadar, O., & Benois-Pineau, J. (2020). Improvements of Motion Estimation and Coding using Neural Networks. arXiv preprint arXiv:2002.10439.

[Sezer08] Sezer, O. G., Harmanci, O., & Guleryuz, O. G. (2008, October). Sparse orthonormal transforms for image compression. In 2008 15th IEEE International Conference on Image Processing (pp. 149-152). IEEE.

[Puri16] Puri, S., Lasserre, S., & Le Callet, P. (2016, March). Annealed learning-based block transforms for HEVC video coding. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1135-1139). IEEE.

[Hartigan79] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics), 28(1), 100-108.

[Segal04] Segal, M. R. (2004). Machine learning benchmarks and random forest regression.

[Hearst98] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. IEEE Intelligent Systems and their applications, 13(4), 18-28.

[Baker98] Baker, M. R., & Patil, R. B. (1998). Universal approximation theorem for interval neural networks. Reliable Computing, 4(3), 235-239.

[Smith17] Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489.

[Qian99] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. Neural networks, 12(1), 145-151.

[Tieleman12] Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. Coursera: Neural networks for machine learning, 4(2), 26-31.

Institut Polytechnique de Paris
91120 Palaiseau, France

[Kingma14] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[Piramanayagam18] Piramanayagam, S., Saber, E., Schwartzkopf, W., & Koehler, F. W. (2018). Supervised classification of multisensor remotely sensed images using a deep learning framework. Remote Sensing, 10(9), 1429.

[Zhao16] Zhao, H., Gallo, O., Frosio, I., & Kautz, J. (2016). Loss functions for image restoration with neural networks. IEEE Transactions on computational imaging, 3(1), 47-57.

[Johnson16] Johnson, J., Alahi, A., & Fei-Fei, L. (2016, October). Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision (pp. 694-711). Springer, Cham.

[Simonyan14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[Kumar19] Kumar Basaveswara, S. (2019), CNN Architectures, a Deep dive, https://towardsdatascience.com/cnn-architectures-a-deep-dive-a99441d18049

[Russakovsky15] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[Huang17] Huang, Y., Shao, L., & Frangi, A. F. (2017). Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6070-6079).

[Uiboupin16] Uiboupin, T., Rasti, P., Anbarjafari, G., & Demirel, H. (2016, May). Facial image super resolution using sparse representation for improving face recognition in surveillance monitoring. In 2016 24th Signal Processing and Communication Application Conference (SIU) (pp. 437-440). IEEE.

[Dai16] Dai, D., Wang, Y., Chen, Y., & Van Gool, L. (2016, March). Is image super-resolution helpful for other vision tasks? In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1-9). IEEE.

[Haris18a] Haris, M., Shakhnarovich, G., & Ukita, N. (2018). Task-driven super resolution: Object detection in low-resolution images. arXiv preprint arXiv:1803.11316.

[Zhang18] Zhang, H., Liu, D., & Xiong, Z. (2018, May). Convolutional neural network-based video super-resolution for action recognition. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018) (pp. 746-750). IEEE.

[Friedman75] Friedman, J. H., Baskett, F., & Shustek, L. J. (1975). An algorithm for finding nearest neighbors. IEEE Transactions on computers, 100(10), 1000-1006.

[Irani91] Irani, M., & Peleg, S. (1991). Improving resolution by image registration. CVGIP: Graphical models and image processing, 53(3), 231-239.

[Freedman11] Freedman, G., & Fattal, R. (2011). Image and video upscaling from local self-examples. ACM Transactions on Graphics (TOG), 30(2), 1-11.

[Sun08] Sun, J., Xu, Z., & Shum, H. Y. (2008, June). Image super-resolution using gradient profile prior. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

Institut Polytechnique de Paris
91120 Palaiseau, France

[Yang10] Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. IEEE transactions on image processing, 19(11), 2861-2873.

[Timofte13] Timofte, R., De Smet, V., & Van Gool, L. (2013). Anchored neighborhood regression for fast example-based super-resolution. In Proceedings of the IEEE international conference on computer vision (pp. 1920-1927).

[Timofte14] Timofte, R., De Smet, V., & Van Gool, L. (2014, November). A+: Adjusted anchored neighborhood regression for fast super-resolution. In Asian conference on computer vision (pp. 111-126). Springer, Cham.

[Dong15a] Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2), 295-307.

[Romano17] Romano, Y., Isidoro, J., & Milanfar, P. (2017). RAISR: rapid and accurate image superresolution. IEEE Transactions on Computational Imaging, 3(1), 110-125.

[Nack17] Nack, J. (2017). Saving you bandwidth through machine learning. https://blog.google/products/google-plus/saving-you-bandwidth-through-machine-learning/

[Jeong10] Jeong, S. C., & Song, B. C. (2010). Fast Super-Resolution Algorithm Based on Dictionary Size Reduction Using k-Means Clustering. ETRI journal, 32(4), 596-602.

[Sandeep16] Sandeep, P., & Jacob, T. (2016). Single image super-resolution using a joint GMM method. IEEE Transactions on Image Processing, 25(9), 4233-4244.

[Bundy84] Bundy, A., & Wallen, L. (1984). Difference of gaussians. In Catalogue of Artificial Intelligence Tools (pp. 30-30). Springer, Berlin, Heidelberg.

[Wang20] Wang, Z., Chen, J., & Hoi, S. C. (2020). Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence

[Dong16] Dong, C., Loy, C. C., & Tang, X. (2016, October). Accelerating the super-resolution convolutional neural network. In European conference on computer vision (pp. 391-407). Springer, Cham.

[Lai17] Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 624-632).

[Wang18a] Wang, Y., Perazzi, F., McWilliams, B., Sorkine-Hornung, A., Sorkine-Hornung, O., & Schroers, C. (2018). A fully progressive approach to single-image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 864-873).

[Haris18b] Haris, M., Shakhnarovich, G., & Ukita, N. (2018). Deep back-projection networks for super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1664-1673

[List03] List, P., Joch, A., Lainema, J., Bjontegaard, G., & Karczewicz, M. (2003). Adaptive deblocking filter. IEEE transactions on circuits and systems for video technology, 13(7), 614-619.

[Dong15b] Dong, C., Deng, Y., Change Loy, C., & Tang, X. (2015). Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 576-584).

Institut Polytechnique de Paris
91120 Palaiseau, France

[Set11] Set, B. S. D. Benchmarks 500 (bsds500), 2011. *URL http://www. eecs. berkeley. edu/Research/Projects/CS/vision/grouping/resources. html*.

[Foi07] Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE transactions on image processing*, *16*(5), 1395-1411.

[Chen18] Chen, H., He, X., Ren, C., Qing, L., & Teng, Q. (2018). CISRDCNN: Super-resolution of compressed images using deep convolutional neural networks. *Neurocomputing*, *285*, 204-219.

[Yang08] Yang, J., Wright, J., Huang, T., & Ma, Y. (2008, June). Image super-resolution as sparse representation of raw image patches. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1-8). IEEE.

[Yu18] Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., & Huang, T. (2018). Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*.

[Lim17] Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 136-144).

[Zhang16] Zhang, R., Isola, P., & Efros, A. A. (2016, October). Colorful image colorization. In *European conference on computer vision* (pp. 649-666). Springer, Cham.

[Deng09] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[He16] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[Baldassarre17] Baldassarre, F., Morín, D. G., & Rodés-Guirao, L. (2017). Deep koalarization: Image colorization using cnns and inception-resnet-v2. *arXiv preprint arXiv:1712.03400*.

[Szegedy17] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

[Baig17] Baig, M. H., & Torresani, L. (2017). Multiple hypothesis colorization and its application to image compression. *Computer Vision and Image Understanding*, *164*, 111-123.

[Zhang] R. Zhang, «Colorful Image Colorization,» [En ligne]. Available: https://github.com/richzhang/colorization.

[Ledig17] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).

[LeCun16] LeCun, Y. (2016). What are some recent and potentially upcoming breakthroughs in deep learning?

Institut Polytechnique de Paris
91120 Palaiseau, France

[Wiki18] Wiki, S. A. (2018). A beginner's guide to generative adversarial networks (gans). *Skymind, last modified https://skymind. ai/wiki/generative-adversarial-network-gan*.

[Goodfellow16] Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

[Mescheder18] Mescheder, L., Geiger, A., & Nowozin, S. (2018). Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406*.

[Rippel17] Rippel, O., & Bourdev, L. (2017). Real-time adaptive image compression. *arXiv preprint arXiv:1705.05823*.

[Mao15] Mao, T. (2015). Mining one hundred million creative commons Flickr images dataset to Flickr tourist index. *International Journal of Future Computer and Communication*, *4*(2), 104.

[Agustsson19] Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., & Gool, L. V. (2019). Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 221-231).

[Theis17] Theis, L., Shi, W., Cunningham, A., & Huszár, F. (2017). Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*.

[Svoboda16] Svoboda, P., Hradis, M., Barina, D., & Zemcik, P. (2016). Compression artifacts removal using convolutional neural networks. *arXiv preprint arXiv:1605.00366*.

[Kuznetsova20] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., ... & Duerig, T. (2020). The open images dataset v4. *International Journal of Computer Vision*, 1-26.

[Cordts16] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213-3223).

[Galteri17] Galteri, L., Seidenari, L., Bertini, M., & Del Bimbo, A. (2017). Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4826-4835).

[Martin01] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001, July). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (Vol. 2, pp. 416-423). IEEE.

[Lin14] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

[Ren15] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

[Gross16] Gross, S., & Wilber, M. (2016). Training and investigating residual nets. *Facebook AI Research*, *6*.

[Shi16] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., ... & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-

Institut Polytechnique de Paris
91120 Palaiseau, France

pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1874-1883)

[Gatys15] Gatys, L., Ecker, A. S., & Bethge, M. (2015). Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems* (pp. 262-270).

[Huang15] Huang, J. B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5197-5206).

[Kim16] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1637-1645).

[Wang18b] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., ... & Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 0-0).

[Blau18] Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., & Zelnik-Manor, L. (2018). The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 0-0).

[ZsDongHao] ZsDongHaohttps://github.com/zsdonghao, A Tensorlayer SRGAN Implementation, https://github.com/tensorlayer/srgan

[Dong17] Dong, H., Supratak, A., Mai, L., Liu, F., Oehmichen, A., Yu, S., & Guo, Y. (2017, October). Tensorlayer: a versatile library for efficient deep learning development. In *Proceedings of the 25th ACM international conference on Multimedia* (pp. 1201-1204).

[Agustsson17] Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 126-135).

[Huiskes08] Huiskes, M. J., & Lew, M. S. (2008, October). The MIR flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval* (pp. 39-43).

[Ma17] Ma, C., Yang, C. Y., Yang, X., & Yang, M. H. (2017). Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, *158*, 1-16.

[Mittal12] Mittal, A., Soundararajan, R., & Bovik, A. C. (2012). Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, *20*(3), 209-212.

[Novak90] Novak, T. P., & Hoffman, D. L. (1990). Residual scaling: an alternative to correspondence analysis for the graphical representation of residuals from log-linear models. *Multivariate Behavioral Research*, *25*(3), 351-370.

[Jolicoeur-Martineau18] Jolicoeur-Martineau, A. (2018). The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734*.

[Ferwerda03] Ferwerda, J. A. (2003, June). Three varieties of realism in computer graphics. In Human Vision and Electronic Imaging VIII (Vol. 5007, pp. 290-297). International Society for Optics and Photonics.

Institut Polytechnique de Paris
91120 Palaiseau, France

[Union12] Union, I. T. (2012). Methodology for the subjective assessment of the quality of television pictures ITU-R recommendation BT. 500-13. Tech. Rep

[Talebi18] Talebi, H., & Milanfar, P. (2018). NIMA: Neural image assessment. IEEE Transactions on Image Processing, 27(8), 3998-4011.

[Moorthy11] 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.

[Zhai05] Zhai, G., Zhang, W., Yang, X., & Xu, Y. (2005, November). Image quality assessment metrics based on multi-scale edge presentation. In IEEE Workshop on Signal Processing Systems Design and Implementation, 2005. (pp. 331-336). IEEE.

[Bensaid Ghaly18] Bensaied Ghaly, R. (2018). Subjective quality assessment: a study on the grading scales: illustrations for stereoscopic and 2D video content (Doctoral dissertation, Evry, Institut national des télécommunications).

[Hamis19] Hamis, S., Zaharia, T., & Rousseau, O. (2019, September). Artifacts reduction for very low bitrate image compression with generative adversarial networks. In 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin) (pp. 76-81). IEEE

[Hamis20] Hamis, S., Zaharia, T., & Rousseau, O. (2020). Optimizing image compression with deep super-resolution techniques. IEEE Consumer Electronics Magazine

[Zhou17] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). East: an efficient and accurate scene text detector. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 5551-5560).

Institut Polytechnique de Paris
91120 Palaiseau, France

Chapitre 10. Annexes

Annexe 1. Comparaison BPG, SRCNN et ré-colorisation

Figure 10.1 : BPG+SRCNN+Colorisation non satisfaisante (taille réelle)

Institut Polytechnique de Paris
91120 Palaiseau, France

Figure 10.2 : BPG+SRCNN+Colorisation réaliste (taille réelle)

Institut Polytechnique de Paris
91120 Palaiseau, France

Annexe 2. Images complètes de celles utilisée Figure 5.34

Original
BPG,
q=41

Institut Polytechnique de Paris
91120 Palaiseau, France

ESRGAN

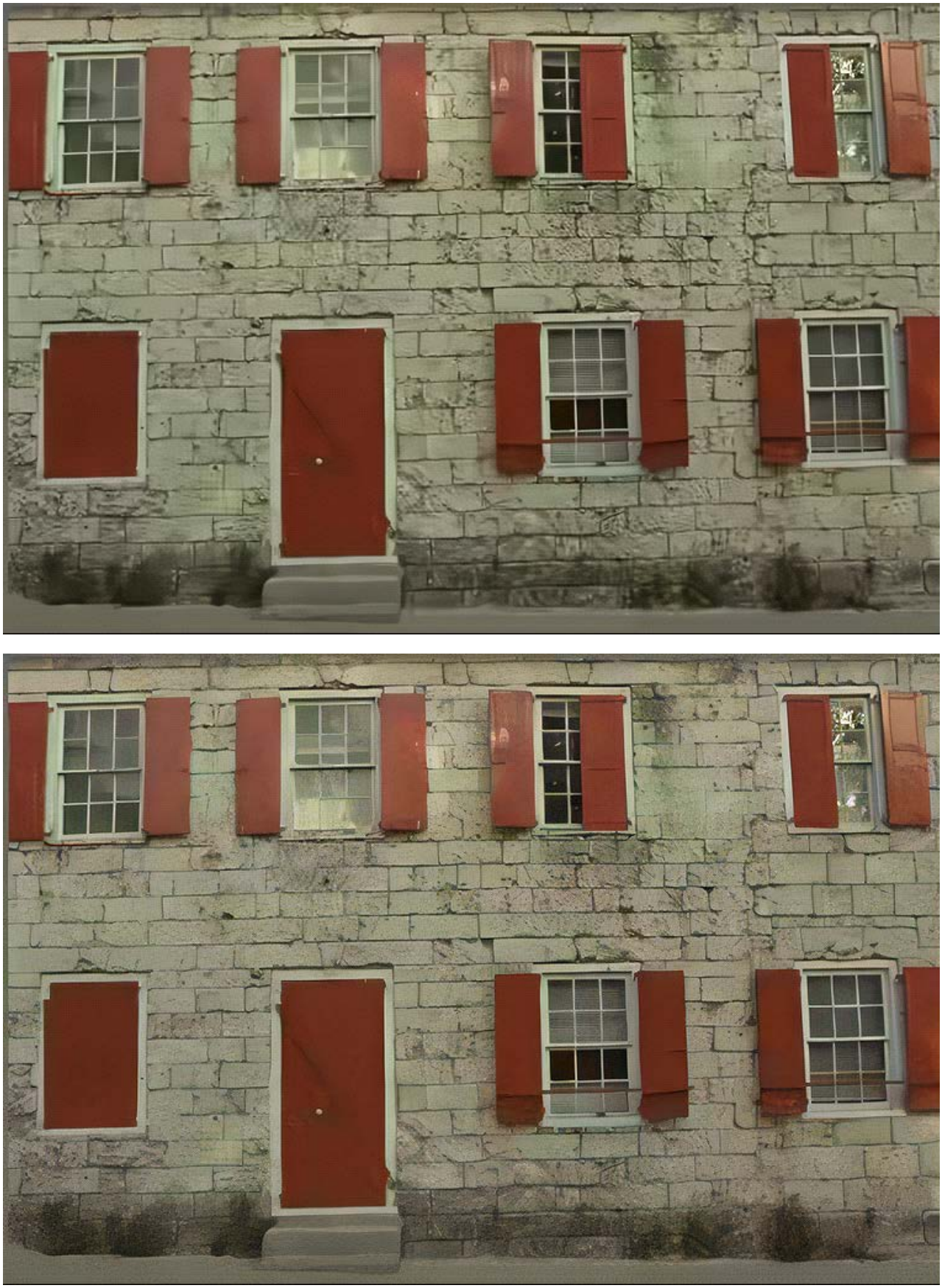
Institut Polytechnique de Paris
91120 Palaiseau, France

SRGAN
modifié

Figure 10.3 : Versions non zoomées de l'image traitée présentée dans la Figure 5.34

Institut Polytechnique de Paris
91120 Palaiseau, France

Annexe 3. Images tailles réelles illustrant les différences de qualité entre SRGAN ×1_gen (en haut) et ESRGAN×1_gen (en bas)



Kodim01

Institut Polytechnique de Paris
91120 Palaiseau, France

Kodim03

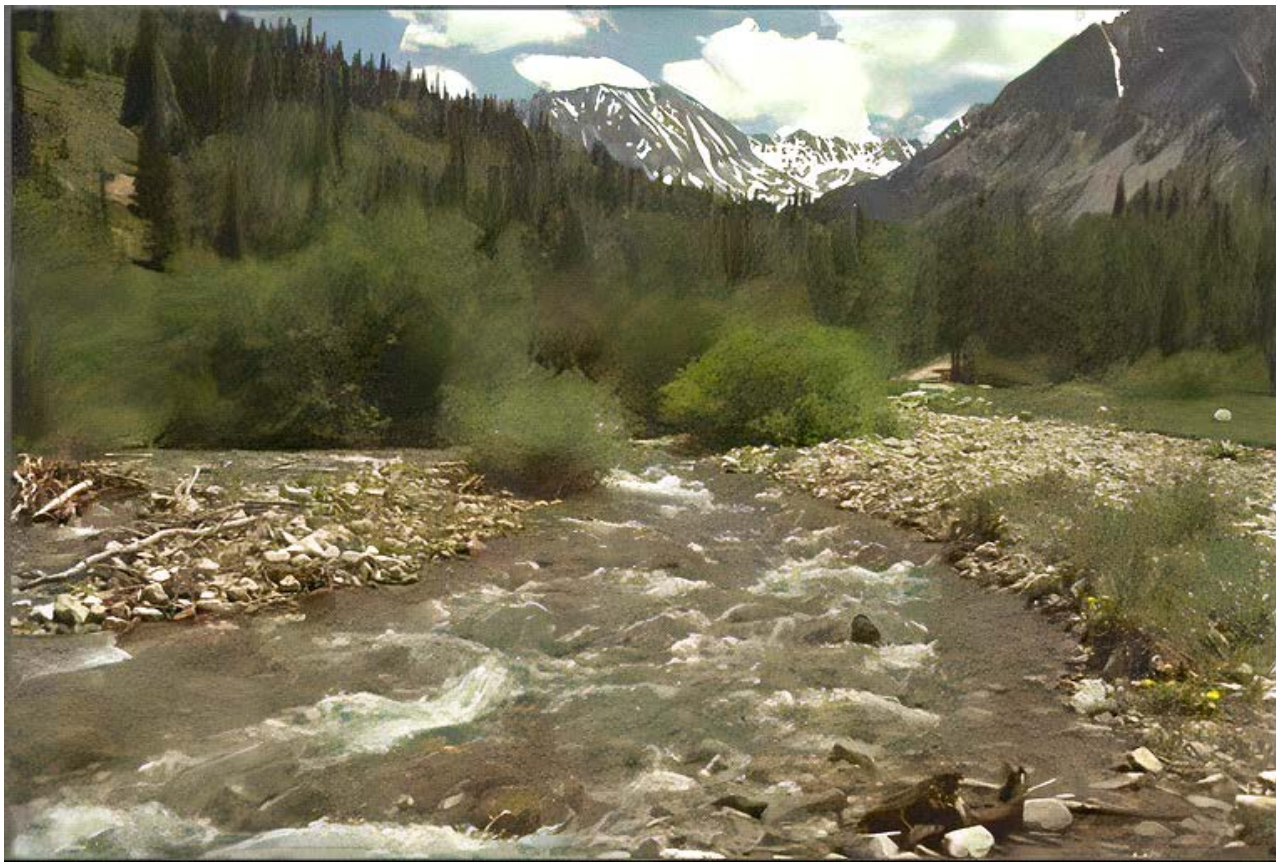
Institut Polytechnique de Paris
91120 Palaiseau, France

Kodim06

Institut Polytechnique de Paris
91120 Palaiseau, France

Kodim12

Institut Polytechnique de Paris
91120 Palaiseau, France



Kodim13

Institut Polytechnique de Paris
91120 Palaiseau, France

Kodim21

Institut Polytechnique de Paris
91120 Palaiseau, France