Guiding neural networks for image colorization through user interactions

Hernan Carrillo

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

SPÉCIALITÉ: INFORMATIQUE

Par : **Hernan CARRILLO**

Colorisation d'images avec réseaux de neurones guidés par l'interaction humaine

Guiding neural networks for image colorization through user interactions

Sous la direction de : Aurélie BUGEAU

Soutenue le : 01/02/2024

Membres du jury :

Aurélie	BUGEAU	Professeure des Universités	Université de Bordeaux	Directrice de thèse
Alasdair	NEWSON	Maître de Conférences HDR	Sorbonne Université	Rapporteur
Julien	RABIN	Maître de Conférences HDR	Université de Caen	Rapporteur
Andrés	ALMANSA	Directeur de Recherche	Université Paris Cité	Examinateur
Valérie	GOUET-BRUNET	Directrice de Recherche	IGN	Examinatrice
Nicolas	PAPADAKIS	Directeur de Recherche	Université de Bordeaux	Président du jury
Michaël	CLÉMENT	Maître de Conférences	Bordeaux INP	Invité (encadrant)
Rémi	GIRAUD	Maître de Conférences	Bordeaux INP	Invité

– 2023 –

Résumé

La colorisation est le processus qui consiste à ajouter des couleurs aux images en niveaux de gris. C'est une tâche importante dans la communauté de l'édition d'images et de l'animation. Bien que des méthodes de colorisation automatique existent, elles produisent souvent des résultats insatisfaisants en raison de défauts tels que le débordement de couleur, l'incohérence, des couleurs non naturelles et la nature non trivial du problème. Par conséquent, une intervention manuelle est souvent nécessaire pour obtenir le résultat souhaité. En conséquence, il y a un intérêt croissant à automatiser le processus de colorisation tout en permettant aux artistes d'ajouter leur propre style et vision. Dans cette thèse, nous étudions divers formats d'interaction en guidant les couleurs sur des zones spécifiques d'une image, ou en les transférant à partir d'une image ou d'un objet de référence. Nous introduisons deux méthodes de colorisation semi-automatiques. Tout d'abord, nous décrivons une architecture d'apprentissage profond pour la colorisation d'images qui prend en compte les images de référence de l'utilisateur. Notre deuxième méthode utilise un modèle de diffusion pour coloriser des dessins en utilisant des indications de couleur fournies par l'utilisateur.

Cette thèse commence par l'état de l'art des méthodes de colorisation d'images, des espaces de couleur, des métriques d'évaluation et des fonctions de perte. Bien que les méthodes de colorisation récentes basées sur des techniques d'apprentissage profond obtiennent les meilleurs résultats, ces méthodes sont basées sur des architectures complexes et un grand nombre de fonctions de perte, ce qui rend difficile leur compréhension. Pour cela, nous utilisons une architecture simple afin d'analyser l'impact de différents espaces de couleur et fonctions de perte.

Ensuite, nous proposons une nouvelle couche d'attention appelée *super-attention* qui utilise des superpixels. Elle permet d'établir des correspondances entre les caractéristiques hautes résolutions de paires d'images cible et référence. Cette proposition permet d'atténuer le problème de la complexité quadratique des couches d'attention. De plus, elle aide à surmonter les défauts de débordement de couleur dans la tâche de colorisation. Nous étudions son utilisation pour le transfert de couleur, et pour la colorisation basée sur des exemples. Nous proposons également une extension de ce modèle afin de guider spécifiquement la colorisation sur des objets segmentés.

Enfin, nous proposons un modèle de diffusion probabiliste basé sur des conditionnements implicites et explicites, pour apprendre à coloriser des dessins au trait. Notre approche permet d'ajouter des interactions utilisateur à travers des indices de couleur explicites tout en s'appuyant sur l'entraînement du modèle de diffusion principal. Nous utilisons un encodeur spécifique qui apprend à extraire des informations sur les indices de couleur fournis par l'utilisateur. Ce modèle permet d'obtenir des images colorisées diverses et de haute qualité.

Mots-clés

Mécanisme d'attention, Colorisation, Transfert de couleurs, Correspondances non locales, Superpixels.

Abstract

Colorization is the process of adding colors to grayscale images. It is an important task in the image-editing and animation community. Although automatic colorization methods exist, they often produce unsatisfying results due to artifacts such as color bleeding, inconsistency, unnatural colors, and the ill-posed nature of the problem. Manual intervention is often necessary to achieve the desired outcome. Consequently, there is a growing interest in automating the colorization process while allowing artists to transfer their own style and vision to the process. In this thesis, we investigate various interaction formats by guiding colors of specific areas of an image or transferring them from a reference image or object. As part of this research, we introduce two semi-automatic colorization frameworks. First, we describe a deep learning architecture for exemplar-based image colorization that takes into account user's reference images. Our second framework uses a diffusion model to colorize line art using user-provided color scribbles.

This thesis first delves into a comprehensive overview of state-of-the-art image colorization methods, color spaces, evaluation metrics, and losses. While recent colorization methods based on deep-learning techniques are achieving the best results on this task, these methods are based on complex architectures and a high number of joint losses, which makes the reasoning behind each of these methods difficult. Here, we leverage a simple architecture in order to analyze the impact of different color spaces and several losses.

Then, we propose a novel attention layer based on superpixel features to establish robust correspondences between high-resolution deep features from target and reference image pairs, called *super-attention*. This proposal deals with the quadratic complexity problem of the non-local calculation in the attention layer. Additionally, it helps to overcome color bleeding artifacts. We study its use in color transfer and exemplar-based colorization. We finally extend this model to specifically guide the colorization on segmented objects.

Finally, we propose a diffusion probabilistic model based on implicit and explicit conditioning mechanism, to learn colorizing line art. Our approach enables the incorporation of user guidance through explicit color hints while leveraging on the prior knowledge from the trained diffusion model. We condition with an application-specific encoder that learns to extract meaningful information on user-provided scribbles. The method generates diverse and high-quality colorized images.

Keywords

Attention mechanism, Colorization, Color transfers, Non-local matching, Superpixels.

Unité de recherche

LaBRI, UMR 5800, Université de Bordeaux, 33400 Talence, France.

Acknowledgements

I am incredibly grateful for the help and support I received from many people on my PhD journey. Without them, I could not have made it this far.

Firstly, I want to express my immense gratitude to my advisors, Aurélie and Michaël. Their guidance and invaluable advice were crucial to me, and their belief in my abilities gave me the strength to keep going. Additionally and more important their patience, courage and human touch helped me get through the tough times.

I also want to thank the jury members for their valuable time reviewing my thesis.

To my colleagues, especially Rupayan, Warren, Jean, and Samuel, were a great source of support and inspiration for many research ideas.

I would like to thank all my friends, without whom these three years in Bordeaux would not have been as interesting, fun, and exciting as they were. I'm incredibly thankful to Roberto, Allison, Ricardo, Gustavo, Leslie, and Alix. We had many great times together, and I am grateful for the good times made even better by their presence.

To my family, especially my parents, Hernán and Sonia, have been my strongest support. I cannot express enough how much their love and sacrifices mean to me. And to my brothers, Henry and Snaider, thank you for your unwavering support and advice.

I am also immensely grateful to the Agence National de Recherche for the Post-ProdLEAP project, which provided the funding that allowed me to pursue this thesis.

Lastly, I want to express my gratitude to everyone who has been part of this journey.

Table of contents

List of Figures

List of Tables

Résumé étendu en français

La colorisation consiste à ajouter des couleurs plausibles à des images en niveaux de gris, dans le but de produire des images visuellement attrayantes tout en évitant les artefacts indésirables ou les couleurs incorrectes. Cette application est utilisée dans de nombreux domaines : la restauration de photos/vidéos anciennes, la post-production cinématographique ou l'animation. Cependant, les processus actuels sont souvent longs et fastidieux, car ils dépendent fortement de l'intervention manuelle de l'utilisateur ou l'artiste. Par exemple, dans l'industrie de la post-production, les outils professionnels disponibles permettent aux artistes d'atteindre des résultats de haute qualité mais nécessitent une longue intervention humaine. Un exemple de méthode de colorisation professionnelle est montré dans la Figure 1. Des images clés sont d'abord manuellement choisies à partir de la vidéo, chaque plan comportant environ 75 images. Elles correspondent à de nouveaux événements ou objets entrant dans la scène. Par la suite, des artistes, documentalistes ou historiens lient manuellement chaque objet segmenté à une couleur appropriée, en tenant compte de l'exactitude historique ou de l'attrait visuel. Cette étape peut prendre jusqu'à un jour pour des plans complexes comportant de nombreux objets ou des visages proches. Un algorithme de suivi d'objet est finalement utilisé pour propager les couleurs de chaque objet au reste des images.

Figure 1: Processus de colorisation pour *MARIA BY CALLAS* - Tom Volf, sorti le 13 décembre 2017. Première ligne : l'image clé originale et sa version colorisée après la post-production. Seconde ligne : chaque étape du traitement, de gauche à droite : segmentation de l'objet, coloration manuelle des objets, et suivi de l'objet avec propagation de la couleur.

L'automatisation du processus de colorisation peut considérablement améliorer le flux de travail des artistes, mais elle est difficile en raison de son ambiguïté inhérente. Cela est dû au fait que plusieurs couleurs plausibles peuvent être attribuées au même pixel gris d'une image, en fonction de divers facteurs, tels que les mêmes objets avec différentes couleurs et/ou des structures complexes sur l'image. L'importance de l'interaction découle justement du fait qu'il n'existe pas de solution unique et objectivement correcte lorsqu'il s'agit de coloriser une image en niveaux de gris (voir Figure 2). De plus, les artistes, les professionnels de la restauration ou les animateurs peuvent avoir des intentions spécifiques

ou des visions créatives en tête lors de la colorisation d'une image. Ces intentions varient considérablement et ne peuvent pas être facilement identifiées par des algorithmes entièrement automatisés. Par conséquent, l'apport de l'utilisateur devient essentiel pour guider le processus de colorisation afin d'être cohérent avec la vision ou les exigences spécifiques de l'utilisateur.

Au cours des vingt dernières années, plusieurs algorithmes de colorisation ont été proposés dans la littérature. Ils sont généralement classés en trois types qui ne sont pas mutuellement exclusifs : la colorisation basée sur des *scribbles* (indices de couleurs sous la forme de traits), la colorisation basée sur des images exemples, et, récemment, les méthodes de colorisation basées sur l'apprentissage. Ce dernier groupe de méthodes utilise des modèles génératifs profonds pour générer des images colorisées basées sur des a priori appris, et obtiennent des résultats prometteurs. Cependant, il reste encore un long chemin à parcourir pour obtenir des résultats satisfaisants pour les applications réelles, en particulier lorsque l'utilisateur souhaite intervenir dans le processus de colorisation automatique.

Figure 2: Exemple de problème d'ambiguïté de colorisation d'image. À partir d'une image en niveaux de gris (première colonne), plusieurs colorisations valides (colonnes suivantes) peuvent être générées. Le t-shirt de la femme peut par exemple prendre différentes couleurs, et les trois sont visuellement plausibles. Figure extraite de (Huang et al., 2022).

Dans la littérature, il existe deux manières principales d'intégrer l'interaction de l'utilisateur dans les modèles d'apprentissage profond pour la tâche de colorisation. Dans la première, l'utilisateur fournit une image de référence qui montre les couleurs désirées à transférer. Dans la deuxième, il peint des traits de couleur sur différentes régions de l'image en niveaux de gris. Dans cette thèse, nous explorons comment les interactions fournies par l'utilisateur peuvent être introduites dans un modèle d'apprentissage profond pour améliorer la précision et le réalisme de la colorisation.

Contexte de la thèse. Cette thèse fait partie du projet ANR PostProdLEAP, dont l'objectif était de développer de nouveaux outils pour la post-production d'archives vidéo en tirant parti à la fois des approches récentes d'apprentissage profond et des approches basées sur les patchs ou variationnelles. Les artefacts fréquemment observés avec les méthodes d'apprentissage profond comprennent la perte de détails, les discontinuités spatiales et temporelles, et le dépassement des couleurs le long des bords. Ces artefacts rendent les approches d'apprentissage profond récentes inappropriées pour la post-production professionnelle.

Dans cette thèse, nous avons exploré et conçu des modèles d'apprentissage profond pour relever les défis de la colorisation automatique d'images, dans le cadre du projet PostProdLEAP. Nous avons appris de discussions avec des artistes et des documentalistes

de Composite Films, une société de post-production spécialisée dans la restauration et la colorisation de films. Ces conversations ont inspiré ce travail de recherche afin de développer de nouvelles techniques intégrant de manière transparente les interactions utilisateur avec l'apprentissage profond. Nos contributions ont pour objectif de faciliter l'implication des artistes dans le processus de colorisation et de leur donner plus de contrôle sur les résultats.

Problématique

La colorisation d'images est une tâche qui consiste à prédire les valeurs de couleur de chaque pixel dans des images en niveaux de gris. Cela est difficile car le modèle doit apprendre les relations complexes entre les intensités des niveaux de gris et les couleurs correspondantes, tout en considérant qu'il existe plusieurs colorisations plausibles (problème mal posé) pour une image donnée en niveaux de gris.

Récemment, il a été démontré que l'apprentissage profond est très efficace pour les tâches de colorisation d'image. Les modèles d'apprentissage profond peuvent apprendre des correspondances complexes entre les données d'entrée et de sortie, même en présence de bruit et autres artefacts. Une approche de la colorisation automatique d'image par apprentissage profond consiste à formuler le problème comme une tâche de régression (Cheng et al., 2015; Larsson et al., 2016). Le modèle est entraîné sur un ensemble d'images en niveaux de gris et leurs images colorisées correspondantes. Il apprend à prédire les valeurs de chrominance de chaque pixel dans l'image en niveaux de gris, compte tenu des intensités de gris. Une autre approche de la colorisation automatique d'image par apprentissage profond consiste à formuler le problème comme une tâche de classification. Le modèle apprend à classer chaque pixel de l'image en niveaux de gris dans des ensembles discrets de couleurs prédéfinies (Zhang et al., 2017; Iizuka et al., 2016). Ces ensembles de couleurs prédéfinies sont généralement des ensembles de couleurs choisis manuellement en fonction de la distribution des couleurs dans l'ensemble de données d'entraînement. Dans les deux approches, le modèle d'apprentissage profond est entraîné pour apprendre une transformation Φ d'une image en niveaux de gris T_L en une version colorisée $\hat{T}$. Cependant, comme détaillé précédemment, il y a une ambiguïté dans la sélection d'une couleur unique pour un pixel gris dans l'image en niveaux de gris. Pour aborder cela, des connaissances préalables de l'utilisateur sont souvent ajoutées au processus de colorisation pour garantir que l'image finale ait des propriétés souhaitables.

Défis et contraintes

Les réseaux de neurones profonds ont obtenu des résultats impressionnants dans la colorisation automatique des images (He et al., 2018; Vitoria et al., 2020; Huang et al., 2022). Cependant, intégrer des *a priori* utilisateurs dans ces réseaux n'est pas direct. Il existe plusieurs défis scientifiques et applicatifs à surmonter :

I. **Comprendre l'intention de l'utilisateur.** Il est important de saisir ce que l'utilisateur souhaite accomplir avec ses directives. Par exemple, veut-il changer la couleur générale de l'image, ou ajuster la couleur d'objets spécifiques. Pour la première question, les méthodes actuelles s'appuient généralement sur la colorisation

en utilisant le contexte complet d'une image de référence. Cependant, pour la seconde question, les utilisateurs utilisent habituellement à de nombreuses indications de couleur, ce qui peut être fastidieux.

Image initiale + points de couleur

Résultat (Yun et al., 2023)

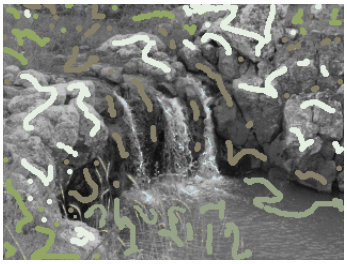


Image initiale + scribbles

Résultat (Levin et al., 2004)

Figure 3: Exemple de méthodes de colorisation utilisant l'interaction de l'utilisateur. La première rangée présente la méthode (Yun et al., 2023) qui tire parti des points des couleurs de l'utilisateur. La deuxième rangée montre les résultats sur une méthode classique (Levin et al., 2004) qui utilise des gribouillis de couleurs de l'utilisateur.

II. **Représenter les directives de l'utilisateur.** Les directives de l'utilisateur prennent différentes formes, telles que des images de référence ou des indices de couleurs. Par exemple, les *scribbles* sont des lignes ou traits à main levée que les utilisateurs dessinent sur des images en gris pour guider le processus de colorisation. Ils constituent un moyen pratique d'indiquer les couleurs souhaitées, en particulier dans les régions complexes ou à leurs frontières. D'autre part, les indices de couleur sont des suggestions plus discrètes, souvent représentées sous forme de points ou de petites taches. Ils sont moins invasifs et plus rapides à appliquer que les *scribbles*, mais ils peuvent exiger que l'algorithme de colorisation fasse plus de suppositions sur les zones environnantes (la Figure 3 montre un exemple de ces deux types d'interactions utilisateurs). Chacun de ces types de directives nécessite une manière unique d'être compris et utilisé par l'architecture du réseau. Ainsi, il est crucial de trouver un équilibre où l'entrée de l'utilisateur a un impact significatif sur les résultats, tout en permettant au réseau de coloriser les régions sans indices et de rectifier toute erreur ou irrégularité dans les directives fournies.

III. **Intégrer les directives de l'utilisateur dans le modèle.** Les directives de

l'utilisateur doivent être intégrées dans le réseau de neurones profond de manière à préserver la qualité globale de l'image colorisée. Cela peut être difficile, car les directives de l'utilisateur peuvent être incohérentes avec les données d'entraînement. Par conséquent, assurer l'équilibre entre les entrées de couleur de l'utilisateur et le maintien de la capacité de généralisation du réseau est une tâche difficile. Par exemple, si un réseau est entraîné à coloriser des animaux et reçoit une image en niveaux de gris d'un chat, le réseau pourrait favoriser les couleurs typiques trouvées dans l'ensemble de données d'entraînement. Mais si un utilisateur fournit des indices de couleur comme un orange clair pour la fourrure, l'approche idéale pourrait consister à mélanger la couleur préférée de l'utilisateur tout en préservant les couleurs naturelles des yeux et du nez du chat à partir des connaissances apprises par le modèle.

IV. **Créer des ensembles de données tout en préservant la généralisation.** Concevoir un grand ensemble de données qui imite les entrées des utilisateurs tout en améliorant la capacité de généralisation d'un réseau de neurones profond n'est pas évident. Par exemple, dans les approches basées sur des *scribbles*, il n'existe pas de moyen évident de simuler des traits de couleurs dans le réseau. Un autre exemple concerne les méthodes de colorisation basées sur des exemples, où différentes métriques sont utilisées pour trouver des paires cible/référence.

V. **Gérer l'ambiguïté des couleurs.** La colorisation d'images est une tâche ambiguë. Cela signifie qu'il peut y avoir de nombreuses colorisations possibles pour la même image en niveaux de gris. Un facteur qui contribue à cette ambiguïté est la texture de l'image. Cela est dû au fait que de nombreux objets dans le monde réel ont des variations de couleur tout en conservant une texture similaire. Par exemple, différents types de pommes peuvent avoir des représentations de texture en niveaux de gris similaires mais des couleurs différentes. Par conséquent, les utilisateurs peuvent fournir des informations de couleur plus précises qu'un modèle d'apprentissage profond entièrement automatique ne peut apprendre par lui-même. Cela est particulièrement vrai pour les régions ou les frontières complexes. Les directives de l'utilisateur peuvent alors être très utiles pour améliorer les résultats des algorithmes de colorisation d'images.

VI. **Éviter le débordement des couleurs.** Un artefact courant dans la tâche de colorisation automatique est le débordement des couleurs, lorsque les couleurs d'un objet se répandent dans les objets adjacents. Par exemple, la Figure 4 montre deux parties où le débordement des couleurs est présent ; dans ces deux parties, la méthode de colorisation mélange le bleu du ciel avec le rose du jouet flottant. Cela se produit lorsque les informations de bord ne sont pas prises en compte dans le processus de colorisation. Par conséquent, développer des approches utilisant des techniques prenant en compte les bords dans le processus de colorisation pourrait être un moyen de gérer les artefacts courants.

Intégrer le guidage de l'utilisateur dans les réseaux neuronaux profonds pour la colorisation d'images nécessite une approche qui aborde les six défis généraux précédents. En plus de ces défis généraux, nous avons identifié d'autres problèmes spécifiques liés aux propositions actuelles de colorisation interactive basée sur l'apprentissage.

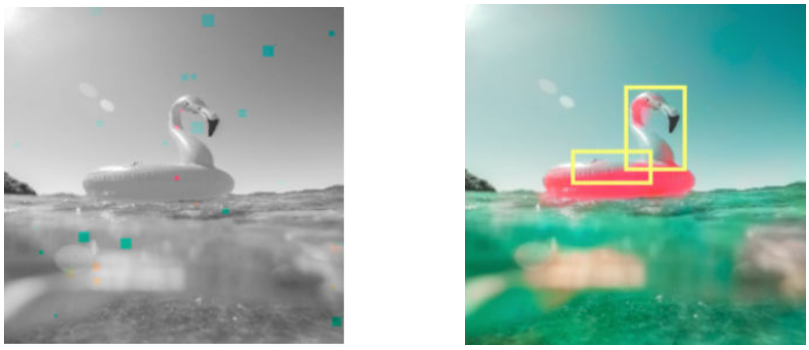


Image initiale + indications de couleur Résultat (Zhang et al., 2017)

Figure 4: Exemple de débordement de couleur dans la colorisation d'image. La première image est l'image en niveaux de gris à coloriser en utilisant des indices de couleur. La seconde image est le résultat de la méthode (Zhang et al., 2017) présentant un débordement de couleur évident.

VII. **Éviter la complexité dans les modèles d'apprentissage profond.** Principalement, les travaux récents qui associent l'interaction de l'utilisateur et les réseaux de neurones profonds se concentrent sur l'augmentation de la taille des modèles et l'incorporation de nombreuses fonctions de perte supplémentaires dans leurs modèles, rendant chaque nouvelle approche plus complexe en termes de calculs et d'explications théoriques.

Par conséquent, pour relever le défi VII., nous avons établi certaines contraintes comme suit :

- Nous visons à utiliser une architecture de réseau génératif simple et des fonctions de perte simples afin de faciliter l'apprentissage, de réduire le nombre de paramètres appris par le réseau et de mieux analyser l'impact des résultats.

Les contraintes ci-dessus aident la phase d'apprentissage, car les réseaux simples ont moins de paramètres et sont moins sujets au sur-apprentissage. Cela les rend également plus faciles à entraîner.Enfin, cela améliore la compréhension et l'analyse des méthodes, car les réseaux simples peuvent aider à isoler les composants clés responsables de leur performance.

Plan de la thèse et contributions

Dans cette thèse, nous nous concentrons sur l'application de la colorisation d'images avec interaction utilisateur, et tout au long de ce travail, plusieurs publications scientifiques ont été publiées avec leurs contributions correspondantes. Pour chacun des chapitres, nous décrivons ci-dessous l'organisation correspondante et ses principales contributions, ainsi que le défi qu'il cible :

Chapitre 2. Ce chapitre commence par une brève introduction aux différents types de colorisation automatique. Ensuite, il définit formellement le problème de la colorisation automatique. Les trois sections suivantes passent en revue la littérature sur

les trois principaux types de méthodes de colorisation automatique : (1) colorisation avec *scribbles*, (2) colorisation avec images de référence, et (3) colorisation par apprentissage sans interaction. Enfin, nous examinons l'état de l'art sur les métriques d'évaluation utilisées pour comparer différentes approches de colorisation automatique.

Contributions :

✓ Nous présentons un aperçu exhaustif des méthodes, métriques d'évaluation et fonction de pertes utilisées dans les applications de colorisation d'images.

Chapitre 3. Dans ce chapitre, nous commençons par proposer une architecture simple de réseau génératif profond pour la colorisation automatique par apprentissage et définissons l'ensemble de données utilisé pour les expériences. L'influence des espaces de couleur et des fonctions de perte sur les résultats de colorisation est ensuite étudiée. Enfin, nous discutons de l'importance de concevoir soigneusement l'architecture et les protocoles d'évaluation pour les algorithmes de colorisation.

Contributions :

✓ Nous étudions l'impact de différents espaces de couleur et de plusieurs fonctions de perte sur la performance d'un processus de colorisation basé sur l'apprentissage profond. Nous répondons spécifiquement à la question est-il crucial de choisir le bon espace de couleur ou certaines fonctions de perte. (Défi VII.)

Chapitre 4. Ce chapitre commence par introduire le mécanisme d'attention et la segmentation en superpixels, appliqués à la tâche d'édition d'image. Ensuite, il présente notre proposition appelée mécanisme de *super-attention*. Il s'agit d'un mécanisme d'attention basé sur une stratégie d'agrégation de superpixels. Enfin, notre méthode est évaluée sur l'application de transfert de couleur.

Contributions :

✓ Nous proposons une nouvelle méthode pour faire correspondre des cartes de caractéristiques de haute résolution provenant de CNNs en utilisant des mécanismes d'attention. Cette méthode permet de résoudre en partie le problème de mise à l'échelle quadratique de l'attention et aide à améliorer le problème de débordement de couleur. Pour illustrer l'intérêt de ces nouveaux blocs méthodologiques, nous nous appuyons sur (Giraud et al., 2017) et incluons ces similarités dans un cadre du transfert de couleur. (Défi V. VI.)

Chapitre 5. Dans ce chapitre, nous commençons par explorer comment les mécanismes d'attention sont utilisés dans la colorisation d'images basée sur des exemples. Nous introduisons ensuite notre architecture générale pour la colorisation d'images par images de référence, qui utilise un bloc de super-attention spécifiquement pour la tâche de colorisation. Une nouvelle approche pour générer un ensemble de données pour entrainer le réseau de colorisation est également explorée. Enfin, nous comparons notre approche aux méthodes récentes sur diverses métriques quantitatives, et démontrons que notre méthode produit des résultats visuellement attrayants.

Contributions :

- ✓ Nous proposons un cadre d'apprentissage profond pour la colorisation d'images basée sur des exemples, qui s'appuie sur des couches d'attention pour capturer des correspondances robustes entre des caractéristiques profondes de haute résolution provenant de paires d'images. Notre bloc appelé super-attention peut apprendre à transférer des caractéristiques de couleur sémantiquement liées à une image de référence à différentes échelles d'un réseau profond. (Défis I., II., III., V., VI., VII.)
- ✓ Nous proposons une stratégie pour choisir des paires d'images cibles/références pertinentes pour former une approche de colorisation d'images basée sur des exemples. (Défi IV.)

Chapitre 6. Ce chapitre commence par explorer l'utilisation de la segmentation dans la tâche de colorisation d'image. Nous introduisons ensuite notre proposition appelée *super-attention masquée*, qui permet aux utilisateurs d'interagir avec des objets spécifiques dans le processus de colorisation. La fin du chapitre présente une évaluation complète de l'approche proposée, incluant à la fois des images complètes et d'objet.

Contributions :

- ✓ Nous développons une nouvelle méthode d'apprentissage profond de bout en bout pour la colorisation basée sur des exemples. Elle prend en entrée des masques d'objets fournis par l'utilisateur. Cette approche vise à guider la colorisation sur des objets spécifiques et significatifs plutôt que sur une image de référence complète. (Défis I., II., III., V., VI., VII.)
- ✓ Nous menons une évaluation approfondie de notre approche à la fois au niveau de l'image entière et au niveau des objets, et la comparons à l'état de l'art.

Chapitre 7. Ce chapitre commence par introduire la tâche de colorisation de dessin au trait et sa littérature. Il explique ensuite les modèles de diffusion, à la fois inconditionnels et conditionnels, pour la génération d'images. Ensuite, une méthode pour conditionner un modèle de diffusion pour la colorisation de dessins au trait est introduite. Enfin, la méthode proposée est comparée aux approches de la littérature en utilisant à la fois des métriques quantitatives et qualitatives.

Contributions :

- ✓ Nous proposons une nouvelle approche interactive pour la colorisation d'images au trait utilisant des modèles probabilistes de diffusion conditionnels. Dans notre approche, l'utilisateur fournit des traits de couleur initiaux pour guider la colorisation. (Défis I., II., III., IV., V.)

Chapitre 8. Ce chapitre résume les contributions de cette thèse et discute des limitations de notre travail, ainsi que des orientations pour des perspectives de recherche.

Publications

Durant la préparation de cette thèse, un certain nombre d'articles ont été publiés.

Articles de journaux

[Carrillo *et al.* 2023] **H. Carrillo,** M. Clément, A. Bugeau. "Exemplar-based image colorization using object-guided attention" *Preprint submitted to: International Journal of Computer Vision (IJCV)*, 2023. (Chapter 6)

Conférences internationales

[Carrillo *et al.* 2023] **H. Carrillo,** M. Clément, A. Bugeau, E. Simo-Serra. "Diffusart: Enhancing Line Art Colorization with Conditional Diffusion Models." *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. (Chapter 7)

[Carrillo *et al.* 2022] **H. Carrillo,** M. Clément, A. Bugeau. "Super-attention for exemplar-based image colorization." *Asian Conference on Computer Vision (ACCV)*, 2022. (Chapter 5)

[Carrillo *et al.* 2022] **H. Carrillo,** M. Clément, A. Bugeau. "Non-local matching of superpixel-based deep features for color transfer." *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2022. (Chapter 4)

Chapitres de livres

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Analysis of Different Losses for Deep Learning Image Colorization." *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022. (Chapter 3)

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Influence of color spaces for deep learning image colorization". *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022. (Chapter 3)

Conférences nationales

[Carrillo *et al.* 2021] **H. Carrillo,** M. Clément, A. Bugeau. "Superpixel-based matching of high-resolution deep features for color transfer." *Journées francophones des jeunes chercheurs en vision par ordinateur (ORASIS)*, 2021. (Chapter 4)

Chapter 1

Introduction

Table of contents

1.1 General context

Colorization involves assigning plausible colors to grayscale images, aiming to produce visually appealing images while avoiding unwanted artifacts or incorrect colors. This application is used in many fields, including restoration of legacy photos/videos, broadcasting, film post-production, and animation. However, current processes are often time-consuming and tedious, as they highly depend on manual intervention from the artist. For instance, in the post-production industry, the tools available for professionals enable artists to reach high-level quality results but require long human intervention. An example of a professional colorization pipeline is shown in Figure 1.1 where keyframes are manually chosen from the video, each shot of about 75 images. They correspond to new events or objects entering the scene. Subsequently, artists or historians manually link each segmented object to an appropriate color, taking into consideration historical accuracy or visual appeal. This step can take up to one day for complex shots of many objects or close faces. Then, an object tracking algorithm is finally used to propagate the colors of each object to the rest of the images.

Figure 1.1: Colorization process for *MARIA BY CALLAS* - Tom Volf, released on December 13th, 2017. Top row: the original key frame and its colorized version after post-production. Bottom row: each processing step, left to right: object segmentation, manual coloring of objects, and object tracking with color propagation.

Automating the colorization process can significantly improve workflow for artists, but it is challenging due to its inherent ambiguity. This is because several plausible colors can be assigned to the same gray pixel of an image, depending on various factors, such as the same objects with different colors and/or complex structures on the image. The importance of interaction comes from the previous fact that there is no single, objectively correct solution when it comes to colorizing a grayscale image (see Figure 1.2). In addition, artists, restoration professionals, or animators may have specific intentions or creative visions in mind when colorizing an image. These intentions vary widely and may not be easily picked out by fully automated algorithms alone. Therefore, user input becomes essential in guiding the colorization process toward aligning with the user's vision or specific requirements.

Over the last 20 years, several colorization algorithms have been proposed in the literature. They usually are classified into three types that are not mutually exclusive: scribble-based colorization, exemplar-based colorization, and, recently, learning-based colorization methods. The last group of methods uses deep generative models to generate

colorized images based on learned priors, achieving interesting results. However, there is still a long run to have satisfactory results for real-world applications, particularly when the user wants to intervene in the automatic colorization process.

Figure 1.2: Example of image colorization ambiguity issue. From a grayscale image (first column), multiple valid colorizations (next columns) can be generated as, for example, the woman's t-shirt can take different colors, and all three are visually pleasant. Figure taken from (Huang et al., 2022).

From the literature, there are two ways to incorporate user interaction into deep learning models for the colorization task. First, the user provides a reference image that shows the desired colors to transfer, or second, paint color scribbles onto different regions on the grayscale image. In this thesis, we leverage user interaction techniques applied to learning-based image colorization and how this can enhance the quality of colorization results. Drawing inspiration from the broader concepts of machine learning, image processing, and computer vision, we explore how user-provided input hints can be introduced into a deep learning model to improve its accuracy and realism in the colorization process.

Context of the thesis. This thesis is part of the ANR PostProdLEAP project, whose goal was to develop new tools for video archive post-production by leveraging both recent deep learning approaches and patch-based and variational approaches. Frequent artifacts observed with deep learning methods included loss of details, spatial and temporal discontinuities, and color bleeding along edges. These artifacts made state-of-the-art deep learning approaches inappropriate for professional post-production.

This thesis explored and designed deep learning models to address the challenges of automatic image colorization in the PostProdLEAP project. We also learned from valuable discussions with artists and documentalists at Composite Films, a post-production company specialized in film restoration and colorization. These conversations inspired this research work to develop new techniques that seamlessly integrate user interactions with deep learning. Our new techniques could make it easier for artists to get involved in the colorization process but also gave them more control over the results. This streamlined their workflow and reduced the monotony of traditional professional software.

1.2 Problem statement

Image colorization is the task of predicting the color values of each pixel in a grayscale image. This is challenging because the model must learn the complex relationships between the grayscale intensities and the corresponding colors while also considering that there are multiple plausible colorizations (ill-posedness) for a given grayscale image.

Recently, deep learning has been shown to be very effective for image colorization tasks. Deep learning models can learn complex mappings between input and output data, even in the presence of noise and other challenges. One approach to automatic image colorization using deep learning is to formulate the problem as a regression task (Cheng et al., 2015; Larsson et al., 2016). The model is trained on a dataset of grayscale images and their corresponding colorized images. It learns to predict the chrominance values of each pixel in the grayscale image, given the grayscale intensities. Another approach to automatic image colorization using deep learning is formulating the problem as a classification task. The model learns to classify each pixel in the grayscale image into sets of predefined colors (Zhang et al., 2017; Iizuka et al., 2016). These sets of predefined colors usually are color bins manually chosen based on the distribution of colors in the training dataset. In both approaches, the deep learning model is trained to learn a transformation Φ from a grayscale image T_L to a colorized version $\hat{T}$. However, as detailed in Section 1.1, there is ambiguity in selecting a unique color for a gray pixel in the grayscale image. To address this, user prior knowledge is often added to the colorization process to ensure that the final image has desirable properties.

In this manuscript, we are mostly interested in different approaches to incorporate user interactions U in a deep learning framework. In detail, we aim to learn a transformation Φ that maps a grayscale image $T_L \in R^{H\times W\times 1}$ with values between $[0, 255]$ to a RGB colorized image $\hat{T} \in R^{H\times W\times 3}$ with values between $[0, 255]^3$ while incorporating previously mentioned user interaction U.

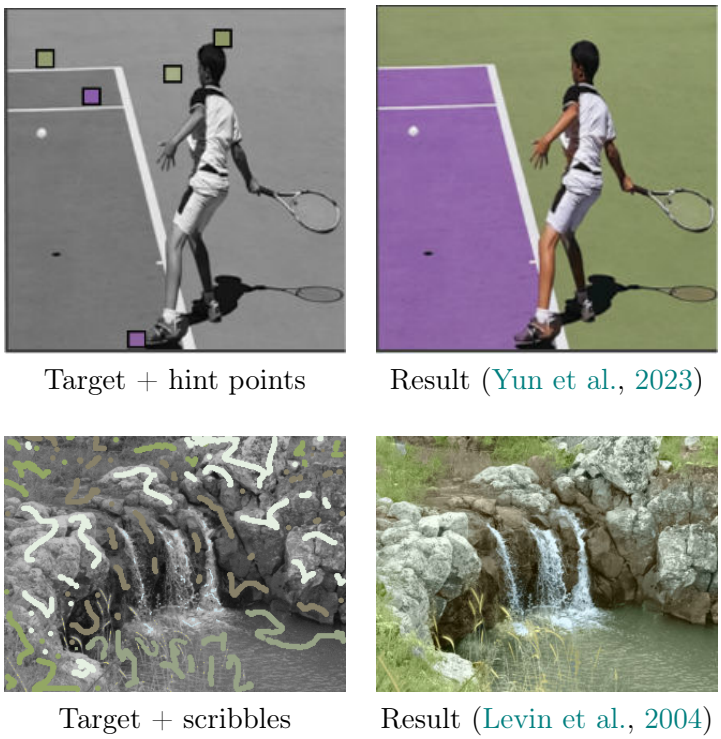
1.3 Challenges and constraints

Deep neural networks have achieved impressive results in automatic image colorization (He et al., 2018; Vitoria et al., 2020; Huang et al., 2022). However, introducing user guidance into deep neural networks for this task is not straightforward. There are some scientific and applicative challenges to overcome, including:

I. **Understanding the user's intent.** It is important to understand what the user wants to achieve with their guidance. For example, do they want to change the overall color of the image, or do they want to adjust the color of specific objects. For the first question, current methods usually rely on guiding the colorization using the full context of a reference image. However, for the second question, users usually rely on numerous color hints, which can be tedious for the task.

II. **Representing the user's guidance.** User guidance comes in various forms, such as reference images, color hints, or scribbles. For example, scribbles are freehand lines or marks that users draw on grayscale images to guide the colorization process. They are a hands-on way to indicate the desired colors, especially in complex regions or boundaries. Color hints, on the other hand, are more discrete suggestions, often represented as dots or small patches. They are less invasive and quicker to apply than scribbles, but they may require the colorization algorithm to make more assumptions about the surrounding areas (Figure 1.3 shows an example of both user interactions). Each of these types of guidance needs a unique way to be understood and used by the network's architecture. Then, it is crucial to strike a balance where the user's

input has a significant impact on the results, yet the network retains its ability to colorize any missing hint regions and rectify any mistakes or irregularities in the given guidance.



Target + hint points | Result (Yun et al., 2023)

Target + scribbles | Result (Levin et al., 2004)

Figure 1.3: Example of colorization methods using user interaction. The first row presents the method (Yun et al., 2023) that leverages user hint points. The second row shows results on a classic method (Levin et al., 2004) that leverages user scribbles.

III. **Incorporating the user's guidance into the model.** The user's guidance needs to be incorporated into the deep neural network in a way that preserves the overall quality of the colorized image. This can be challenging, as the user's guidance may be inconsistent with the training data for the deep neural network. Therefore, ensuring the balance between user color inputs and maintaining the network generalization capacity is a difficult task. For example, a deep neural network is trained to colorize animals, and it receives a grayscale image of a cat; the network might favor typical colors found in the training set, like tabby browns. But if a user inputs color hints like a light orange for the fur, a potential approach would be to blend the user's preferred color while preserving the cat's natural eye and nose colors from the model's learned knowledge.

IV. **Creating datasets while preserving generalization.** Designing a large dataset that mimics user inputs while improving the generalization task in the deep neural network is not straightforward (Zhang et al., 2017; Ci et al., 2018). For example, in scribble-based approaches, there is no formal way to simulate scribble inputs to the network. Another example are exemplar-based colorization methods, where different metrics are used to automatically find target-reference pairs for training.

V. **Handling color ambiguity.** Image colorization is an ambiguous task. This means that there can be many different possible colorizations for the same grayscale image. One factor that contributes to this ambiguity is the texture of the image. This is because many objects in the real world have variations in color while maintaining a similar texture. For example, different types of apples can have similar grayscale texture representations but different colors. Therefore, users can provide more accurate color information than a fully automatic deep learning model can learn on its own. This is especially true for complex regions or boundaries. This is a way user guidance can be very helpful in improving the results of image colorization algorithms.

VI. **Avoiding color bleeding.** A common artifact in the automatic colorization task is color bleeding when colors from one object spread into adjacent ones. For example, Figure 1.4 shows two parts where color bleeding is present; in those two parts, the colorization method mixes the color blue from the sky with the pink float toy. The previous happens when edge information is not considered in the colorization process. Therefore, developing approaches using edge-aware techniques within the colorization process could be a way to handle common artifacts.

Target + color hints Result (Zhang et al., 2017)

Figure 1.4: Example of color bleeding in image colorization. The first image is the grayscale image to be colorized using color hints. The second image results from the method (Zhang et al., 2017) presenting clear color bleeding.

In essence, coupling user guidance into deep neural networks for image colorization requires an approach that tackles the previous six general challenges. In addition to the previous general challenges, we identified other specific issue related to current learning-based interactive colorization proposals.

VII. **Avoiding complexity in deep learning models.** Mainly, recent works that couple user guidance and deep neural networks focus on the augmentation of the learned parameters and incorporating many additional loss terms to their frameworks, making each new approach more complex in terms of computations and theoretical explanations (for instance, should we consider using two or more learning models?. Is the network overfitting the data? or is it beneficial to apply 4 or 5 losses?)

To attack the challenge VII. we set some constraints as follows:

- We aim to use a simple generative network architecture and simple losses in order to facilitate training, reduce the number of parameters learned by the network, and better analyze the impact of the results.

The above constraints help to facilitate training, as simple networks have fewer parameters and are less prone to overfitting. This not only makes them easier to train but also helps in getting a clearer comprehension of our proposed method. Finally, it improves understanding and analysis of the framework, as simple networks can help isolate the key components responsible for their performance.

1.4 Outline of the thesis and contributions

In this thesis, we focus on the practical application of image colorization with user interaction, and throughout this work, several scientific publications have been published with their corresponding contributions. For each of the chapters, we describe the corresponding organization and its main contributions, as well as the challenges that it targets. These are summarized as follows:

Chapter 2. This chapter begins with a brief introduction to the different types of automatic colorization. Next, it formally defines the problem of automatic colorization. The following three sections review the literature on the three main types of automatic colorization methods: (1) colorization with scribbles, (2) colorization with reference images, and (3) colorization by learning without interaction. Finally, we review the state-of-the-art on evaluation metrics used to compare different automatic colorization approaches.

Contributions:

✓ We present an exhaustive overview of state-of-the-art methods, evaluation metrics, and losses used in image colorization applications.

Chapter 3. This chapter starts by proposing a simple deep generative network architecture for automatic colorization by learning and defines the dataset used for the experiments. It then investigates the impact of color spaces and loss functions on the colorization results. Finally, it discusses the importance of carefully designing the architecture and evaluation protocols for colorization algorithms.

Contributions:

✓ We study the impact of different color spaces and several losses on the performance of a deep learning-based colorization process. We specifically answered the question of whether it is crucial to choose the right color space or certain loss functions. (Challenge VII.)

Chapter 4. This chapter begins by introducing the basis of the attention mechanism and the super-pixels algorithm applied to the image editing task. Then, it presents our proposal called *super-attention* mechanism, which is an attention mechanism based on a superpixel-pooling strategy. Finally, it evaluates our method for the problem of color transfer.

Contributions:

- ✓ We propose a new method for matching high-resolution feature maps from CNNs using attention mechanisms. This method solves the quadratic scaling problem of all-to-all attention and helps improve the color bleeding issue. To illustrate the interest of these new methodological blocks, we build upon (Giraud et al., 2017) and include these similarities in a non-local color fusion framework. (Challenges V. VI.)

Chapter 5. This chapter begins by exploring how attention mechanisms are used in exemplar-based image colorization. It then introduces our general architecture for examplar image colorization that makes use of a super-attention block specifically for the colorization task. Next, it explores a novel approach to generate a dataset for training an examplar image colorization network. Finally, it compares our approach to state-of-the-art methods on various quantitative metrics, demonstrating that our method produces visually appealing results.

Contributions:

- ✓ We propose a deep learning framework for exemplar-based image colorization, which relies on attention layers to capture robust correspondences between high-resolution deep features from pairs of images. Our block called super-attention can learn to transfer semantically related color features from a reference image at different scales of a deep network. (Challenges I., II., III., V., VI., VII.)
- ✓ We propose a strategy for choosing relevant target-reference image pairs for training an exemplar-based image colorization approach. (Challenge IV.)

Chapter 6. This chapter starts by exploring the use of segmentation in the image colorization task. it then introduces our proposal called *Masked super-attention*, which allows the users to interact with specific objects in the colorization process. Finally, it presents a comprehensive evaluation of the proposed approach, including both full-level and object-level images.

Contributions:

- ✓ We develop a novel end-to-end deep learning framework for exemplar-based colorization that integrates user-provided object masks. This approach aims to guide the colorization on specific and meaningful objects rather than a full reference image. (Challenges I., II., III., V., VI., VII.)
- ✓ We conduct a complete and comprehensive evaluation of our approach at both the full-image level and object-level images, comparing it to state-of-the-art methods.

Chapter 7. This chapter starts by introducing the line art colorization task and reviewing the existing literature. It then explains diffusion models, both unconditional and conditional, for image generation. Next, it proposes a method for conditioning a diffusion model to colorize line art. Finally, it evaluates the proposed method against state-of-the-art approaches using both quantitative and qualitative metrics.

Contributions:

- ✓ We propose a novel interactive approach for line art colorization using conditional Diffusion Probabilistic Models (DPMs). In our proposed approach, the user provides initial color strokes for colorizing the line art. (Challenges I., II., III., IV., V.)

Chapter 8. This chapter summarizes the contributions of this thesis and discusses the limitations of our work, as well as directions for future research.

1.5 Publications

During the preparation of this thesis, a number of articles were published.

Journal articles

[Carrillo *et al.* 2023] **H. Carrillo,** M. Clément, A. Bugeau. "Exemplar-based image colorization using object-guided attention" *Preprint submitted to: International Journal of Computer Vision (IJCV)*, 2023. (Chapter 6)

International conferences

[Carrillo *et al.* 2023] **H. Carrillo,** M. Clément, A. Bugeau, E. Simo-Serra. "Diffusart: Enhancing Line Art Colorization with Conditional Diffusion Models." *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. (Chapter 7)

[Carrillo *et al.* 2022] **H. Carrillo,** M. Clément, A. Bugeau. "Super-attention for exemplar-based image colorization." *Asian Conference on Computer Vision (ACCV)*, 2022. (Chapter 5)

[Carrillo *et al.* 2022] **H. Carrillo,** M. Clément, A. Bugeau. "Non-local matching of superpixel-based deep features for color transfer." *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2022. (Chapter 4)

Book chapters

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Analysis of Different Losses for Deep Learning Image Colorization." *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022. (Chapter 3)

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Influence of color spaces for deep learning image colorization". *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022. (Chapter 3)

National conferences

[Carrillo *et al.* 2021] **H. Carrillo,** M. Clément, A. Bugeau. "Superpixel-based matching of high-resolution deep features for color transfer." *Journées francophones des jeunes chercheurs en vision par ordinateur (ORASIS)*, 2021. (Chapter 4)

Chapter 2

A review of image colorization

Table of contents

Summary

In this chapter, we present a walkthrough of the state-of-the-art methods, current evaluation metrics, losses, and datasets used in the literature for the image colorization tasks. To provide a clearer picture, we begin by introducing the general problem formulation, as well as the different types of automatic colorization approaches. We present a brief introduction to the different color spaces that have been used for colorization in the literature. Then, we present a literature review on the three main types of automatic colorization methods: colorization with scribbles, colorization with reference images, and colorization by learning. This is in order to get better insights into the strengths and weaknesses of each method. Finally, we review the state-of-the-art of evaluation metrics, datasets, and losses used to train and compare automatic colorization approaches.

Contributions

The main contribution of this chapter is the following:

- We conduct a literature review and classify existing colorization methods regarding the color spaces and losses they use.

Related publications

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Analysis of Different Losses for Deep Learning Image Colorization." *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022.

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Influence of color spaces for deep learning image colorization". *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022.

2.1 Introduction

As detailed in Chapter 1, image colorization is the process of adding color to black and white images. While in the past, people had to manually color photos by hand, today's technology allows us to do this partially automatically using computers. Over the years, several colorization approaches have been proposed in the literature and can be classified into three types: scribble-based colorization, exemplar-based colorization, and learning-based colorization. However, each of these methods presents its drawbacks; for example, scribble-based methods (Levin et al., 2004; Huang et al., 2005; Qu et al., 2006) require the user to manually provide color hints or scribbles in various regions of the image. This process can be time-consuming and may demand a level of artistic skill, making it less convenient for an automated colorization task. For examplar-based (Welsh et al., 2002; Bugeau et al., 2014; Chia et al., 2011), where it can be costly to find reference images that are semantically similar. And finally, learning-based methods (Zhang et al., 2016; Deshpande et al., 2017; Vitoria et al., 2020) strongly depend on the quality of the dataset. In cases where the training data is biased or insufficiently representative of the colorization application, it can lead to inaccuracies, including challenges in generalizing to unfamiliar content or colors. Therefore, there is a growing interest in incorporating semantic cues from a user-provided reference image to reduce dependency on dataset-learned information and generalize to user-preferred colors.

The aim of this chapter is to give a detailed look into the current literature on image colorization. In this chapter, we are going to explore the fundamentals of automatic colorization, introducing the main challenges and the general problem formulation. From there, we will cover the different color spaces used in different methods. In addition, we will survey evaluation metrics, losses, and datasets currently used within the state-of-the-art methods in the image colorization process.

The outline for this chapter is the following, we formally define the problem of automatic colorization in Section 2.2. Then, Section 2.3 presents the different color spaces that have been used for colorization in the literature. The following three sections 2.4, 2.5 and 2.6 introduce the literature review on the three main types of automatic colorization methods: (1) colorization with scribbles, (2) colorization with reference images, and (3) colorization by learning. Finally, Sections 2.7 and 2.8 review the state-of-the-art of evaluation metrics, datasets, and losses used to train and compare automatic colorization approaches.

2.2 Problem formulation

Given a grayscale image T_L, the goal of automatic image colorization is to produce a corresponding color image $\hat{T}$, where each pixel in $\hat{T}$ is represented as a triplet of Red (R), Green (G), and Blue (B) color channels $\hat{T} = (R, G, B)$ with values in range $[0 - 255]$ for each channel. To find these triplets, the idea is to find a mapping function Φ that takes grayscale image T_L as input and produces the color image $\hat{T}$ as output. T_L is a grayscale image that contains only one channel encoding the luminance. Most methods propose to work in a luminance-chrominance space to constrain the colorization task to generate the chrominance channels based on a given luminance channel. This constraint is advantageous for several reasons. First, the dimensionality reduction as the problem becomes retrieving two chrominance channels instead of the triplet RGB. Second, the

Target image | Result (Huang et al., 2022) | Expected colors

Figure 2.1: Example of failure case. The first column presents a grayscale image to be colorized. The column shows the colorized result from the method (Huang et al., 2022) without any user interaction. The final image shows the ground-truth expected colors.

brightness of the images is represented in the luminance channel, which is related to human perception, and therefore, keeping the original luminance in the colorized image retains the original image's structure and contours as well as perceived brightness and contrast (perceptual consistency). Finally, luminance-chrominance separation often leads to more realistic and visually pleasing results. It ensures that the derived colors are consistent with the brightness of the original grayscale image. Based on the previous reasons, most of the methods in the literature keep this luminance-chrominance spaces approach and then convert it into the RGB color space.

In this manuscript, we opt to utilize the luminance-chrominance CIELAB color space. This choice comes from its inherent benefits, which were outlined at the end of the previous paragraph. Furthermore, the CIELAB color space has the special feature of perceptual uniformity, as highlighted by (Connolly et Fleiss, 1997). In essence, this means that any change in color value is perceived consistently, regardless of its position in the color space. As a result, color variations appear smooth and inherently natural to the human eye. In this case, instead of directly retrieving a color image $\hat{T}$ in RGB, we predict the two chrominance channels $\hat{T}_{ab}$ by means of a mapping function Φ:

$$\hat{T}_{ab} = \Phi(T_L), \tag{2.1}$$

once the chrominance channel is retrieved, then we can directly find $\hat{T}$ from $\hat{T}_{Lab}$ by a color space transformation from CIELAB to RGB color space (see Section 2.3 for more details).

As discussed in Section 2.1, one of the main challenges in automatic colorization is the ill-posed nature of the problem. To address this challenge, learning-based colorization methods have been proposed. These methods leverage deep learning techniques and large datasets of color images to learn meaningful color mappings. To model the colorization problem, a deep neural network learns the mapping Φ which is a complex function parameterized by the model's weights and biases. Nonetheless, fully leveraging learning-based colorization without interaction is not capable of solving the ill-posed nature of the problem (see Figure 2.1). A solution is to consider the user's prior knowledge to ensure that the final image has the desirable properties of the artist. Therefore, in this work, we aim to learn a transformation Φ that maps a grayscale image T_L to a colorized image $\hat{T}$ while

incorporating user interaction U:

$$\hat{T}_{ab} = \Phi(T_L \,|\, U), \tag{2.2}$$

where user interaction U can be either a reference image R input by the user or user color scribbles S.

2.3 Basis of color spaces

In this section, we present the different color spaces that have been used for colorization in the literature.

Color images are traditionally saved in the RGB color space. A grayscale image contains only one channel that encodes the luminosity (perceived brightness of that object by a human observer) or the luminance (absolute amount of light emitted by an object per unit area). A way to model this luminance Y (MacAdam, 1937), which is close to the human perception of luminance is:

$$Y = 0.299R + 0.587G + 0.114B, \tag{2.3}$$

where R, G and B are, respectively, the amount of light emitted by an object per unit area in the low, medium and high frequency bands that are visible by a human eye. Colorization aims to retrieve color information from a grayscale image. As said in Section 2.2, to do so and to easily constrain the luminance channel, most methods propose to work in a luminance-chrominance space. The problem becomes the retrieval of two chrominance channels given the luminance Y. There exists several luminance-chrominance spaces. Two of them are mostly used for colorization. The first one, YUV (CIE, 1998), historically used for a specific analog encoding of color information in television systems, is the result of the linear transformation:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51498 & -0.10001 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

The reverse conversion from YUV and RGB is simply obtained by inverting the matrix. The other linear space that has been used for colorization is YCbCr.

The CIELAB color space, also referred to as Lab or La*b* or L$\alpha\beta$, defined by the International Commission on Illumination (CIE) in 1976, is also frequently used for colorization. It has been designed such that the distances between colors in this space correspond to the perceptual distances of colors for a human observer. The three channels become uncorrelated. The transformation from RGB to Lab (and the reverse) is non linear. First, it is necessary to convert the RGB values to the CIEXYZ color space:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.769 & 1.7518 & 0.13 \\ 1 & 4.5907 & 0.0601 \\ 0 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Then, the transformation to Lab is given by:

$$\begin{aligned} L &= 116\,[f(Y/Y_n) - 16]\,, \\ a &= 500\,[f(X/X_n) - f(Y/Y_n)]\,, \\ b &= 200\,[f(Y/Y_n) - f(Z/Z_n)]\,, \end{aligned}$$

with

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (\frac{6}{29})^3, \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise.} \end{cases}$$

Where X_n, Y_n and Z_n describe a specified white achromatic reference illuminant. Obviously, the reverse operation from Lab to RGB is also non linear.
Despite RGB or luminance-chrominance color spaces, few methods relying on hue-based spaces have been proposed for colorization. For instance, (Larsson et al., 2016) rely on a hue-chroma-luminance space.

Table 2.1 lists the color spaces used in deep learning colorization methods described in the next subsection. It distinctly appears that the Lab color space is the most widely used. We will further discuss this choice in Section 3.3.

		RGB	YUV	YCbCr	Lab	hue/chroma	Comparison
	(Cheng et al., 2015)		•				
	(Iizuka et al., 2016)				•		•
Using GANs	(Vitoria et al., 2020)				•		
	(Cao et al., 2017)	•	•				•
	(Yoo et al., 2019)	•			•		
	(Antic, 2019)	•	•				
Histogram prediction	(Larsson et al., 2016)				•	•	•
	(Zhang et al., 2016)				•		
	(Mouzon et al., 2019)				•		
User-guided	(Zhang et al., 2017)				•		
	(He et al., 2018)				•		
	(Lu et al., 2020)				•		
	(Blanch et al., 2021)				•		
	(Yin et al., 2021)				•		
	(Huang et al., 2022)	•			•		
	(Chang et al., 2023)				•		
	(Yun et al., 2023)				•		
	(Chapter 5)				•		
	(Chapter 6)				•		
	(Chapter 7)				•		
Diverse	(Deshpande et al., 2017)	•					
	(Guadarrama et al., 2017)			•			
	(Kumar et al., 2021)	•					
Object-aware	(Su et al., 2020)				•		
	(Pucci et al., 2021)				•		
	(Kong et al., 2021)				•		
Survey	(Gu et al., 2019)	•					

Table 2.1: Color spaces used in deep learning methods for image colorization.

In general terms, as can be seen in Table 2.1, most methods work in a luminance-chrominance space, and the loss functions to optimize are in general defined in the same space. Hence, converting from and to RGB to one of these luminance/chrominance spaces is not involved in the backpropagation step. Once the training is performed, at inference time the chrominance values given by the network together with the luminance component are converted back to the RGB color space. However, this operation tends to perform an abrupt value clipping to fit in the RGB cube, hence modifying both the original luminance values and the predicted chrominance values. Three libraries are most commonly used for the conversion step: the color module of scikit-image (Zhang et al., 2016; Larsson et al., 2016; Zhang et al., 2017; Royer et al., 2017), the color space conversions functions of OpenCV (Iizuka et al., 2016; Vitoria et al., 2020) and the differentiable color transformation library from Kornia (Riba et al., 2020).

2.4 Interacting with scribbles

The first category of colorization methods relies on color priors coming from scribbles drawn by the user (see Figure 2.2). These colors are propagated to all pixels by diffusion schemes. The first manual colorization method based on scribbles was proposed by (Levin et al., 2004). It solves an optimization problem to diffuse the chrominances of scribbles with the assumption that chrominances should have small variations where the luminance has small variations. To reduce the number of needed scribbles, (Luan et al., 2007) first use scribbles to segment the image before diffusing the colors. (Yatziv et Sapiro, 2006) propose a simple yet fast method by using geodesic distances to blend the chrominances given by the scribbles. In (Huang et al., 2005), edge information is extracted to reduce color bleeding. (Heu et al., 2009) use pixel priorities to ensure that important areas end up with the right colors. Other propagation schemes include probabilistic distance transform (Lagodzinski et Smolka, 2008), discriminative textural features (Kawulok et al., 2012), structure tensors (Drew et Finlayson, 2011), non local graph regularization (Lézoray et al., 2008), matrix completion (Wang et Zhang, 2012; Yao et James, 2015) or rank minimization (Ling et al., 2015). As often described in the literature, with these manual approaches, the contours are not well preserved. To cope with this issue, in (Ding et al., 2012), scribbles are automatically generated after segmenting the image and the user only needs to provide one color per scribble. However, all manual methods suffer from the following drawback: if the target represents a complex scene, the user interaction becomes very important. On the other hand, these approaches propose a global optimization over the image, thus leading to spatial consistency in the result.

Figure 2.2: Example of scribble-based image colorization taken from (Levin et al., 2004). The user draws color that are successively diffused to neighbor pixels according to come constraints.

2.5 Interacting with reference images

The second category of colorization methods concerns exemplar-based methods which rely on a color reference image as prior. The first exemplar-based colorization method was proposed by (Welsh et al., 2002). It makes the assumption that pixels with similar intensities or similar neighborhood should have similar colors. It extends the texture synthesis approach by (Efros et Leung, 1999): the final color of one pixel is copied from the most

similar pixel in a reference input color image. The similarity between pixels relies on patch-based metrics (see Figure 2.3). This approach has given rise to many extensions in the

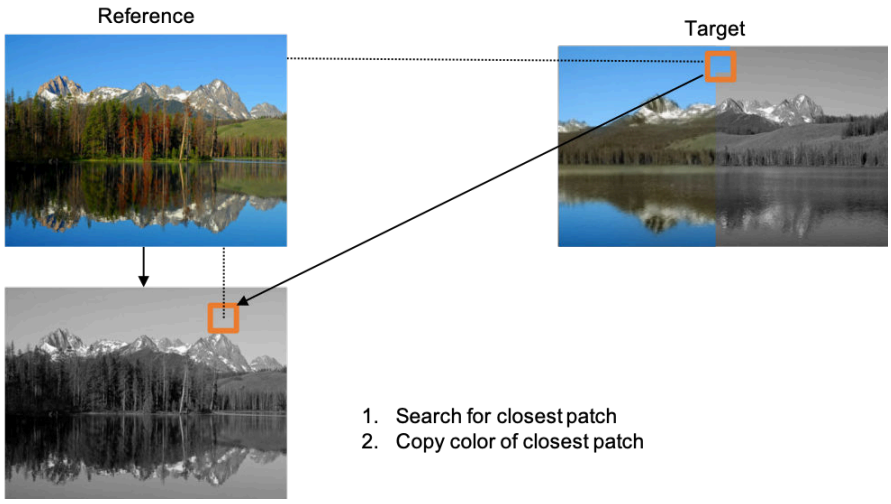


Figure 2.3: Principle of exemplar-based image colorization. Methods in this category have proposed different similar patch search strategies, and techniques to add spatial consistency when copying patch colors.

literature (Di Blasi et Reforgiato, 2003; Liu et Zhang, 2012). In particular, many works have focused on choosing or designing appropriate features for matching pixels (Chia et al., 2011; Gupta et al., 2012; Bugeau et Ta, 2012).

To overcome the spatial consistency and color bleeding problems in automatic methods, several works rely on image segmentation. For instance, (Irony et al., 2005) proposes to determine the best matches between the target pixels and regions in a pre-segmented source image. With these correspondences, micro-scribbles from the source are initialized on the target image and colors are propagated as in (Levin et al., 2004). (Tai et al., 2005) build a probabilistic segmentation of both images where one pixel can belong to many regions. They use it to transfer color between any two regions having similar statistics with an Expectation-Maximization scheme. Other methods use superpixels (Achanta et al., 2012), which are clusters of neighboring pixels with similar attributes, like color or texture. Rather than processing each pixel individually in an image, a superpixel groups several pixels together into a larger and coherent unit. The use of this approach presents advantages, such as dimensionality reduction within the image, as it groups pixels with similar characteristics. It also preserves image structure as it adheres to the object's boundaries within the image, which helps to reduce color bleeding in the image colorization task (for more details, see Section 4.3). This technique has been used in colorization methods such as (Gupta et al., 2012), which extract different features from the superpixels (Ren et Malik, 2003) of the target image and match them with the source ones. The final colors are computed by imposing spatial consistency as in (Levin et al., 2004). (Li et al., 2017b) extract low and high-level features on superpixels of the reference to form a dictionary then used as a dictionary-based sparse reconstruction problem. Sparse representation was previously used for colorization in (Pang et al., 2013) where images are segmented from scribbles. These approaches incorporate local consistency into automatic methods

via segmentation. In (Charpiat et al., 2008), spatial consistency is solved with graph cuts after estimating for each pixel the conditional probability of colors. In (Bugeau et al., 2014; Pierre et al., 2014) each pixel can only take its chrominance (or RGB color) among a reduced set of possible candidates chosen from the reference image. The final color is chosen using a variational formulation. In the same trend, (Fang et al., 2019) proposes a superpixel based variational model. In (Li et al., 2017a), the distribution of intensity deviation for uniform and non-uniform regions is learned and used in a Markov Random Field (MRF) model for improved consistency. Finally, (Li et al., 2019) propose cross-scale local texture matching, which are then fused using global graph-cut optimisation.

A major problem of this family of methods is the high dependency on the reference image. (Chia et al., 2011) therefore propose to rely on several reference images obtained from an Internet search based on semantic information.

2.6 Colorization by learning

Since 2012, deep learning approaches, and in particular Convolutional Neural Networks (CNNs), have become very popular in computer vision and computer graphics communities. Automatic learning-based colorization methods use large datasets to learn how to map each grayscale pixel in an input image to a specific color value.

2.6.1 Without user interactions

The first deep learning-based colorization methods were proposed in (Cheng et al., 2015; Deshpande et al., 2015). In (Cheng et al., 2015), a fully automated system extracts hand-crafted low and high features and feeds them as input to a three-layer fully connected neural network trained with a L2 loss. The network predicts the U and V channels of the YUV luminance-chrominance space. The authors also add an optional clustering stage where the images are divided in different types of scenes, according to the previously extracted semantic features. Then, a different neural network is trained for each of the clusters.

Later on, papers focused more on *end-to-end approaches* (see Figure 2.4).

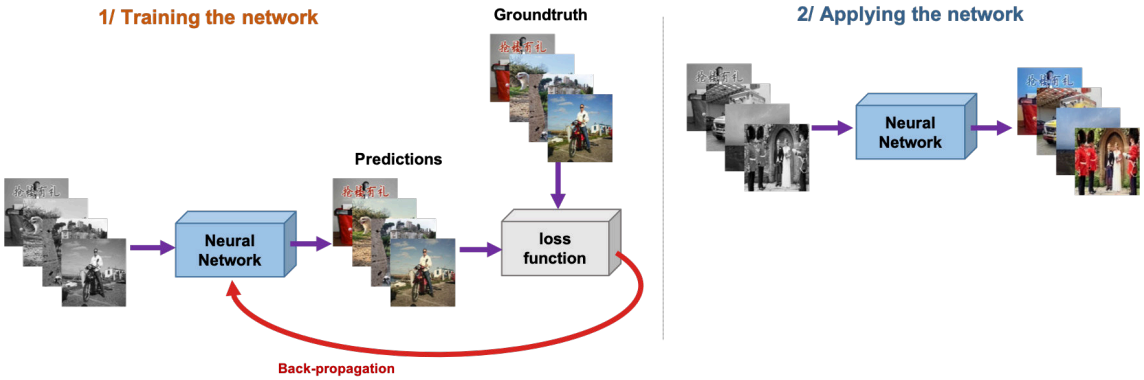


Figure 2.4: Principle of basic end-to-end colorization networks.

For instance, (Wan et al., 2020) proposes to combine neural networks with color propagation, first training a neural network to colorize interest points of extracted superpixels

and then propagating those colors by optimizing an objective function. In older work, (Iizuka et al., 2016) proposed an end-to-end colorization framework based on CNNs to infer the *ab* channels of the CIELab color space, jointly training the network for classification and colorization in a labeled dataset. Other architectures have been used in the literature, for instance, the generative adversarial networks (GANs) (Goodfellow et al., 2014). One of the earliest and most influential GAN-based colorization methods was (Isola et al., 2017), which proposes the so-called image-to-image approach pix2pix, which uses a U-Net generator and a patch GAN discriminator to map an input grayscale image to a colorized output image. This method has been extended and improved in many ways, such as (Nazeri et al., 2018) or (Cao et al., 2017), both proposing a conditional GAN and concatenating a noise channel onto the generator layers to avoid noise attenuation and make the colorization results more diversified. Another GAN-based colorization method is ChromaGAN (Vitoria et al., 2020), which extends (Iizuka et al., 2016) by proposing to learn the semantic image distribution without needing a labeled dataset. ChromaGAN combines a color error loss, a class distribution loss, and an adversarial Wasserstein GAN (WGAN) loss. For more details about the architecture of (Vitoria et al., 2020), see Figure 2.5. DeOldify (Antic, 2019) is another end-to-end image and video colorization method mapping the missing chrominance values to the grayscale input image. It introduced a training procedure, NoGAN, allowing the model to produce high-quality colorizations with minimal flickering across video frames.

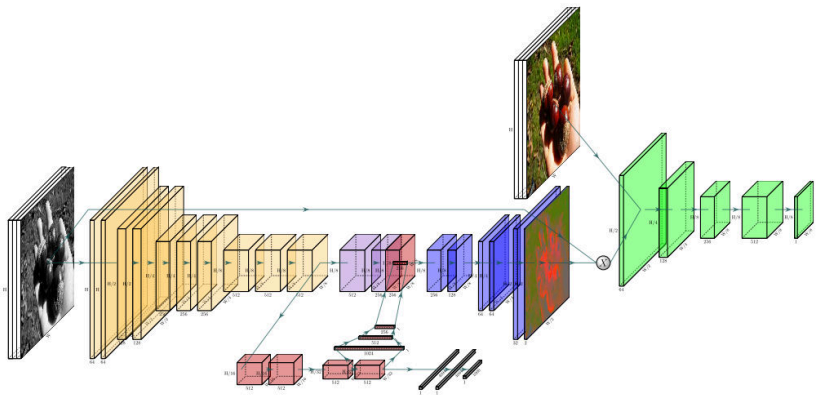


Figure 2.5: Example of the architecture from the method proposed by (Vitoria et al., 2020).

Other approaches focus on predicting distribution instead of reconstructing the full image. This is because regression does not handle multimodal color distributions well (Larsson et al., 2016). (Larsson et al., 2016; Zhang et al., 2016) address this issue by *predicting distributions* over a set of bins, as it was initially done in the exemplar-based method (Charpiat et al., 2008). They, therefore, rely on a discretization of color spaces. Later in (Zhang et al., 2016), the *ab* output space is quantized into bins with grid size ten, and the 313 values which are in gamut are kept. The inference is the annealed mean of the distribution (for architecture details see Figure 2.6). In (Mouzon et al., 2019), the resulting statistical distribution of the colors for each pixel of the image is calculated by a CNN and then introduced into a variational approach to reconstruct the image.

Some methods have been designed to generate diverse colorizations as there is not one

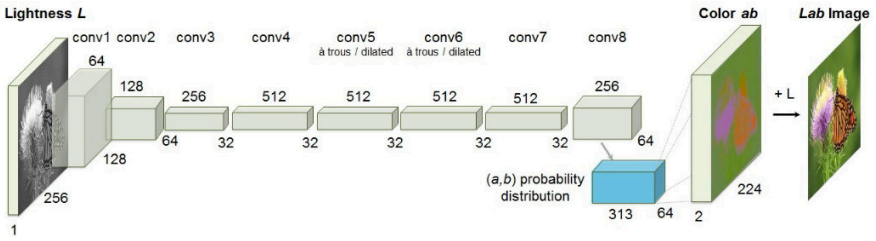


Figure 2.6: Example of the architecture from the method proposed by (Zhang et al., 2016).

unique solution to the colorization problem. This type of method automatically generates a variety of colorized outputs, showing the many possible ways a grayscale image could be colorized. The diversity of the generation comes from the ill-posed nature of the colorization problem, as mentioned in Section 2.2. For instance, a grayscale shirt image could be colorized in green, purple, green, or many other colors, and all would be valid. A diverse unconditional method is (Deshpande et al., 2017), which relies on a variational auto-encoder (VAE) to learn a low-dimensional embedding of color spaces. The mapping from a grayscale input image to color distribution of the latent space is done by learning a mixture density network (MDN). At test time, it is possible to sample the conditional model and use the VAE decoder to generate diverse color images. In their PixColor model, (Guadarrama et al., 2017) first train a conditional PixelCNN (Oord et al., 2016) to generate multiple latent low resolution color images, then train a second CNN to generate the final high resolution images. This embedding is then fed to the autoregressive PixelCNN++ model which predicts a distribution of image chromacity. The colTran model proposed by (Kumar et al., 2021) is based on an axial transformer (Ho et al., 2019) autoregressive model. ColTran includes three networks, all relying on column/row self-attention blocks: the autoregressive model that estimates low resolution coarse colorization, a color upsampler, and a spatial upsampler.

Another way to colorize images using deep learning is by decomposing the scene into objects. This can help to address one of the main drawbacks of most deep learning-based colorization methods, which is color bleeding across different objects. (Su et al., 2020) proposes to colorize a grayscale image in an instance-aware fashion. It trains three separate networks: a first one that performs global colorization, a second one, for instance colorization, and a third one that fuses both colorization networks. In general, after fusing both results, the global colorization will be enhanced. The instances per image are obtained by using a standard pre-trained object detection network, Mask R-CNN (He et al., 2017). (Pucci et al., 2021) propose to improve (Zhang et al., 2016) by using a network that is more aware of image instances, in the spirit of (Su et al., 2020), by combining convolutional and capsule networks (Figure 2.7 show the framework overview). They train from end-to-end a single network, which first generates a per-pixel color distribution followed by a final convolutional layer that recovers the missing chrominance channels as opposed to (Zhang et al., 2016) that computes the annealed mean on the per-pixel color distribution network's output. They train the network by minimizing the cross-entropy between per-pixel color distributions and L2 loss on the chrominance channels. (Kong et al., 2021) proposes to colorize a grayscale image by training a multitask network for colorization and semantic segmentation in an adversarial manner. It trains a U-Net type network with a three-term

cost function: a color regression loss in terms of hue, saturation, and lightness, the cross-entropy on the ground-truth and generated semantic labels, and an adversarial loss for generating more perceptually pleasant images. The main objective of the proposal is to reduce color bleeding across the edges. However, previous deep learning-based colorization methods without interaction reduce colorization time but lack user-specific requirements compared to purely manual colorization.

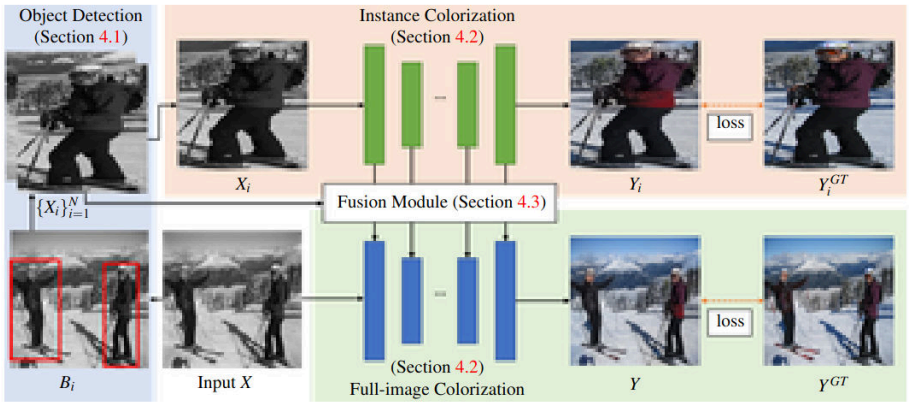


Figure 2.7: Framework overview from the method proposed by (Su et al., 2020).

2.6.2 With user interaction

The previous subsection presents methods that did not involve any user guidance. Nevertheless, unconditional diverse colorization methods can be adjusted to include some user interactivity or feedback. That is, while users are not able to guide the colorization from the start, they can choose from several (diverse) colorized outputs for the most appealing or accurate result. However, in this section, we will focus on methods that users can actively guide the colorization through their input. In the literature, this type of user guidance can appear in the form of reference color images, color hint points or scribbles, text prompts, or even a combination of all the previous ones.

Using reference images. Recently (He et al., 2018) proposed a fully automatic image colorization system that used an end-to-end neural network to calculate the similarity between the reference image and the target image before color transfer (see Figure 2.8 for more details). Their image retrieval algorithm based on the PatchMatch algorithm (Barnes et al., 2009) automatically suggests reference images by analyzing luminance and semantic features to reduce manual work further. The authors of (Xu et al., 2020), inspired by the style transfer techniques, used AdaIN (Huang et Belongie, 2017) to design a colorization framework leveraging characteristics of stylization in feature extracting and blending. It consists of a fast transfer sub-net and a robust colorization sub-net. They initially generate a colorized image by the fast transfer network, and then the images are refined by a second colorization sub-net.

Recently, *attention mechanisms* (Vaswani et al., 2017) have been applied to colorization approaches. These mechanisms where initially proposed for transformer architectures and they identify and focus on specific features of the input data that are most relevant for a

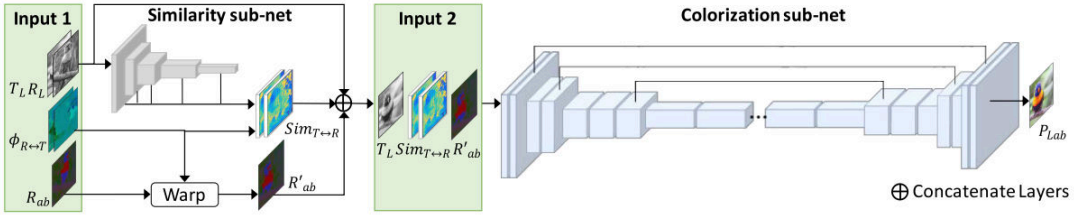


Figure 2.8: Framework overview from the method proposed by (He et al., 2018).

specific task. The base application was natural language processing (NLP), where the attention allowed the model to focus on different parts of a source sequence when translating it into a target sequence. Their model demonstrated a boost in performance, especially on longer sequences, improving on issues from the conventional sequence-to-sequence models that struggled with long sequences due to fixed-length encoding. The success of attention in natural language processing tasks made it a hot topic, which soon found applications in computer vision (Wang et al., 2018b). In this new area, the attention mechanism learns to compute what is known as attention (similarity) maps between image features instead of language sequences. These maps of similarities essentially highlight regions of the images that the model has to "pay attention to". The main idea behind the attention mechanism is the non-local matching operation as proposed originally in (Buades et al., 2005). This operation allows the model to capture long-range dependencies and relationships within the input data rather than just focusing on local information. Lately, the attention mechanism has been applied to the context of colorization, as in (Lu et al., 2020) where they propose an attention-based colorization framework that considers semantic characteristics and a color histogram of a reference image as priors to the final result. Then, (Yin et al., 2021) implement a general attention-based colorization framework that uses the color histogram of the reference image as a prior to eliminate ambiguity in the dataset. Additionally, a sparse loss function is designed to ensure the successful fusion of color information. Improvement over (Yin et al., 2021) was suggested in (Blanch et al., 2021), where they introduced the axial attention mechanism for guiding the transfer of color attributes from the reference image to the target image. Even though these methods provide promising colorization results, they may suffer from the quadratic complexity problem of attention layers and, therefore, can only perform non-local matching at low resolution. In (Cherel et al., 2024), they propose a method for overcoming previous issues by introducing the Patch-based Stochastic Attention Layer (PSAL). In the method, they introduce an optimized attention layer inspired by the stochastic PatchMatch algorithm (Barnes et al., 2009) and patch-based techniques. While the PatchMatch is known for its fast and efficient capability to identify approximate nearest neighbors, it is not differentiable when using a single nearest neighbor (NN). Therefore, to address the issue, they suggest an approach based on patch aggregation. This ensures end-to-end training in deep neural network architectures. However, even though the PSAL method achieves a lighter memory load on the attention computation, they do not use low-level features (just features at the bottleneck as classic attention). Finally, for a deeper dive into the attention mechanism and a possible solution for the previous complexity challenge will be shown in Chapter 4.

Using color hints. The first approach that combines color hints and deep learning is (Zhang et al., 2017). Here, the authors proposed a user-guided image colorization method based on a convolutional neural network. The user can guide the colorization task by providing local or global color hints. For local hints, the user can assign scribble-based color hints to specific pixels or areas of the grayscale image. The proposed CNN model then propagates the color from the scribble across the image. Later, (Yun et al., 2023) proposed iColoriT, which is an interactive image colorization approach that leverages a Vision Transformer (Dosovitskiy et al., 2021) to propagate local color hints to the right region in the image by leveraging the global receptive field of Transformers. The self-attention mechanism of the Transformers enables the proposed framework to selectively colorize relevant regions with only a few local hints. However, a drawback of this method is that it can be sensitive to the quality of the color hints provided by the user. If the color hints are not accurate or representative of the desired colors, the method might struggle to produce coherent colors.

Using text prompts. These methods aim to colorize grayscale images based on a text description of the desired colors. The goal is to produce colorization results that are consistent with the description, even if the description is incomplete or ambiguous. (Manjunatha et al., 2018) proposed the first language-based colorization method. This method uses a neural network to colorize grayscale images by means of feature-wise affine transformations that enable the network to inject the language condition into the image features. Improvement over (Manjunatha et al., 2018) was made by (Bahng et al., 2018). The authors propose a model that can generate several color palettes that link the semantics of the input text and the colorized luminance input image. The model can recognize a single word, phrase, or sentence thanks to the proposed module Palette-and-text. Lately, (Chang et al., 2023) introduced a new transformer-based framework for text-based image colorization (Figure 2.9 presents the overview of the framework) that adaptively learns the correspondence between instance regions and color words. The method learns to associate color words with image regions based on the context of the description and the visual features of the image. However, existing text-based image colorization methods rely on annotated correspondence between object words and color words, which is time-consuming and expensive to create. These methods also have a limited vocabulary, making them unable to currently generate realistic colorization results for complex images or images with colors not seen in training.

Using multi-modal data inputs. (Huang et al., 2022), design the first multi-modal framework for colorization. It supports colorization with multiple user hints, both unconditional and conditional approaches, such as stroke, exemplar, text, and its combinations. It is a two-stage framework. The first stage converts the multi-modal input into a common representation of hint points. Then, in the second stage, a Transformer-based network is to generate the colorized image from the hint points. However, it is computationally expensive due to its large number of parameters and requires a large and varied number of datasets to train in a multi-modal manner.

Table 2.2 summarizes all methods deep learning methods presented in Section 2.6, providing details on their particular inputs (other than the obvious grayscale image), their outputs, their architectures and pre- and post-processing steps. Here, FCONV stands for

	Additional inputs	Network	Network's output	Post-processing
(Cheng et al., 2015)	hand-crafted features	3 layers FC	UV	joint bilateral filtering
(Iizuka et al., 2016)	–	CNNs (local/global)	*ab*	upsampling
(Wan et al., 2020)	superpixels features	FC net	interest points' color	propagation and refinement
Using GANs				
(Vitoria et al., 2020)	–	CNNs (local/global) + PatchGAN	*ab*	upsampling
(Cao et al., 2017)	–	FCONV generator with multi-layer noise + PatchGAN	UV/RGB (diverse)	–
(Antic, 2019)	–	U-Net + self-attention + GAN	RGB	YUV conversion + cat(original Y/UV) + RGB conversion
Histograms Prediction				
(Larsson et al., 2016)	–	VGG-16 + FC layers	distributions	expectation
(Zhang et al., 2016)	–	VGG-styled net	distributions	annealed mean
(Mouzon et al., 2019)	–	(Zhang et al., 2016)	distributions	variational model
User-guided				
(Zhang et al., 2017)	color hints	U-Net	distributions + *ab*	–
(He et al., 2018)	color reference	similarity sub-net + U-Net	bidirectional similarity maps	–
(Yin et al., 2021)	color reference	Cross-attention + U-Net	Histograms + *ab*	–
(Blanch et al., 2021)	color reference	Axial attention + U-Net	*ab*	–
(Huang et al., 2022)	text + color reference + color hints	transformer + CLIP	RGB	–
(Chang et al., 2023)	text	transformer	*ab*	–
(Yun et al., 2023)	color hints	Transformer + local stabilizing layer	*ab*	–
(Chapter 5)	Color reference	Super-attention + Unet	*ab*	–
(Chapter 6)	Color reference + mask	Super-attention + Unet	*ab*	–
(Chapter 7)	Color hints	Diffusion models	*RGB*	–
Diverse colorization and autoregressive models				
(Deshpande et al., 2017)	–	cVAE + MDN	diverse colorization	–
(Guadarrama et al., 2017)	–	PixelCNN + CNN	diverse colorization	–
(Kumar et al., 2021)	–	axial transformer + self-attention blocks	diverse colorization	–
Object-aware				
(Su et al., 2020)	object bounding boxes	U-Net (global/instance)	*ab*	–
(Pucci et al., 2021)	–	CNN + capsule net	*ab*	–
Survey				
(Gu et al., 2019)	–	U-Net	RGB	–

Table 2.2: Short description of deep networks for image colorization, their input other than grayscale image, output.

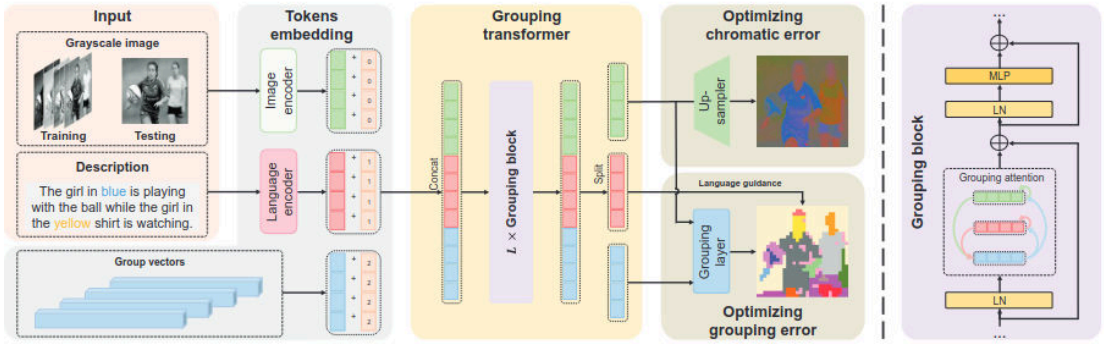


Figure 2.9: Framework overview from the method proposed by (Chang et al., 2023).

fully convolutional, FC for fully connected, and U-Net for a U-Net-like network. This summary table is only provided for deep learning methods since, in this manuscript, we focus mainly on deep learningstrategies.

2.7 Datasets and evaluation metrics

2.7.1 Datasets used in literature

To train and test the deep learning methods presented in the previous Section 2.6, different datasets have been used. Before deep into the explanation of each dataset, Table 2.3 presents a brief summary of datasets commonly used in colorization methods. These datasets contain from one thousand (DIV2K (Agustsson et Timofte, 2017)) to millions of images (ImageNet (Deng et al., 2009)). Image dimensions also vary a lot, from 32×32 in CIFAR-10 (Krizhevsky et al., 2009) to 2K resolution in DIV2K. Other differences concern the content of the images itself. Some datasets are very specific to a type of image: faces (bedrooms (LSUN (Yu et al., 2015)). Other present various scenes as Places (Zhou et al., 2017) with 205 scene categories, COCO (Lin et al., 2014) with 80 object categories and 91 stuff categories, and SUN (Xiao et al., 2010) with 899 scene categories. Figure 2.10 shows examples of images found in some of the detailed datasets. For other types of images, for example, works such as (Jin et al., 2021; Xu et al., 2023), train used legacy photos for training; however, such datasets are not currently open-source. Table 2.4 summarizes the use of these datasets in colorization methods.

Dataset Name	Image Resolution	Content Description	Training	Testing
DIV2K	Up to 2K	Diverse images	800	100
ImageNet	469 x 387 in avg	1k object per categories	1.2M	50k
CIFAR-10	32 x 32	10 object categories	50k	10k
LSUN/bedrooms	256 x 256	Images on bedrooms	3M	300
Places	256 x 256	205 scene categories	2.5M	10k
COCO	640×480 in avg	80 object and 91 stuff categories	118k	5k
SUN	Varies	899 scene categories	76k	11k

Table 2.3: Datasets used in colorization methods.

When it comes to automatic image colorization, using previous datasets is straightforward: we simply create grayscale-color pairs from the same images. However, for interac-

	SUN	ImageNet	COCO	CIFAR-10	DIV2K	Pascal VOC	Places	LSUN bedroom or church	testing on historic BW photo	Others	Remark / Other
(Cheng et al., 2015)	•										
(Iizuka et al., 2016)							•		•		
Using GANs											
(Vitoria et al., 2020)		•							•		
(Nazeri et al., 2018)				•			•				
(Cao et al., 2017)								•			
(Yoo et al., 2019)											Yumi, Monster, etc.
(Antic, 2019)		•									training on 1-3% of ImageNet images
Histograms prediction											
(Larsson et al., 2016)	•	•									
(Zhang et al., 2016)		•				•					training on 1.3M ImageNet images
User-guided											
(Zhang et al., 2017)		•									
(He et al., 2018)		•									training on 700k ImageNet image on 7 categories
(Yin et al., 2021)		•									
(Blanch et al., 2021)		•	•								training on 225k ImageNet in 750 categories
(Huang et al., 2022)		•									
(Chang et al., 2023)			•								59K training and 2.4K validation images
(Yun et al., 2023)		•									
Chapter 5			•								
Chapter 6			•								
Chapter 7										•	Danbooru dataset
Diverse											
(Deshpande et al., 2017)		•						•			LFW
(Guadarrama et al., 2017)		•									
(Royer et al., 2017)		•		•							
(Kumar et al., 2021)		•									
Object-aware											
(Su et al., 2020)		•	•				•				
(Pucci et al., 2021)		•	•				•				
(Kong et al., 2021)						•					
Survey											
(Gu et al., 2019)					•						
(Anwar et al., 2020)											own Natural-Color Dataset

Table 2.4: Datasets used in the literature for colorization by learning.

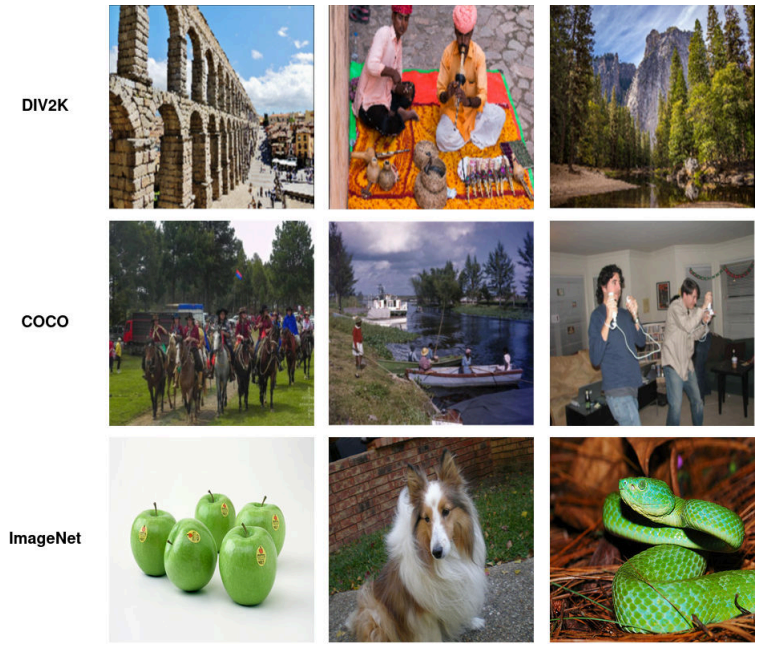


Figure 2.10: Example of images found in datasets: DIV2K (Agustsson et Timofte, 2017), COCO (Lin et al., 2014) and ImageNet (Deng et al., 2009).

tive colorization techniques, like exemplar-based colorization, the process is more complex. These methods often compute image similarities within the dataset to find top-k related reference images. The choice of the similarity metric is still a case of study within this task (refer to Chapter 5). For scribble-based approaches, we typically create synthetic color traces using basic line forms (see Chapter 7). Nowadays, there is no established strategy for managing datasets in interactive colorization, which makes interpreting user inputs in deep learning networks a challenge (see Section 1.3).

2.7.2 Evaluation metrics used in literature

Over the last twenty years, the field of image colorization has evolved, and the need for robust evaluation metrics to quantitatively assess the quality of the colorization has become increasingly important. Therefore, we are going to present the most used evaluation metrics in colorization methods. But first, a color image is assumed to be defined in a discrete setting. Here, Ω denotes a discrete domain defined by an $H \times W$ pixel grid, where $H, W \in \mathbb{N}$. The function T is established on this discrete Ω with values in $\mathbb{R}^C$ where C is the number of channels. Typically, T is represented as a real-valued matrix of dimensions $H \times W \times C$, indicating the image's values. The number of channels C can be set to 3 if T signifies a color image or 2 if the objective is to reconstruct the two chrominance channels. In the latter scenario, the input grayscale image remains unchanged during the colorization process.

Peak signal-to-noise ratio (PSNR): measures the ratio between the maximum value of a color target image $T : \Omega \to \mathbb{R}^C$ and the mean square error (MSE) between $\hat{T}$

and a colorized image $\hat{T} : \Omega \to \mathbb{R}^C$. That is,

$$\begin{aligned}\mathrm{PSNR}(\mathrm{T}, \hat{\mathrm{T}}) = 20 \log_{10}(\max T) \\ - 10 \log_{10} \left(\frac{1}{HWC} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{C} (T(ijk) - \hat{T}(ijk))^2 \right),\end{aligned} \tag{2.4}$$

where $C = 3$ when working in the RGB color space and $C = 2$ in any luminance-chrominance color space as YUV, Lab and YCbCr. The PSNR score is considered as a reconstruction measure tending to favor methods that will output results as close as possible to the ground-truth image in terms of the MSE.

Structural Similarity Index (SSIM): intends to measure the perceived change in structural information between two images. It combines three measures to compare images color (l), contrast (c) and structure (s):

$$\mathrm{SSIM}(\mathrm{T}, \hat{\mathrm{T}}) = \mathrm{l}(\mathrm{T}, \hat{\mathrm{T}})\mathrm{c}(\mathrm{T}, \hat{\mathrm{T}})\mathrm{s}(\mathrm{T}, \hat{\mathrm{T}}) = \frac{(2\mu_{\mathrm{T}}\mu_{\hat{\mathrm{T}}}) + \mathrm{c}_1}{\mu_{\mathrm{T}}^2 + \mu_{\hat{\mathrm{T}}}^2 + \mathrm{c}_1} \frac{(2\sigma_{\mathrm{T}}\sigma_{\hat{\mathrm{T}}} + \mathrm{c}_2)}{\sigma_{\mathrm{T}}^2 + \sigma_{\hat{\mathrm{T}}}^2 + \mathrm{c}_2} \frac{(\sigma_{\mathrm{T}\hat{\mathrm{T}}} + \mathrm{c}_3)}{\sigma_{\mathrm{T}}\sigma_{\hat{\mathrm{T}}} + \mathrm{c}_3} \tag{2.5}$$

where μ_T (resp. σ_T) is the mean value (resp. the variance) of image T values and $\sigma_{T\hat{T}}$ the covariance of T and $\hat{T}$. c_1, c_2, c_3 are regularization constants to avoid division by zero.

Fréchet Inception Distance (FID): (Heusel et al., 2017) is a quantitative measure used to evaluate the quality of the outputs' generative model and aims at approximating human perceptual evaluation. It is based on the Fréchet distance (Dowson et Landau, 1982), which measures the distance between two multivariate Gaussian distributions. FID is computed between the feature-wise mean and covariance matrices (μ_T, Σ_T) of the features extracted from an Inception v3 neural network applied to the ground-truth images and those of the generated images $(\mu_{\hat{T}}, \Sigma_{\hat{T}})$:

$$\mathrm{FID}((\mu_{\mathrm{T}}, \Sigma_{\mathrm{T}}), (\mu_{\hat{\mathrm{T}}}, \Sigma_{\hat{\mathrm{T}}})) = \|\mu_{\mathrm{T}} - \mu_{\hat{\mathrm{T}}}\|_2^2 + \mathrm{Tr}(\Sigma_{\mathrm{T}} + \Sigma_{\hat{\mathrm{T}}} - 2\Sigma_{\mathrm{T}}\Sigma_{\hat{\mathrm{T}}})^{1/2}. \tag{2.6}$$

Fréchet Inception Distance infinity ($\mathbf{FID}_\infty$): In (Chong et Forsyth, 2020) show that the bias in the FID metrics depends on the particular model being evaluated, so a specific model may get a better score than another simply because the bias term is smaller. The number of samples heavily influences this effect. More precisely, FID is linear to $1/N$, where N is the number of generated samples. In (Chong et Forsyth, 2020), they propose a called FID infinity, this evaluation metric extrapolates FID scores to obtain an effectively bias-free estimate of scores computed with an infinite number of samples. Their method involves randomly sampling images from a generated dataset of size N in k intervals, each containing N_{itv} images. They calculate a FID score for each of these intervals and perform linear regression on these k data points to determine the bias-free FID metric, denoted as $\mathrm{FID}_{\mathrm{N_{itv}}}$. This metric is particularly useful for comparing our test set at the object level in Chapter 6, especially as our current dataset split comprises only $N = 1k$ images and is susceptible to this bias issue. In detail, we let $k = 15$ as the default value in their metric. In addition, we choose to calculate FID_{300} and FID_{600} as they are sufficient to know the true tendency of the metric. Finally, to ensure robustness and reliability in our results, since the metric relies on randomly sampled intervals from the test set, we evaluate FID_{300} and FID_{600} ten times. The final results are obtained by calculating the average and standard deviation across these ten evaluations.

Learned Perceptual Image Patch Similarity (LPIPS): (Zhang et al., 2018b). This metric measures the perceptual similarity between two images. It is based on the idea that the human visual system perceives images as a collection of overlapping patches and that the similarity between two images can be measured by comparing the perceptual similarity of their corresponding patches. It computes a weighted L2 distance between deep features of a pair of images T and $\hat{T}$:

$$\text{LPIPS}(T, \hat{T}) = \sum_{l} \frac{l}{H_l W_l} \|\omega_l \odot (\Phi_l(T)_{ij} - \Phi_l(\hat{T})_{ij})\|_2^2, \tag{2.7}$$

where H_l (resp. W_l) is the height (resp. the width) of feature map Φ_l at layer l and ω_l are weights for each features. Note that features are unit-normalized in the channel dimension.

Learned perceptual image patch similarity w.r.t reference (LPIPS_R) (Mechrez et al., 2018). This metric, also known as contextual loss, measures the perceptual similarity between non-aligned images. This means that this metric can evaluate the semantic content even if the image appearance of the image is altered. In this case, the predicted images T and the reference image $\hat{T}$:

$$\mathcal{L}_{\text{CX}}(T, \hat{T}) = -\log\|\|\Phi_l(T) - \Phi_l(\hat{T})\|_1 \tag{2.8}$$

Where Φ is the VGG19 and l represents the layers.

Histogram intersection similarity (HIS): (Isola et al., 2017). This metric evaluates the similarity of the global color distributions of the two images. However, this metric becomes contradictory if the ground-truth and reference have different color distributions. In other words, a good histogram intersection similarity (HIS) score would lead to poor scores in terms of structural similarity (SSIM), learned perceptual image patch similarity (LPIPS), and the Fréchet Inception Distance (FID). As a consequence, we consider the reference image as color guidance to our network in generating a more plausible and realistic colorization. Thus, we regard the HIS score between the ground-truth target images and the reference images as the optimal score in this context, representing what would be achieved with perfect predictions. Then the equation use for calculating ΔHIS:

$$\Delta HIS = |\mathcal{T}_{\text{hist}}(T_H, R_H) - \mathcal{T}_{\text{hist}}(\hat{T}_H, R_H)|, \tag{2.9}$$

where T_H, $\hat{T}_H$, and R_H represent the chrominance histogram calculated in the ab space for the target ground-truth, predicted image, and reference image, respectively and $\mathcal{T}_{\text{hist}}$ refers to histogram intersection metric (Puzicha et al., 1997), that is defined as the symmetric $\mathcal{X}^2$ distance and it is calculated as follows:

$$\mathcal{T}_{\text{hist}}(T_H, R_H) = 2\sum_{q=1}^{Q} \frac{(T_H(q) - R_H(q))^2}{T_H(q) + R_H(q) + \epsilon}, \tag{2.10}$$

where ϵ prevents division by zero and q represents the histogram bins.

Table 2.5 summarizes the quantitative evaluation metrics more generally used in the literature on image colorization.

	Quantitative								User Study		
	L1/ MAE	**L2/ MSE**	**PSNR**	**SSIM**	**LPIPS**	**FID**	**HIS**	**Other**	AMT Fooling Rate	Naturalness	Other
(Cheng et al., 2015)				•							
(Iizuka et al., 2016)								•		•	
Using GANS											
(Vitoria et al., 2020)			•							•	
(Nazeri et al., 2018)	•							•			
(Cao et al., 2017)		•	•								•
Histograms Prediction											
(Larsson et al., 2016)		•	•					•			
(Zhang et al., 2016)								•	•		
User-guided											
(Zhang et al., 2017)			•						•		
(He et al., 2018)			•					•	•		
(Bahng et al., 2018)								•			
(Yin et al., 2021)							•	•			•
(Blanch et al., 2021)				•			•	•		•	
(Huang et al., 2022)					•	•		•		•	
(Chang et al., 2023)			•	•	•					•	
(Yun et al., 2023)			•		•						
(Chapter 5)				•	•		•				
(Chapter 7)				•	•	•					
(Chapter 6)				•	•	•	•	•			
Diverse											
(Deshpande et al., 2017)		•									•
(Guadarrama et al., 2017)				•					•		
(Royer et al., 2017)											•
(Kumar et al., 2021)						•			•		
Object-aware											
(Su et al., 2020)			•	•	•						
(Pucci et al., 2021)			•		•						
(Kong et al., 2021)			•	•				•			
Survey											
(Gu et al., 2019)			•	•							•

Table 2.5: Evaluation metrics used by deep learning methods for image colorization.

2.8 Losses functions for colorization by learning

Image colorization aims to hallucinate the missing color information of a given grayscale image by, as in the case of learning-based methods, directly learning a mapping from the grayscale to the color information by minimizing a chosen objective function. The objective function favors the desired properties the estimated colorization should satisfy. Due to the ill-posed nature of the problem, in most cases, one does not aim to recover the actual ground-truth color –that is, the real color of the actual scene captured in the grayscale image–, but rather to produce a plausible colorization for a human observer. Accordingly, choosing the right way to train such networks is not trivial. The network could end up penalizing a good solution far away from the ground-truth data or estimating an average of all possible correct solutions. Alternatively, instead of directly learning the per-pixel chrominance information, some methods learn a per-pixel color distribution to, afterward, sample from it the color at each pixel. In principle, this could encourage the mapping to be one-to-many, which can be desirable. However, how to properly capitalize and train

such networks to account for the different possible solutions having both, geometric and semantic meanings remains an open problem.

In the following, the different losses used in the literature of image colorization are described and related to some representative works that capitalize on them. Table 2.6 summarizes it.

MSE or L2 loss . This loss is commonly used in image colorization (Cheng et al., 2015; Zhang et al., 2016; Vitoria et al., 2020; Huang et al., 2022) (see also Table 2.6). It measures the squared difference between the predicted color image $\hat{T}$ and the ground-truth color image T. The loss is computed as follows:

$$\text{MSE}(T, \hat{T}) = \frac{1}{HWC} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{C} (T_{ijk} - \hat{T}_{ijk})^2, \tag{2.11}$$

where H and W are the height and width of the image, and C is the number of color channels. For T and $\hat{T}$ are color images (usually the predicted and the ground-truth data) $C = 3$, or $C = 2$ in the case that T and $\hat{T}$ are chrominance images. Although while the training with this loss can lead to a more stable solution, it is not robust to outliers in the data and penalizes large errors while being more tolerant of small errors.

MAE or L1 loss. Also known as the Manhattan distance, it is a loss function that measures the absolute difference between the predicted color image $\hat{T}$ and the ground-truth color image T, and it is defined as follows,

$$\text{MAE}(T, \hat{T}) = \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{C} \|T_{ijk} - \hat{T}_{ijk}\|_1. \tag{2.12}$$

Extensions on the same loss have been done with the **Smooth L1** or **Huber loss**. The Huber loss combines the MSE loss and the MAE loss. The idea behind this is to make it robust to high and low frequent outliners. And it is calculated as follows:

$$L_{1_{smooth}} = \begin{cases} 0.5 \left(T - \hat{T}\right)^2, & \text{if } |T - \hat{T}| < 1 \\ |T - \hat{T}| - 0.5, & \text{otherwise.} \end{cases} \tag{2.13}$$

Several works (Su et al., 2020; He et al., 2018; Yin et al., 2021) use MAE, MAE^c or Smooth L1 losses either alone or combined with other losses (cf. Table 2.6).

Previous L_2, L_1 and Smooth L_1 aim to find a solution close to the ground-truth. This can be counterproductive to the idea that image colorization has multiple possible solutions (Johnson et al., 2016). Additionally, both metrics can be poorly related to perceptual quality. Figure 2.11 exemplifies the previous statement, where two images might have small pixel-wise differences but might look perceptually different as a result in the middle column (MSE small and high LPIPS), or on the contrary, they might have large pixel-wise differences but appear similar to the human perception (third column high MSE and low LPIPS). Therefore, aiming at favoring a solution keeping from the ground-truth, not the exact values but more perceptual or style features, the following error losses have been proposed and used for colorization purposes.

Feature Loss. The feature reconstruction loss (Gatys et al., 2016; Johnson et al., 2016) is a perceptual loss that encourages images to have similar feature representations as the

Original image | Average on chrominance channels | Shift on chrominance channels

Figure 2.11: Example of pixel-wise L_2 (MSE) loss against perceptual loss (LPIPS). The first image shows the original color image where the corresponding losses have been calculated. The second column shows the same image with chrominance channels (ab) average. In the third column, it is applied over the original image, a fixed shift of intensities over each of the ab channels.

ones computed by a pretrained network, denoted here by Φ. Let $\Phi_l(T)$ be the activation of the l-th layer of the network Φ when processing the image T; if l is a convolutional layer, then $\Phi_l(u)$ will be a feature map of size $H_l \times W_l \times C_l$. The *feature reconstruction* loss is the normalised squared Euclidean distance between feature representations, that is,

$$\mathcal{L}^l_{\text{feat}}(T, \hat{T}) = \frac{1}{H_l W_l C_l} \left\| \Phi_l(T) - \Phi_l(\hat{T}) \right\|_2^2. \tag{2.14}$$

It penalizes the output reconstructed image when it deviates in feature content from the target.

Aiming to favor more diverse and perceptually plausible colorization results, losses based on *Generative Adversarial Networks* (GANs) (Goodfellow et al., 2014) have been introduced in the colorization literature (Blanch et al., 2021; Yin et al., 2021; Nazeri et al., 2018; Vitoria et al., 2020). GANs are a kind of generative methods where the goal is to learn the probability distribution of the considered dataset by learning to generate new samples as if they where coming from that dataset. In the case of GANs, the learning is done by an adversarial learning strategy. In the following paragraphs we are going to explain some of the most used GANs losses in the literature, however for more details and other GANs losses please refer to the appendix A.1.

Vanilla GAN. The first GAN proposal by (Goodfellow et al., 2014) is based on a game theory scenario between two networks competing one against another. The first network called generator, denoted by G, aims to generate samples of data $\mathcal{P}_g$ as similar as possible to the ones of real data $\mathcal{P}_r$. The second network, called discriminator, aims to classify between real and generated data. To do so, the discriminator, denoted here by D, is trained to maximize the probability of correctly distinguishing between real examples and samples created by the generator. On the other hand, G is trained to fool the discriminator by generating realistic examples. The adversarial loss of the vanilla GAN is defined as:

$$\mathcal{L}_{\text{adv}}(G, D) = \mathbb{E}_{T \sim \mathcal{P}_r}[\log D(T)] + \mathbb{E}_{\hat{T} \sim \mathcal{P}_g}[\log(1 - D(\hat{T}))], \tag{2.15}$$

where T is a real data sample and $\hat{T}$ is a generated data sample. Then, the min-max

adversarial optimization problem is defined as,

$$\min_G \max_D \mathcal{L}_{\text{adv}}(G, D). \tag{2.16}$$

Wasserstein GAN. Although vanilla GANs have achieved good results in many domains, they have some drawbacks like convergence, vanishing gradients and mode collapse problems. Therefore, some modifications from the original GAN have been proposed. For example, the *Wasserstein GAN* (WGAN), proposed by (Arjovsky et al., 2017) in their work they make use of the standard GAN and extend it to use the Wasserstein distance as a metric for measuring the discrepancy between the data distribution and the model distribution. the proposed loss is calculated as follows:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{T \sim \mathcal{P}_r}[D(T)] - \mathbb{E}_{\hat{T} \sim \mathcal{P}_g}[D(\hat{T})]. \tag{2.17}$$

Here, $\max_{D \in \mathcal{D}}$ indicates the maximization over the set of 1-Lipschitz functions $\mathcal{D}$, which act as the critic. $\mathbb{E}_{x \sim \mathcal{P}_r}[D(x)]$ is the expectation of the critic's output over the real data distribution $\mathbb{P}_r$. In general, the critic tries to maximize this function, which corresponds to estimating the Wasserstein distance, while the generator tries to minimize it, effectively reducing the distance between the real and generated distributions.

To enforce the 1-Lipschitz condition, in (Gulrajani et al., 2017) the authors propose a *Gradient Penalty* (GP) term constraining the L2 norm of the gradient while optimizing the original WGAN during training. The resulting loss for the WGAN-GP can be defined as:

$$\min_G \max_D \left(\mathbb{E}_{T \sim \mathcal{P}_r}[D(T)] - \mathbb{E}_{\hat{T} \sim \mathcal{P}_g}[D(\hat{T})] - \lambda \mathbb{E}_{\widehat{T} \sim \widehat{\mathcal{P}}}[(||\nabla_{\widehat{T}} D(\widehat{T})|| - 1)^2] \right) \tag{2.18}$$

The last term in Equation (2.18) provides a tractable approximation to enforce the norm of the gradient of D to be less than 1. In this equation, $\widehat{\mathcal{P}}$ is the distribution that samples uniformly along straight lines between pairs of points sampled from the real data distribution $\mathcal{P}_r$ and the generated data distribution $\mathcal{P}_g$. The authors of (Gulrajani et al., 2017) motivated it by a theoretical result showing that the optimal discriminator D contains straight lines connecting samples in the ground-truth space and samples in the space of generated data. Finally, the authors ensure that this gradient penalty term improves the stability of the training process.

2.9 Conclusions and future works

In this chapter, we have provided a comprehensive overview of the automatic colorization task. We have formally defined the problem of automatic colorization and discussed the different color spaces that have been used for this task. We have reviewed the state-of-the-art in evaluation metrics, datasets, and losses used to train and compare automatic colorization approaches. We have also delved into the three main types of image colorization methods: colorization with scribbles, colorization with reference images, and colorization by learning. Finally, we discussed the challenges of each of the three types of automatic colorization methods, where we can underline the ill-posed nature of the problem and the need to generalize to unfamiliar content or colors.

Despite the challenges, automatic colorization has made significant progress recently. However, there is still room for improvement, such as increasing accuracy in challenging scenarios (e.g., images with complex textures) and handling color ambiguity, which is still an open issue in learning-based methods. While we have explored methods that use user guidance to address previous challenges, they have not fully solved the issues. Therefore, work on handling interaction in deep learning methods has to be done. In addition, another challenge is to develop methods that are less complex, more efficient, and easier to use in real-world scenarios.

To address previous challenges, in Chapter 3, we explore and choose a simple baseline for our deep learning colorization framework. Next, in Chapter 4, we introduce a novel block that can help to avoid color bleeding issues. And finally, for Chapters 5, 6, we focus on incorporating user color cues into a deep learning colorization framework to reduce dependency on dataset-learned information and generalize to user-preferred colors.

		MAE	smooth-L1	MSE	GANs	KL on distrib.	CE on distrib.	KL for class.	CE for class.	neg log-likelihood	Feature loss	LPIPS	HIS	Others
Survey	winner of (Gu et al., 2019)	•		•					•					
Object aware	(Kong et al., 2021)			•	•									
	(Pucci et al., 2021)			•			•							
	(Su et al., 2020)		•											
Diverse	(Kumar et al., 2021)									•				
	(Guadarrama et al., 2017)	•								•				
	(Deshpande et al., 2017)			•						•		•		
User guided	Chapter 7		•	•										
	Chapter 6		•									•		
	Chapter 5											•		
	(Yun et al., 2023)		•	•										•
	(Chang et al., 2023)		•								•			
	(Huang et al., 2022)	•			•		•				•			•
	(Yin et al., 2021)	•			•							•	•	•
	(Blanch et al., 2021)		•		•								•	•
	(Lu et al., 2020)		•		•								•	•
	(He et al., 2018)		•								•			
	(Zhang et al., 2017)		•				•							
Histogram prediction	(Mouzon et al., 2019)			•			•							
	(Zhang et al., 2016)			•			•							
	(Larsson et al., 2016)			•		•								
Using GANs	(Antic, 2019)				•						•			
	(Yoo et al., 2019)		•		•									
	(Nazeri et al., 2018)			•	•									
	(Vitoria et al., 2020)			•	•			•						
	(Iizuka et al., 2016)			•					•					
	(Cheng et al., 2015)			•										

Table 2.6: Losses used to train deep learning methods for image colorization. CE stands for Cross-Entropy and KL for Kullback-Leibler divergence.

Chapter 3

Exploring a baseline encoder-decoder for image colorization

Table of contents

Summary

In this chapter, we examine how various color spaces and loss functions impact the process of automatic image colorization. First, we define the type of generative deep neural network architecture and dataset used for our experiments. Second, we study the influence of color spaces on the results obtained by training a deep neural network on different color space settings. In this section, we specifically answer the question: "Is it crucial to correctly choose the right color space in deep learning-based colorization?". Third, we answer the question: "Does the choice of the objective function plays an important role in the colorization results?". For that, we analyze the impact of the different loss functions on the estimated colorization results. Finally, we discuss the importance of carefully designing the architecture and evaluation protocols for colorization algorithms.

Contributions

The main contributions of this chapter is the following:

- We study the impact of different color spaces and several losses on the performance of a deep learning-based colorization process. We specifically answered the questions of whether it is crucial to choose the right color space or certain loss functions.

Related publications

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Analysis of Different Losses for Deep Learning Image Colorization." *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022.

[Ballester *et al.* 2022] C. Ballester, A. Bugeau, **H. Carrillo,** M. Clément, R. Giraud, L. Raad, P. Vitoria. "Influence of color spaces for deep learning image colorization". *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, 2022.

3.1 Introduction

This chapter explores the impact of different color spaces and loss functions on the performance of a deep learning-based image colorization approach. As discussed in Chapter 2, image colorization aims to hallucinate the missing color information of a given grayscale image by, as in the case of our work, a learning-based method, directly learning a mapping from the grayscale to chromatic information by minimizing a chosen objective function. To analyze the results of these mappings using learning methods, it is necessary to isolate the contribution of each key component of the approach, namely the used color space, the deep learning architecture, and losses.

Deep learning methods have made significant progress in image colorization in recent years, producing pleasant and vivid results (Vitoria et al., 2020; Huang et al., 2022; Yun et al., 2023). However, due to the ill-posed nature of the problem, often, one does not aim to recover the actual ground-truth color but a plausible one, which makes the choice of a loss function non-trivial. Additionally, in most cases, a single loss function can be insufficient to retrieve the colorized information for the image, so often combinations of two or more losses are used (see Table 2.5 from the previous chapter). In addition, as discussed in the previous chapter, current methods primarily used the CIELab color space, where the luminance channel is provided to the network and the chrominance channels are reconstructed (Pucci et al., 2021; Blanch et al., 2021; Chang et al., 2023). However, there is no evidence that using this particular color space leads to better colorization results. In fact, it is possible that converting from the Lab to RGB color space could generate color artifacts in the final result (Pierre et al., 2014). Finally, recent deep learning frameworks for image colorization are more and more complex in terms of parameters, often using two or more models simultaneously (He et al., 2018; Huang et al., 2022), which makes the training process slow and potentially difficult to reproduce. The combination of these three components makes difficult the analysis of specific contributions related to each new method.

Based on our literature analysis (Chapter 2), in this chapter we defined a simple yet effective baseline architecture for analyzing the influence of color spaces and losses. For the comparison, we focus on the most common color spaces in the literature: RGB, YUV, and Lab color spaces. Regarding the losses, we choose functions that favor colorization results in terms of pixel intensity values (reconstruction losses) as well as losses that enforce perceptual coherence (perceptual losses). To the best of our knowledge, there is currently no study about their influence over the results. Finally, again, based on the literature review, we set a uniform training procedure to ensure fair comparisons. Experiments encompass qualitative and quantitative analysis.

The rest of this chapter is organized as follows. Section 3.2 details the framework used to analyze the influence of the different color spaces and losses, including the chosen architecture, training procedure, and evaluation metrics. Next, in Sections 3.3 and 3.4, we present quantitative and qualitative colorization results on a classical image dataset to demonstrate the influence of the loss function an color spaces on the results. Finally, Section 3.5 presents the main conclusions of this chapter.

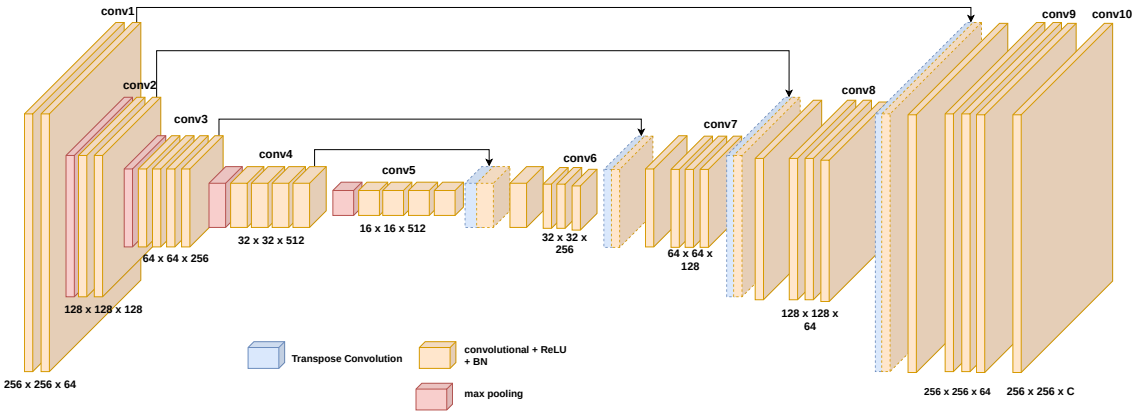


Figure 3.1: Summary of the baseline U-Net architecture used in our experiments. It outputs a $256 \times 256 \times C$ image, where C stands for the number of channels, being equal to 2 when estimating the missing chrominance channels and to 3 when estimating the RGB components.

3.2 Experimental setup

In this section, we present the framework that we will use for evaluating the influence of color spaces and losses on image colorization results. First, we detail the architecture and, second, the dataset used for training and testing.

3.2.1 Colorization framework

The architecture used in our experiments is an encoder-decoder U-Net deep network composed of five stages (see Figure 3.1). All convolutional blocks are composed of two 2D convolutional layers with 3×3 kernels, each one followed by 2D batch normalization (BN) and a ReLU activation. For the encoder, downsampling is done with max pooling layers after each convolutional block. After each downsampling, the number of filters is doubled in the following block. For the decoder, upsampling is done with 2D transposed convolutions (4×4 kernels with stride 2). At a given stage, the corresponding encoder and decoder blocks are linked with skip connections: feature maps from the encoder are concatenated with the ones from the corresponding upsampling path and fused using 1×1 convolutions. More details can be found in Table 3.1. The encoder architecture is identical to the CNN part of a VGG-19 network (Simonyan et Zisserman, 2015). It allows us to start from pretrained weights initially used for ImageNet classification. Moreover, the encoder architecture choice was motivated by the fact that most deep learning-based approaches use a VGG-type architecture to generate the missing chrominances.

The training settings are described as follows:

- Optimizer: Adam.
- Learning rate: 2e-5 as in ChromaGAN (Vitoria et al., 2020).
- Batch size: 16 images (approx. 11 GB RAM usage on Nvidia Titan V).
- All images are resized to 256×256 for training which enable using batches. In practice, to keep the aspect ratio, the image is resized such that the smallest dimension

Layer type	Output resolution
Input	3 x H x W
Conv1 + Max-pooling	64 x H/2 x W/2
Conv2 + Max-pooling	128 x H/4 x W/4
Conv3 + Max-pooling	256 x H/8 x W/8
Conv4 + Max-pooling	512 x H/16 x W/16
Conv5 + Conv. Transpose (I)	512 x H/8 x W/8
Conv6 + Conv. Transpose (II)	256 x H/4 x W/4
Conv7 + Conv. Transpose (III)	128 x H/2 x W/2
Conv8 + Conv. Transpose (IV)	64 x H x W
Conv9	64 x H x W
Conv10	C x H x W

Table 3.1: Detailed architecture and output resolution for each block.

matches 256. If the other dimension remains larger than 256, we then apply a random crop to obtain a square image. Note that the random crop is performed using the same seed for all trainings.

When generating images, it is crucial to remain in the range of acceptable values of color spaces. In particular, we must ensure that the final image takes values between 0 and 255. In our implementation, we use simple clipping on final RGB values. Other strategy are sometimes considered as in (Iizuka et al., 2016) where the ab components are globally normalized so they lie in the [0,1] range of the Sigmoid activation function.

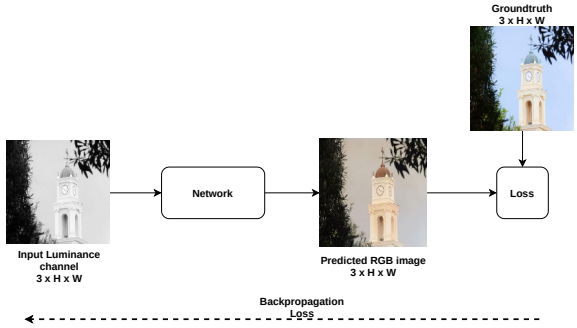
3.2.2 Training and testing images

Throughout our experiments, we use the COCO dataset (Lin et al., 2014), containing various natural images of different sizes. COCO is divided into three sets that approximately contain 118k, 5k, and 40k images that, respectively, correspond to the training, validation and test sets. Note that we carefully remove all grayscale images, which represent around 3% of the overall amount of each set. Although larger datasets such as ImageNet have been regularly used in the literature, COCO offers a sufficient number and a good variety of images so we can efficiently train and compare numerous models.

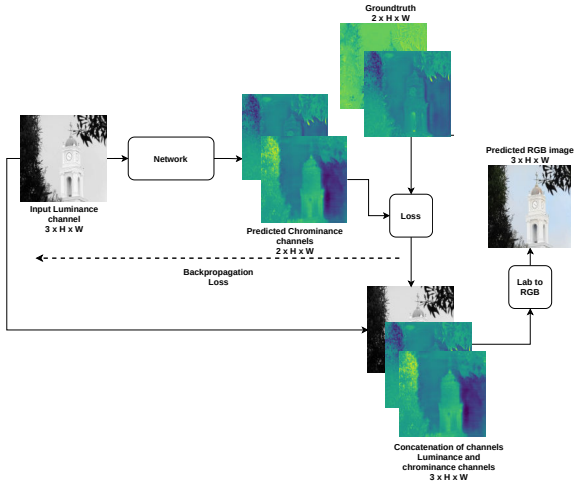
3.3 Influence of color spaces

The goal of the whole colorization process is to generate RGB images that look visually natural. When training on different color spaces (more details in color spaces in Section 2.3), one must decide which color space is used to compute losses and when is the conversion back to RGB performed. In this section, we propose to experiment with three learning strategies to compare RGB, YUV, and Lab color spaces (see Figure 3.2):

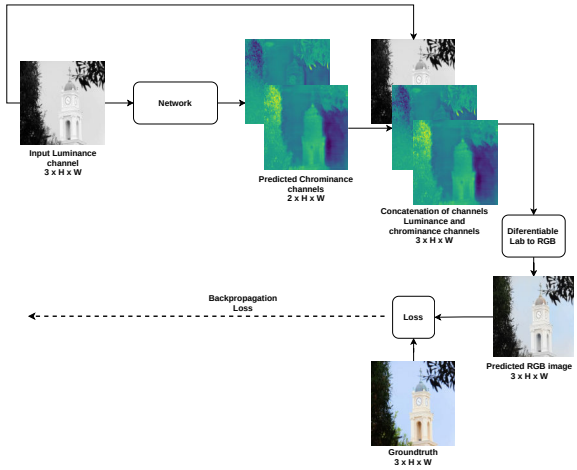
- *RGB*: in this case, the network takes as input a grayscale image L and directly estimates a three-channels RGB image of size $256 \times 256 \times 3$. The loss is done directly in the RGB color space. This strategy is illustrated in Figure 3.2a.
- *YUV and Lab Luminance/chrominance*: in this case, the network takes as input a grayscale image considered as the luminance (L for Lab, Y for YUV) and outputs



a. Learning strategy directly predicting the RGB colors.



b. Learning strategy predicting the two chrominance channels.



c. Learning strategy predicting the two chrominance channels, then converting to RGB.

Figure 3.2: Illustration of the different learning strategies for our proposed framework.

two chrominance channels (a, b or U, V). The loss compares the output with the corresponding chrominance channels of the ground-truth image converted to the luminance/chrominance space. After concatenating the initial luminance channel to the inferred chrominances, the image is converted back to RGB for visualization purposes. This strategy is illustrated in Figure 3.2b.

- *LabRGB*: as in the previous case, the network takes as input the luminance and estimates the corresponding two chrominance channels. After concatenating with the corresponding luminance channel, they are converted to the RGB color space and the loss is computed directly there. Notice, that in this last case, as the loss is computed on RGB color space, the conversion must be done in a way that is differentiable to be able to compute the gradient and allow the back-propagation step. We perform the color conversion using the color module in the `Kornia` library. Kornia (Riba et al., 2020) is a differentiable library that consists of a set of routines and differentiable modules to solve generic computer vision problems. It allows classical computer vision tasks to be integrated into deep learning models. Computing the loss on RGB images instead of chrominance ones enables to ensure images are similar to ground-truth after the clipping operation needed to fit into the RGB cube. This strategy is illustrated in Figure 3.2c.

The rest of the section presents quantitative and qualitative results obtained with the three strategies discussed above.

Remark: During training, all images are resized to 256× 256. In addition, one advantage of using luminance/chrominance spaces is that only chrominance channels are predicted. It is, therefore, possible to keep the original structure content of the luminance channels.

Evaluation setup. For this analysis, we have considered, as loss function, the L2 loss and the VGG-based LPIPS as in Equation (2.7) which was introduced in (Ding et al., 2021) as a generalization of the feature loss (Johnson et al., 2016).

Note that to compute the VGG-based LPIPS loss, the output colorization always has to be converted to RGB, even for YUV and Lab color spaces (as in Figure 3.2(c)), because this loss is computed with a pre-trained VGG expecting RGB images as input. Since VGG-based LPIPS is computed on RGB images, the two strategies *Lab* and *LabRGB* are the same. Our experiments have shown that same conclusions can be drawn with other losses. For testing, we apply the network to images at their original resolution, while training is done on batches of square 256×256 images.

Quantitative evaluation. There is no standard protocol for quantitative evaluation of automatic colorization methods. We choose to rely on the more generally used and more recent ones: L1 (MAE), L2 (MSE), PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018b) and FID (Fréchet Inception Distance) (Dowson et Landau, 1982), which are previously defined in Section 2.7.

The results are presented in Table 3.2. In terms of these metrics, the best results are obtained with YUV color space except for L1 and FID, even if not by much. The results in Table 3.2 also indicate that Lab does not outperform other color spaces when

Color space	Loss function	L1 ↓	L2 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
RGB	L2	**0.04458**	0.00587	22.3136	0.9255	0.1606	**7.4223**
YUV	L2	0.04469	**0.00562**	**22.5052**	**0.9278**	**0.1593**	7.6642
Lab	L2	0.04488	0.00585	22.3283	0.9250	0.1613	8.1517
LabRGB	L2	0.04608	0.00589	22.2989	0.9209	0.1698	8.3413
RGB	LPIPS	0.04573	0.00577	22.3892	0.9197	0.1429	**3.0576**
YUV	LPIPS	0.04460	**0.00557**	**22.5438**	0.9097	**0.1400**	3.3260
Lab	LPIPS	**0.04374**	0.00566	22.4699	**0.9228**	0.1403	3.2221

Table 3.2: Quantitative evaluation of colorization results for different color spaces. Metrics are used to compare ground-truth to every images in the 40k test set. Best and second best results by column are in bold and underlined respectively.

using a classic reconstruction loss (L2), while better results are obtained when using the VGG-based LPIPS. Thus, using a feature based reconstruction loss is better suited as was already the case in exemplar-based image colorization methods (He et al., 2018; Yin et al., 2021) where different features for patch-based metrics were proposed for matching pixels. LabRGB strategy gets the worst quantitative results based on Table 3.2. One would expect to get the "best of both" color spaces while recovering from the loss of information in the conversion process. However, this is not reflected with these particular evaluation metrics. The LabRGB line for VGG-based LPIPS is not included, as it would be identical to the Lab one. Also, note that the quantitative evaluation is performed on RGB images as opposed to training which is done for specific color spaces (RGB, YUV, Lab and LabRGB).

Qualitative evaluation. We qualitatively analyze the results obtained by training the network with different color spaces as explained in Section 3.2.

Figure 3.3 shows results on images and objects (here person skiing, stop sign and zebra) with strong contours that were highly present in the training set. The colorization of these images is good for any color space. Nevertheless, YUV has the tendency to sometimes create artifacts that are not predictable. This is visible with the blue stain in the YUV-L2 zebra and the yellow spot in the YUV-LPIPS zebra. One can also notice that the overall colorization tends to be more homogeneous with LabRGB-L2 than with Lab-L2 as can be seen for instance on the wall behind the stop signs, the grass and tree leafs in the zebra image which suggests that it might be better to compute losses over RGB images. A similar remark is valid for the VGG-based LPIPS results as can be seen for instance in the homogeneous colorization of the sky in the person skiing image where the loss is again computed over the RGB image. This indicates that there could be an additional influence on the results when using VGG-based LPIPS given that the predicted color image is converted back to RGB before backpropagation.

Figure 3.4 presents results on images where the final colorization is not consistent over the whole image. On the first row, the color of the water is stopped by the chair legs. On the second row the color of the grass and the sky are not always similar on both side of the hydrant. LabRGB seems to reduce this effect. This happens when strong contours seem to stop the colorization and is independent on the color space. Global coherency can only be obtained if the receptive field is large enough and that self-similarities present in natural images is preserved. These results highlight that efforts must be put on the design

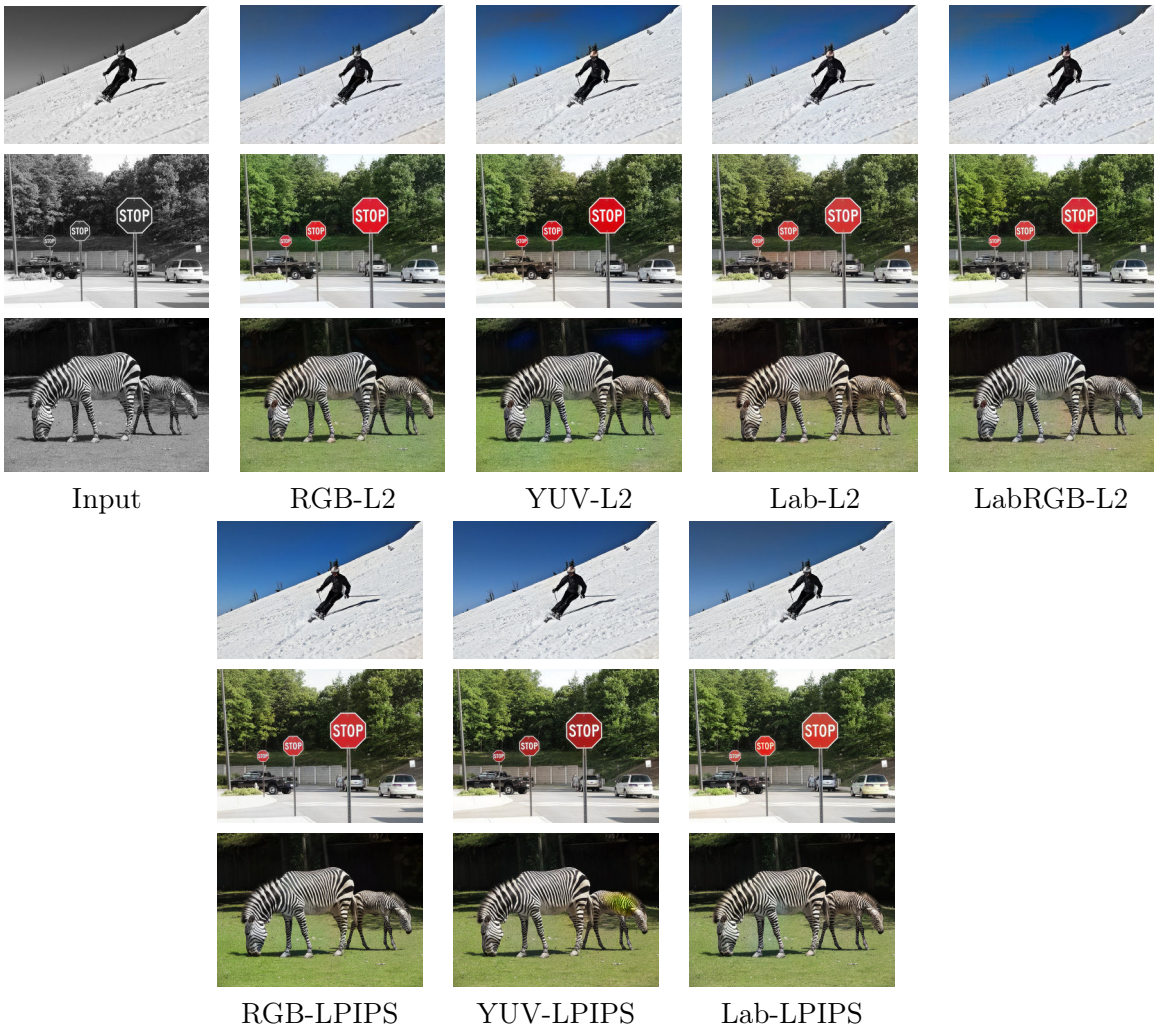


Figure 3.3: Colorization results with different color spaces on images that contain objects, have strong structures and that have been seen many times in the training set. The three first rows are with L2 loss and the three last ones with VGG-based LPIPS.

of architectures that would impose these constraints.

One major problems in automatic colorization results come from color bleedings that occur as soon as contours are not strong enough. Figure 3.5 illustrates this problem in different contexts. On the first row, the color from the flowers bleeds to the wall. On the second row, the green of the grass bleeds to the shorts. Finally, on the last row, the green of the grass bleeds to the neck of the background cow. These effects are independent from the color space or the loss. Some methods reduce this effect by introducing semantic information (e.g., (Vitoria et al., 2020)) or spatial localization (e.g., (Su et al., 2020), (Chapters 4, 5), while others achieve to reduce it by considering segmentation as an additional task (e.g., (Kong et al., 2021), (Chapter 6)). Finally, Figure 3.6 presents colorization of images containing many different objects. We see that final colors might

Figure 3.4: Colorization results with different color spaces on images that exhibit strong structures that may lead to inconsistent spatial colors. The two first rows are with L2 loss and the two last one with VGG-based LPIPS.

be dependent on the color spaces, and are more diverse and colorful with Lab color space. LabRGB strategy with L2 loss is probably the more realistic, statement that holds with the VGG-based LPIPS.

The qualitative evaluation does not point to the same conclusion as the quantitative one. According to Table 3.3, the best colorization is obtained for YUV color space. However, the qualitative analysis shows that even if in some cases colors are brighter and more saturated in other ones it creates unpredictable color stains (yellowish and blueish). This raises the question on the necessity to design specific metrics for the colorization task, which should be combined with user studies. Also, in the qualitative evaluation one can observe that when working with LabRGB instead of Lab the overall colorization result looks more stable and homogeneous as opposed to what is concluded in the quantitative evaluation.

3.4 Influence of losses

To compare the influence of the objective loss in the colorization results, we train the network described in Section 3.2 by changing the objective loss. In particular, we train the network with the L1 loss, the L2 loss, the VGG-based LPIPS, the combination of WGAN plus L2 losses, and the combination of WGAN and VGG-based LPIPS. To the best of our knowledge, the combination of the VGG-based LPIPS loss with a WGAN training procedure is novel and has not been proposed in the recent literature.

Figure 3.5: Colorization results with different color spaces on images that contain small contours which lead to color bleeding. The two first rows are with L2 loss and the two last ones with VGG-based LPIPS.

For each of the these losses, depending on the chosen color space, we estimate:

- either the two (a, b) chrominance channels given the luminance channel L as input;
- or the three (R, G, B) color channels given a grayscale image as input.

Figure 3.6: Colorization results with different color spaces on images that contain several small objects which end up with different colors depending on the color spaces used. The three first rows are with L2 loss and the three last ones with VGG-based LPIPS.

In the rest of the section we present a quantitative and qualitative comparison for all of these combinations. Note that to compute the VGG-based LPIPS loss, the output colorization always has to be converted to RGB (in a differentiable way), even for Lab color space, because this loss is computed with a pre-trained VGG expecting RGB images as input. To this end, we have used the Kornia implementation of differentiable color space

conversions (Riba et al., 2020).

Color space	Loss function	MAE ↓	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Lab	L1	0.04407	0.00589	22.3020	**0.9268**	0.1587	8.8109
Lab	L2	0.04488	0.00585	22.3283	0.9250	0.1613	8.1517
Lab	LPIPS	**0.04374**	**0.00566**	**22.4699**	0.9228	**0.1403**	3.2221
Lab	WGAN+L2	0.04459	0.00582	22.3512	0.9243	0.1609	7.6127
Lab	WGAN+LPIPS	0.04383	0.00568	22.4541	0.9223	0.1406	**3.1045**
RGB	L1	**0.04385**	0.00587	22.3119	**0.9268**	0.1583	8.0125
RGB	L2	0.04458	0.00587	22.3136	0.9255	0.1606	7.4223
RGB	LPIPS	0.04573	**0.00577**	**22.3892**	0.9196	**0.1429**	3.0576
RGB	WGAN+L2	0.05256	0.00651	21.8667	0.8559	0.2469	15.4780
RGB	WGAN+LPIPS	0.04901	0.00679	21.6806	0.9137	0.1495	**2.6719**

Table 3.3: Quantitative evaluation of colorization results for different loss functions. Metrics are used to compare ground-truth to every images in the 40k test set. Best and second best results by column are in bold and underlined respectively.

Quantitative evaluation. Table 3.3 shows the quantitative results comparing five losses, namely, the L1 loss, the L2 loss, the VGG-based LPIPS, the combination of WGAN plus L2 losses, and the combination of WGAN and VGG-based LPIPS (denoted in Table 3.3 as L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS, respectively). The first five rows display this assessment when the used color space is Lab (*i.e.*, the model estimates the two ab chrominance channels), while for the last five rows the used color space is RGB (*i.e.*, the model estimates the three RGB color channels). In particular, let us remark that the quantitative evaluations are always performed in the final RGB color space. Thus, even when the model is trained to estimate the ab chrominance channels, the resulting Lab color image is converted to the RGB color space to compute the evaluation metrics.

From the results in Table 3.3 we observe that for the analyzed dataset, the models trained with the VGG-based LPIPS loss function provide overall better quantitative results, for both Lab and RGB color spaces. This is especially true for the perceptual metrics LPIPS and FID, as they are strongly correlated to this loss function. The fact that the VGG-based LPIPS training loss is computed on RGB color space (as this loss is computed with a pre-trained VGG expecting RGB images as input) and also are all quantitative results might be related to the performance. In the same spirit, we can observe a slight correlation between the used training loss and the quantitative metric. For instance, when training with L1, MAE results are better. However, we can see that L2 loss is not at the top in any of the metrics, while we could have expected in the case of MSE or PSNR, but this is not the case.

Nevertheless, no strong tendency clearly emerges from this table: for many metrics, the different losses do not differ so much from one another and could be in the margin of error. From our analysis, we hypothesize that, apart from the chosen objective function, the network architecture design, and the training process, may play a very important role as a prior on the colorization operator.

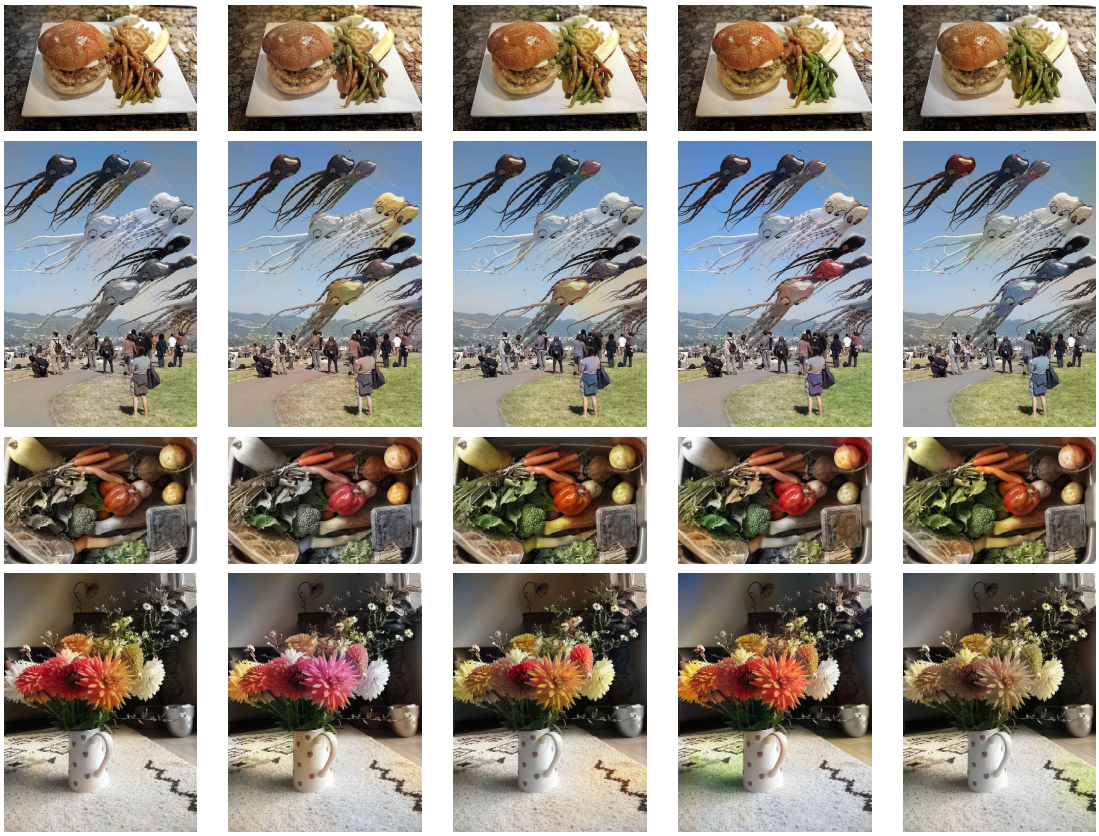


Figure 3.7: Examples where multiple objects are in the same image. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2 and WGAN+LPIPS. The used color space is Lab for all the cases (that is, the model estimates two ab chrominance channels).

Qualitative evaluation. Figures 3.7, 3.8 and 3.9 show a qualitative experimental comparison of the five losses, namely, L1, L2, VGG-based LPIPS, WGAN+L2 and WGAN+VGG-based LPIPS. In all cases, the models were trained on the Lab color space. Still, we recall that any model based on VGG-based LPIPS loss requires to convert the predicted image to the RGB color space in a differentiable way (*i.e.*, with Kornia (Riba et al., 2020)).

In Figure 3.7, we can see some results obtained for each of the studied losses in images with multiple objects. We can observe that each loss brings slightly different colors to objects. Overall, VGG-based LPIPS and WGAN losses generate shinier and more colorful images (it can be seen, for instance in the sky, grass, and vegetables), although we can observe colorful examples in the case of the L2 loss in the example of the flowers or vegetables. However, WGAN hallucinates more unrealistic colors as can be seen on the table or the wall on the image with a flower of the last row of Figure 3.7. This effect can be reduced by improving architecture and semantic features (*e.g.*, (Vitoria et al., 2020)) or by introducing spatial localization (*e.g.*, (Su et al., 2020)). Besides, by comparing the two last columns obtained with the models trained with the adversarial strategy WGAN combined

Figure 3.8: Examples to evaluate shyniness of the results. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2 and WGAN+LPIPS. The used color space is Lab for all the cases.

with, respectively, the L2 or the VGG-based LPIPS, one can observe that WGAN+VGG-based LPIPS tends to homogenize colors (*e.g.*, some of the balloons take similar color to the sky on the second row; the flowers on the fifth have grayish colors, more similar to the wall). WGAN+VGG-based LPIPS also tends to have less bleeding than WGAN+L2.

The generation of more vivid colors with VGG-based LPIPS and WGAN losses in also visible on Figure 3.8. The grass and bushes are more green and look more natural. However, none of the losses give consistency to all the limbs of the tennis player on the first row (*e.g.*, the right leg).

Figure 3.9 shows results on objects, here zebra and stop sign, with strong contours that were highly present in the training set. The colorization of this object is impressive for any loss. None of the losses manage to properly colorize the person near the center car on the first row. This type of example, could be improved by learning high-level semantics on the image content. This is because high-level semantics brings contextual understanding to the model, making it capable of recognizing the overall scenario of the image, for instance, bringing object relationships, which can help to better choose object colors. Finally, high-level semantics not only improves the accuracy of color predictions but also ensures that the resulting colorized image feels more natural and contextually appropriate for the human eye. In the next Chapter 4, we will propose a novel approach to combine low and high-level semantics based on superpixels and attention mechanisms.

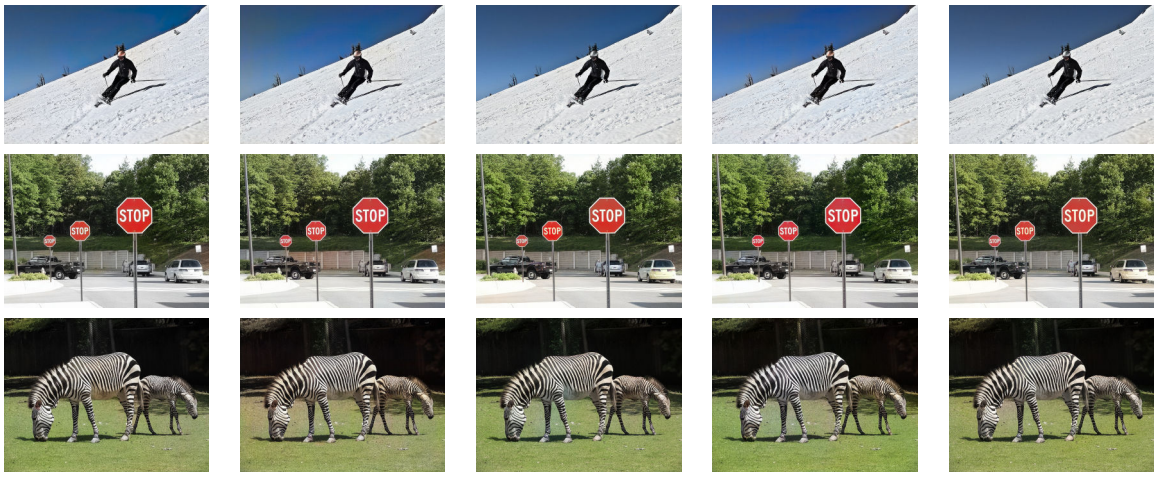


Figure 3.9: Colorization results on images that contain objects have strong structures and that have been seen many times in the training set. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2 and WGAN+LPIPS. The used color space is Lab for all the cases.

3.5 Conclusions and future works

This chapter has presented the role of the color spaces and loss functions in automatic colorization by learning. Using a fixed standard generative network, we have shown, qualitatively and quantitatively, that choosing the right color space and the right loss functions is not straightforward and might depend on several factors, such as the architecture or the type of images. We found that the YUV color space provides slightly better results in quantitative terms, but at the cost of more artifacts. However, we argue that most efforts should be made on the network architecture design, as it could have a stronger impact on the colorization results than the choice of color space or loss function. In our analysis, we used a U-Net-based architecture. For the employed architecture, the models, including the VGG-based LPIPS loss function, provide overall slightly better results, especially for the perceptual metrics LPIPS and FID.

Additionally, results in Section 3.3 showed one of the common problems using automatic colorization methods which is the color bleeding, that occurs when the contours of the grayscale image are not strong enough. Therefore, in Chapter 4, we face this challenge by developing a novel edge-aware attention mechanism that helps to avoid this bleeding issue, also, it is capable of calculating similarities with high-resolution features maps. Finally, after analyzing the impact of architectures and losses, we will use the colorization deep-learning architecture presented in this chapter as the baseline, as well as the VGG-based LPIPS loss function, for our proposal introduced in Chapter 5. Interesting topics for future research include developing methods that can learn to project the colorized image into the RGB color cube while preserving good image quality (Pierre et al., 2015b), as well as methods that can produce multiple outputs in the same trend as HistoGAN (Afifi et al., 2021).

Chapter 4

Super-Attention mechanism and its application to color transfer

Table of contents

Summary

In this chapter, we propose a new method for matching high-resolution feature maps from CNNs using attention mechanisms. To avoid the quadratic scaling problem of all-to-all attention, this method relies on a superpixel-based pooling dimensionality reduction strategy. From this pooling, we efficiently compute non-local similarities between pairs of images. To illustrate the interest of these new methodological blocks, we apply them to the problem of color transfer between a target image and a reference image. While previous methods for this application can suffer from poor spatial and color coherence, our approach tackles these problems by leveraging on a robust non-local matching between high-resolution low-level features. Finally, we highlight the interest in this approach by showing promising results in comparison with state-of-the-art methods.

Related publications

[Carrillo *et al.* 2021] **H. Carrillo,** M. Clément, A. Bugeau. "Superpixel-based matching of high-resolution deep features for color transfer." *Journées francophones des jeunes chercheurs en vision par ordinateur (ORASIS)*, 2021.

[Carrillo *et al.* 2022] **H. Carrillo,** M. Clément, A. Bugeau. "Non-local matching of superpixel-based deep features for color transfer." *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2022.

Contributions

The main contributions in this chapter are:

- We propose the super-features encoding block, which extracts deep feature maps using superpixel decomposition.
- We propose a robust non-local similarity between super-features using an attention mechanism.
- We build upon (Giraud et al., 2017) and include these similarities in a non-local color fusion framework, achieving promising results on several target and reference image pairs.

4.1 Introduction

In this chapter, we introduce a methodological core of a novel attention block that will be used in several contributions of this thesis. Here, we propose an attention block that matches high-resolution feature maps from CNNs in a computationally efficient manner. Our method relies on a superpixel-based pooling dimensionality reduction strategy to avoid the quadratic scaling problem of all-to-all attention. From this pooling, we efficiently compute non-local similarities between pairs of images. To illustrate the interest of these new methodological blocks, we apply them to the problem of color transfer between a color target image and a color reference image. Also, this block can be directly applied to a more general task such as colorization (see Chapter 5 and Chapter 6).

Color transfer is a technique that aims to transfer the color characteristics of a reference image to a target image. This technique has a wide range of applications, such as image editing, image restoration, and style transfer. In recent years, deep learning-based methods have achieved state-of-the-art results in color transfer (He et al., 2019; Lee et al., 2020b). However, many of these methods suffer from two main limitations. First, they rely on all-to-all attention mechanisms to match feature maps between the target and reference images. This can lead to quadratic scaling in the number of pixels, making the computation intractable for high-resolution images. and second, focus on matching high-level semantic features, which can lead to poor spatial and color coherence in the resulting colorized image. Therefore, we use our super-attention block to tackle the problems of poor spatial and color coherence by leveraging a robust non-local matching between high-resolution, low-level features. Finally, we present a comprehensive evaluation of our proposed method, comparing it to state-of-the-art color transfer methods.

The outline for this chapter is the following: Sections 4.2 and 4.3, we introduce the basis of the attention mechanism and the superpixels algorithm applied to the image editing task, respectively. Then, in Section 4.4, we introduce the literature on the color transfer task. Next, in Section 4.5, we detailed our proposal called *super-attention* mechanism, which avoids the quadratic scaling problem of all-to-all attention by using a superpixel-based pooling dimensionality reduction strategy. Then, in section 4.6, we extend our super-attention block to the color transfer application based on the color fusion framework (Giraud et al., 2017). Finally, in section 4.7, evaluate our method for the color transfer task.

4.2 Attention as a non-local operator

Non-local operators were introduced in image processing in (Buades et al., 2005) with the non-local means framework, initially used to filter out image noise by computing a weighted mean of all pixels in an image. Non-local means allow remote pixels to contribute to the filtered response, achieving less loss of details. The general equation for the non-local algorithm is the following:

$$I(p) = \frac{1}{C(p)} \sum_{p_n \in \Omega} f(p, p_n) \cdot I_n(p_n), \tag{4.1}$$

where $I(p)$ and $I_n(p_n)$ are the denoised image and noisy image with p and p_n repre-

senting the pixels position. The weight function $f(p, p_n)$ is based on the similarity between the pixels in position p and p_n and their neighborhoods (the function f is often defined as a Gaussian function of the Euclidean distance). The $C(p)$ term is the normalization constant to ensure that the weights sum up to one, defined as $C(p) = \sum_{p_n \in \Omega} f(p, p_n)$. The goal of the weight function is to ensure that pixels with similar neighborhoods have higher weights. Taking advantage of this, the non-local means algorithm was extended to non-local features matching for super-resolution (Glasner et al., 2009), or inpainting (Wexler et al., 2004), proving to achieve robust global features similarities.

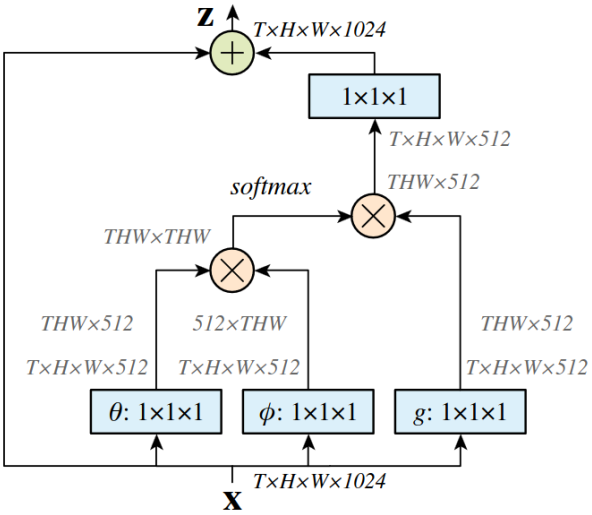


Figure 4.1: Diagram of a self-attention block from (Wang et al., 2018b), where video tensor shapes are denoted by their dimensions, for instance, $T \times H \times W \times C$. Matrix multiplication and element-wise summation are symbolized by "$\otimes$" and "$\oplus$", respectively. Blue boxes indicate $1 \times 1 \times 1$ convolutions, and the softmax function is applied row-wise.

Non-local similarities in neural network architectures were introduced in (Vaswani et al., 2017), in this work, they proposed a new deep learning architecture called transformers. A transformer is an end-to-end neural network approach that includes self-attention and cross-attention layers, which compute non-local similarities between multi-level features. Figure 4.1 a detailed diagram of a self-attention mechanism.

This type of architecture succeeds as a state-of-the-art method due to the capacity and flexibility of these attention blocks. The recent work (Wang et al., 2018b) has bridged the gap between the self-attention mechanism (Vaswani et al., 2017) and non-local means. They stated that the self-attention mechanism captures long-range dependencies between deep learning features by considering all features in the calculation (an example is shown in Figure 4.2). From Equation (4.1) in (Wang et al., 2018b), they derive a non-local operation formulation for deep neural networks:

$$z(p) = \frac{1}{C(x)} \sum_{p_n \in \Omega} f(x_p, x_{p_n}) \cdot g(x_{p_n}), \tag{4.2}$$

Here, x is an input signal (often features from videos or images), and z is the output signal of the same size as x. The function g does an unary operation that computes a

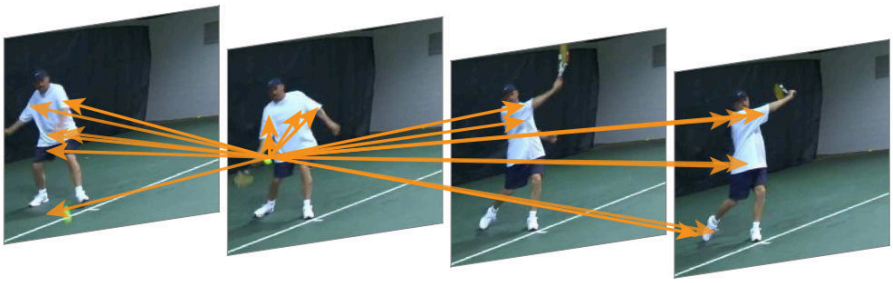


Figure 4.2: Example of self-attention operation for video classification in Kinetics from (Wang et al., 2018b). To predict the position (beginning of the arrows), attention allows the integration of non-local information from different locations and frames (end of arrows).

representation of the input signal at the position p_n. And finally, the response is normalized by a factor C(x). In the classic non-local mean (Buades et al., 2005), a usual choice for f is the Gaussian function. An extension, for example, of this function is to compute similarity in an embedding space, reducing the complexity of this calculation. Similarly, the self-attention module presented in (Vaswani et al., 2017) is a special case of non-local operations in the embedded gaussian version; that is, for a given p, $\frac{1}{C(x)} \sum_{\forall j} f(x_p, x_{p_n})$ can seem as the $softmax$ computation along p_n. Therefore, by replacing the previous assumption on Equation (4.2) we have the self-attention calculation from (Vaswani et al., 2017) $z = softmax\left(x^T W_\theta^T W_\phi x\right) g(x)$. The previous shows another point of view of relating attention blocks to classic image processing non-local means.

Recently, the authors of (Zhang et al., 2019), inspired by the self-attention mechanism, presented the cross-attention mechanism in the computer vision context, which computes the similarity between different feature maps (target and reference images). Both self-attention and cross-attention are mechanisms that help models understand and relate different features from signals. As mentioned before, self-attention allows a model to look at all regions of a signal(image, video, etc) and understand the context and relationships within it. In contrast, the cross-attention goes a step further by connecting two different signals, such as matching features from two different images, allowing the model to focus on which parts of the first image are most relevant with respect to the second one. The principal drawback of such mechanism is the non-local operation, which has to be done on features with low dimensions due to computational overhead. In addition, low-resolution features usually do not carry sufficient information for calculating a robust pairwise similarity. For instance, deep features mainly carry high-level semantic information related to a precise application (*i.e.,* classification) that can be less relevant for high-resolution similarity calculation or matching purposes.

In this chapter, we compute similarities between high-resolution deep features obtained from pre-trained convolutional neural networks, as this retains rich low-level characteristics. Due to the dimensionality issue, we exploit existing superpixels extractor in order to match these high-resolution features. To illustrate the interest of this super-features matching operation, we apply it to the problem of color transfer. Color transfer aims at changing the color characteristics of a target image by copying the ones from a reference image. Ideally, the result must reach a visually pleasant image, avoiding possible artifacts or improper colors. It covers various applications in areas such as photo enhancement, films

post-production, and artistic design. Transferring the right colors requires computing meaningful similarities between the reference and the target images. These similarities must preserve important textures and structures of the target image. Therefore, we will show that high-resolution features are essential.

4.3 Superpixels in image editing

Exploiting superpixel representation allows finding interesting region's characteristics in images, such as color and texture consistency (Achanta et al., 2012). Many advantages can be derived using this type of decomposition, for instance, dimensionality reduction by grouping pixels with similar characteristics (Van den Bergh et al., 2015). The most used superpixel algorithm is called SLIC (Achanta et al., 2012). This method first assigns regularly spaced cluster centers and iteratively allocates pixels to the nearest cluster based on a distance metric that emphasizes both color similarity and spatial closeness, typically in the CIELAB space (see Figure 4.3). The cluster centers are then recalculated as the mean of their assigned pixels. This process repeats until the clusters stabilize, usually after a few iterations. Many image processing and analysis applications can benefit from the superpixel algorithm. For example, image analysis typically involves recognizing and locating the different objects present in an image. This recognition can be done by a preliminary segmentation of the objects, which are then classified. Additionally, superpixels effectively capture the objects in an image, adhering to their edges while significantly compacting their representation. Figure 4.4 shows an example of superpixel representation using (Achanta et al., 2012) method. Consequently, they have been used in many image-processing tasks. For instance, they are employed for object localization (Fulkerson et al., 2009), edge detection (Arbelaez et al., 2009), video tracking (Reso et al., 2013), saliency detection (Yang et al., 2010), and multi-class object segmentation (Tighe et Lazebnik, 2010).

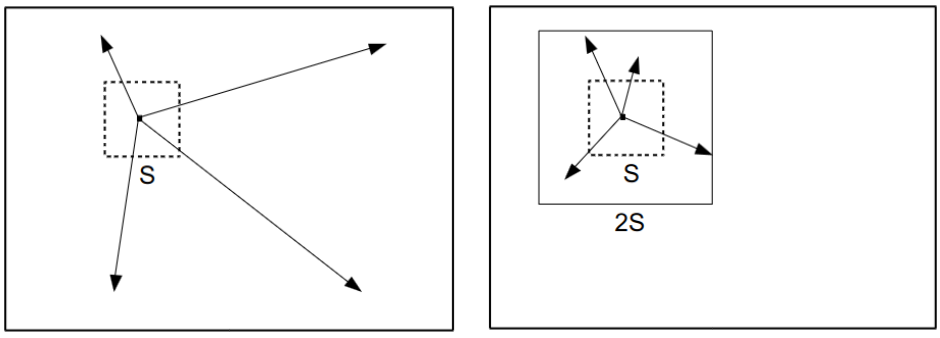


Figure 4.3: Illustration of superpixel constrained search region from (Achanta et al., 2012). (a) standard k-means searches starting from a superpixel of size $S \times S$ to the entire image. (b) SLIC searches in a limited area of size $2S \times 2S$.

However, the irregular form of the representation (variable size/shape) makes its usage difficult in computer vision tasks, especially the ones using deep learning approaches.

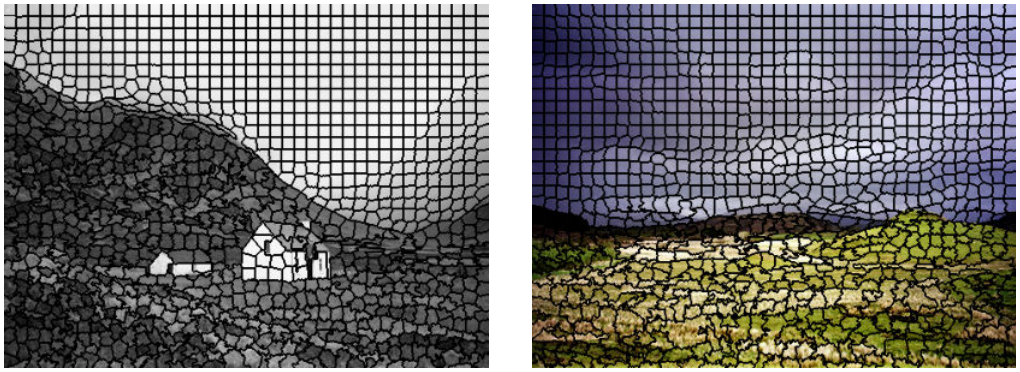


Figure 4.4: Example of decomposing an image using the SLIC algorithm (Achanta et al., 2012).

Nevertheless, some works have proposed some representation to cope with this issue. For instance, (Ihsan et al., 2020) uses a superpixel label map as an input image to a neural network to extract meaningful information for clothing parsing application. (He et al., 2015) presents the SuperCNN as a deep neural network approach for salient object detection. It uses superpixels to describe two 1-D sequences of colors in order to reduce the computational burden. Nonetheless, neither of the existing approaches effectively encodes deep learning features for each superpixel.

4.4 Literature on color transfer

Transferring the right colors requires computing meaningful similarities between the reference and the target images. These similarities must preserve important textures and structures of the target image. Most works on color transfer have focused on choosing the characteristics on which to compute similarities. These characteristics can be hand-crafted or learned using deep learning methods. The first one extracts image features by relying on manually predefined descriptors (*i.e.,* HOG (Dalal et Triggs, 2005), SIFT (Lowe, 2004)); however there is no guarantee that the descriptors are well suited for the task. The second solves this issue by learning the features from image dataset and leveraging on a training procedure, nonetheless feature dimensionality increases enforcing the usage on low-resolution images. Features similarities can be matched using global information of the images (*i.e.,* color histograms); or local information such as matching small regions on the images (*i.e.,* cluster segmentation, superpixel decomposition). In the literature, color transfer techniques can be classified into three classes: classic global-based methods, classic local-based methods, and deep learning methods.

Global methods. This type of method considers global color statistics and does not take into account any spatial information. It was initially introduced in (Reinhard et al., 2001) which uses basic statistical tools (*i.e.,* mean, standard deviation) to match target and reference color information. (Pitié et Kokaram, 2007; Xiao et Ma, 2006) extend color matching on different color spaces to find an optimal color mapping between the images.

(Frigo et al., 2015; Ferradans et al., 2013) propose a global illuminant matching based on optimal transport color transfer for enforcing artifacts-free results. More complex methods such as (Murray et al., 2012) rely on Gaussian Mixture Models to create compressed signatures that ensure a compact representation of color characteristics between images. Nevertheless, as mentioned in (Pitié, 2020), these methods fail to ensure spatial consistency on resulting colors when content change (*i.e.,* transferring day and night images).

Local methods. This type of method relies on spatial color mappings (*i.e.,* segmentation, clustering) to match local regions of the target image and the reference image. (Liu et al., 2016) uses superpixel level style-related and style-independent feature correspondences. (Arbelot et al., 2017) implement a texture-based framework for matching local correspondence. Alternatively, (Tai et al., 2005) uses a probabilistic segmentation in order to impose spatial and color smoothness among local regions. Still, the method does not provide control over the matched superpixel. (Giraud et al., 2017) overcomes this limitation by proposing a constrained approximate nearest neighbor (ANN) patches and a color fusion framework on superpixels. In detail, the proposed method involves three main steps: superpixel segmentation, approximate nearest neighbor (ANN) matching, and color fusion. The input colors images are first segmented into superpixels using the SLIC algorithm (Achanta et al., 2012), which groups pixels with similar color and spatial proximity. Second, for each superpixel in the target image, an ANN search is performed to find the best matching superpixel in the source image. They constrained the search to limit the diversity of neighboring superpixels in the source image, which helps capture the global color palette of the source image. Finally, the colors of the matched superpixels are then transferred to the target image using a color fusion framework, which preserves the structure and initial exposure of the target image while adapting to the selected source colors. Figure 4.5 shows an example of color transfer using (Giraud et al., 2017) method. However, in this type of local method, target and reference images are required to share strong similarities.

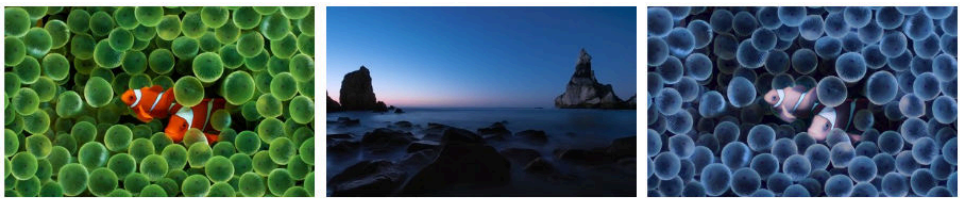


Figure 4.5: Example of color transfer using the method proposed by (Giraud et al., 2017). In color transfer, the inputs are the target and reference color images. The objective of this task is to apply the color palette of the source image to the target image while preserving the content and structure of the target image.

Deep learning methods. This method brings to the matching semantic-related characteristics from the target image and reference image. Recently (Lee et al., 2020b) propose a deep neural network architecture that leverages on color histogram analogy for color

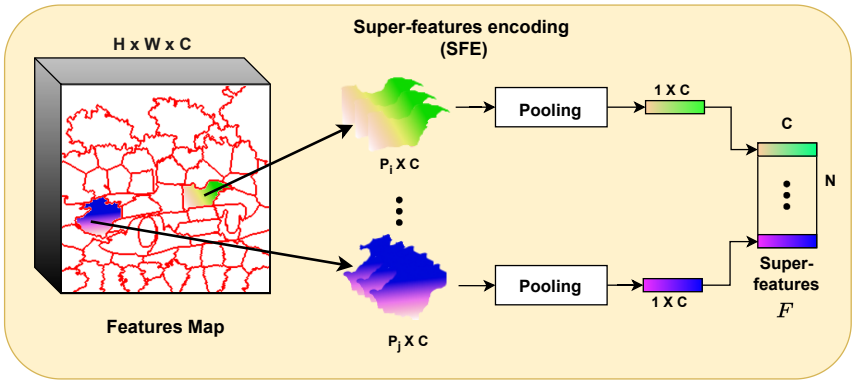


Figure 4.6: Diagram of our super-features encoding proposal (SFE). This proposal takes as input a feature map of size $H \times W \times C$, in which each superpixel is extracted and encoded in vectors of size $C \times P_i$ pixels. Afterward, the vectors are pooled channel-wise and, finally, stacked in the super-features matrix F with size $C \times N$ number of superpixels.

transfer. The latter uses target and reference histograms as input to exploit global histogram information over a target input image. (He et al., 2019) relies on semantically meaningful dense correspondence between images. Nonetheless, this type of method relies on pure semantic features (low-level features), which leads to imprecise results if images from different scenes or instances are used.

4.5 Super-attention mechanism

In this section, we present our superpixel based framework to match high-resolution features between two RGB images T and R of size $\mathbb{R}^{H \times W \times 3}$. In the following, we will refer to R as the target image and T as the reference image to be consistent with the color fusion application.

4.5.1 Super-features encoding

Let f_{T_ℓ} and f_{R_ℓ} be feature maps from a convolutional neural network at layer ℓ of T and R respectively. In the following, we will consider features coming from pre-trained deep convolutional networks, but our method could be applied to other types of hand-crafted features. More precisely, we focus on features extracted at the first three layers of a deep network, as they provide low-level features that suit diverse types of images. These feature maps then have high dimensions, typically the same size as the input image, times C channels with $H \times W \times C$ where $C = 64$, 128 or 256 for example.

A critical drawback of using high-resolution features for matching operations is the high computational complexity. Let the number of features in a feature map be $D = H \times W \times C$, then the complexity of the pixel-wise similarity computation is $\mathcal{O}\left(D^2\right)$. To solve this quadratic complexity problem, we implement an encoding layer based on superpixel representation. We first generate a superpixel map using a superpixel decomposition algorithm on the initial color images. Let us denote the target superpixel map by S_T, and the reference one by S_R. Each of these maps contains N_T and N_R superpixels respectively with P_i pixels each, where i is the superpixel index. Next, we extract features of size $C \times P_i$ for each

superpixel. These extracted features are then pooled spatially by averaging channel-wise and stacked as a matrix of size $C \times N$ called super-features F. Figure 4.6 illustrates this process. To sum up, the initial feature maps (f_{T_ℓ} and f_{R_ℓ}) pass from size $H \times W \times C$ to super-features encoding (F_{T_l} and F_{R_l}) of size $N_T \times C$ and, $N_R \times C$, making feasible operations such as correlation calculation between high-resolution features maps.

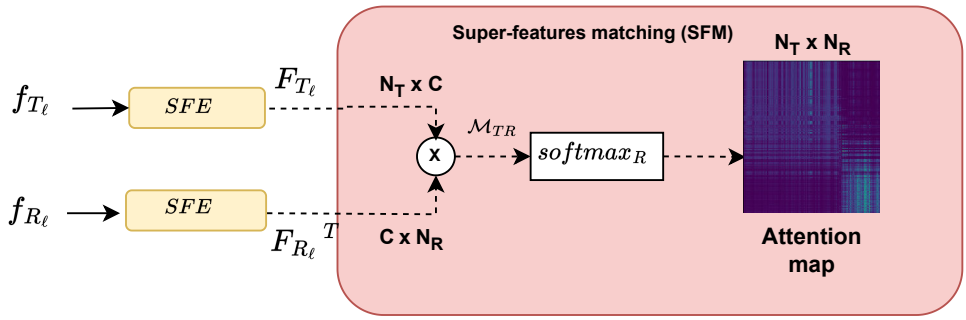


Figure 4.7: Diagram of our super-features matching (SFM). This layer takes a reference feature map f_R and a target feature map f_T as an input, and outputs an attention map at superpixel level by means of a non-local operation.

4.5.2 Super-features matching

Our super-features provide a compact encoding to compute the correlation between high-resolution deep learning features. Here, we take inspiration from the attention mechanism (Zhang et al., 2019) to achieve a robust matching between target and reference super-features. The process is illustrated in Figure 4.7. Mainly, we exploit non-local similarities between the target and the reference super-features by computing the attention map at layer ℓ as:

$$\mathcal{A}_\ell = \mathrm{softmax}_{\mathrm{R}_\ell}(\mathcal{M}_{T_\ell R_\ell}/\tau). \tag{4.3}$$

The $\mathrm{softmax}_{\mathrm{R}}$ operation normalizes row-wise the input into probability distributions, proportionally to the number of target superpixels N_R. Then, the final attention map $\mathcal{A}$ is the weighted sum of the attention maps at the first three layers ℓ:

$$\mathcal{A} = \frac{\sum_{\ell=1}^{3} \omega_\ell \mathcal{A}_\ell}{\sum_{\ell=1}^{3} \omega_\ell}. \tag{4.4}$$

The matrix $\mathcal{M}_{TR}$ is a correlation matrix between the target and reference super-features and is computed as:

$$\mathcal{M}_{T_\ell R_\ell}(i,j) = \frac{(F_{T\ell}(i) - \mu_{T_\ell}) \cdot (F_{R_\ell}(j) - \mu_{R_\ell})}{\|F_{T_\ell}(i) - \mu_{T_\ell}\|_2 \cdot \|F_{R_\ell}(j) - \mu_{R_\ell}\|_2}, \tag{4.5}$$

where μ_T and μ_R are the mean of each super-feature. We found that this normalization keeps correlation values less sensitive to changes on τ for different images. The attention map Equation (4.3) is the same non-local operator as the one proposed by (Zhang et al.,

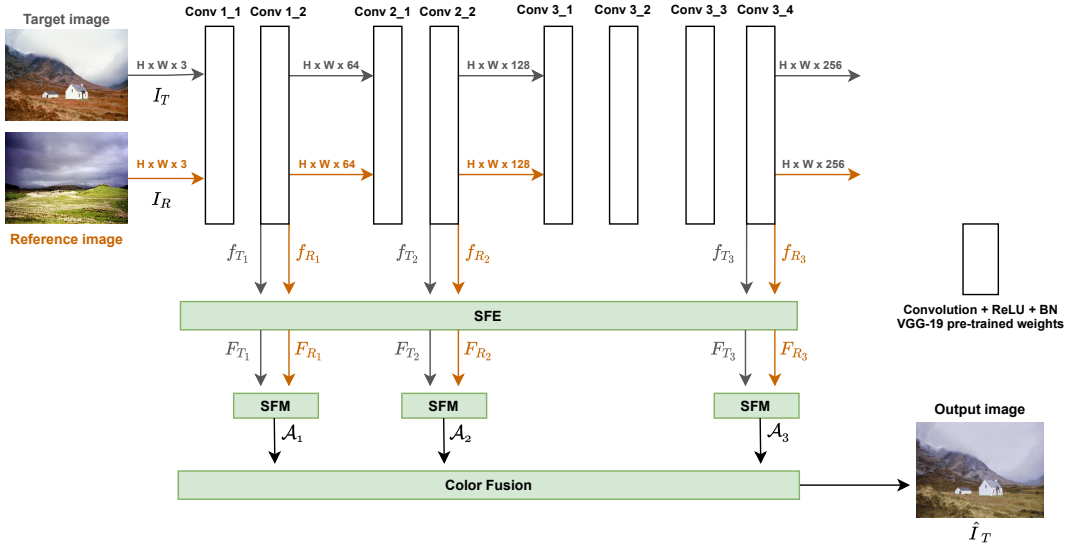


Figure 4.8: Diagram of our method using the first three levels of a modified VGG-19 architecture as our feature extractor. In our method, we remove max-pooling layers from the baseline VGG-19 architecture to capture similarities between high-resolution feature maps ($H \times W \times C_\ell$). Also, the diagram presents our two new blocks, the super-features encoding block (SFE) and the super-features matching block (SFM).

2019). However, their computation requires low-resolution features due to the inherent quadratic complexity problem (as mentioned in Section 4.5.1).

We solve this complexity problem thanks to our super-features encoding approach. Let $n = H \times W$ be the number of pixels in an image. Then, the number of features in a deep learning feature map is $D = n \times C$ which translate into a computational complexity of $\mathcal{O}(D^2) = \mathcal{O}(n^2C^2)$. In contrast, with our novel super-features encoding, if we set the number of superpixels in the order of $\sqrt{n}$, then instead we rewrite with $D_s = \sqrt{n} \times C$, resulting in $\mathcal{O}(D_s^2) = \mathcal{O}(n \times C^2)$. As $C \ll n$ can be ignored, we go from a quadratic to a linear complexity operation $\mathcal{O}(n)$. As a result, we can incorporate the correlation operation on large deep learning features from both target and reference images. Conversely, (Zhang et al., 2019) can only rely on deep-level features, usually the bottleneck features (*i.e.*, $H/8 \times W/8 \times C$) for similarities calculation.

4.6 Color transfer application

We now present our color transfer method. It consists of three blocks: 1) super-features encoding (SFE), 2) super-features matching (SFM), and 3) color fusion framework. The process is illustrated in Figure 4.8.

Our objective is to transfer colors from a reference R to a target image T. Concretely, this will be done by passing colors from R to T based on pairwise feature-related similarities.

To match colors at superpixel level, we rely on the attention map $\mathcal{A}$ and the average of each superpixel color. Specifically, we apply our attention map as a soft-weight on the average colors, resulting in a smooth correspondence.

Figure 4.9: Direct super-features matching using different τ values. Each of the rows depicts our results for direct matching using super-features from the first, second, and third layers. The use of the first-level features helps to preserve fine details and ensures color consistency. However, the second and third layers bring a more colorful and diverse matching between target and reference super-features.

Figure 4.9 shows a direct super-features matching between the target and reference images from Figure 4.8. This direct matching uses the weighted average color of its correspondence to replace the target's superpixels colors. Each row depicts the impact of different high-resolution feature maps from the first three levels of a pre-trained VGG-19. The first level (first row) brings fine details and spatial and color consistency onto the direct matching, while deeper features (second and third row) seem more sensitive to color features. Figure 4.9 also illustrates (in each column) the influence of temperature τ on the superpixel attention map. We can see that the probability distribution is over-smoothed (*i.e.,* gray average colors) for larger values of τ (*i.e.,* $\tau = 0.1$), meaning that several reference super-features match one target super-feature. Otherwise, a small τ value results in a hard one-to-one matching between a target and reference super-features (*i.e.,* $\tau = 1 \times 10^{-4}$).

4.6.1 Color fusion framework

Direct superpixel matching by averaging colors is not sufficient to obtain visually satisfying results. Image details are indeed lost at the superpixel level, as pixel features are averaged inside a superpixel (*i.e.,* door, windows, etc., in Figure 4.9). Therefore, we need to transfer color at the pixel level from our superpixel matching.

For clarity in further equations, we denote the position of the centroid $\bar{X}$ and color

centroids $\bar{I}$ of a superpixel j in an image I as:

$$\bar{X}(j) = \frac{\sum_{p \in S(j)} p}{P_j}$$

and

$$\bar{I}(j) = \frac{\sum_{p \in S(j)} I(p)}{P_j}$$

respectively, where S is the superpixel grid and j the current superpixel, and P_j is the number of pixels in superpixel j.

Inspired by the formulation of (Giraud et al., 2017), we compute the new value $\hat{T}(p)$ of each pixel p of the target as a weighted average of reference superpixel representative colors:

$$\hat{T}(p) = \frac{\sum_{j=1}^{N_R} W(p,j)\bar{R}(j)}{\sum_{j=1}^{N_R} W(p,j)}. \tag{4.6}$$

The weight matrix W depends firstly on the distance between pixel p and all target superpixel as in (Giraud et al., 2017), and secondly, on our attention map computed on super-features:

$$W(p,j) = \sum_{i=1}^{N_T} d(p,i)\mathcal{A}(i,j). \tag{4.7}$$

The intuition behind the attention map is the addition of more relevant information about reference super-features into the transfer process. The distance between pixel p and superpixel centroids is computed over both positions and colors with a Mahalanobis-like formulation:

$$d(p,i) = \exp\left(-\left(V_T(p) - \bar{V}_T(i)\right)^T \Sigma_i^{-1} \left(V_T(p) - \bar{V}_T(i)\right)\right), \tag{4.8}$$

with position and color vectors being $V(p) = [p, T(p)]$ and $\bar{V}_T(j) = \left[\bar{X}_T(j), \bar{T}(j)\right]$, and the spatial and colorimetric covariances of pixels in superpixel i:

$$\Sigma_i = \begin{pmatrix} \delta_s^2 \operatorname{Cov}(p) & 0 \\ 0 & \delta_c^2 \operatorname{Cov}(T(p)) \end{pmatrix}. \tag{4.9}$$

Parameters δ_s and δ_c weight the influence of color and spatial information, respectively.

Finally, as in (Giraud et al., 2017), after color fusion we apply a post-processing step using a color regrain algorithm (Pitié et al., 2005), which eventually matches the color distribution of R and the gradient of I_T. Figure 4.10 presents an example of our color transfer framework compared to the result of (Giraud et al., 2017). Visually, our results present better spatial consistency of colors. For instance, the sky on our results has more natural smooth color transitions compared to non-natural ones with (Giraud et al., 2017) (*i.e.,* yellow to blue).

4.6.2 Implementation details

Superpixel segmentation is done using the SLIC algorithm (Achanta et al., 2012), in which the number of superpixel depends on the actual size of the image. Experimentally, we set the number of superpixel as $3 \times \sqrt{n}$ where n is the number of pixels in the current image.

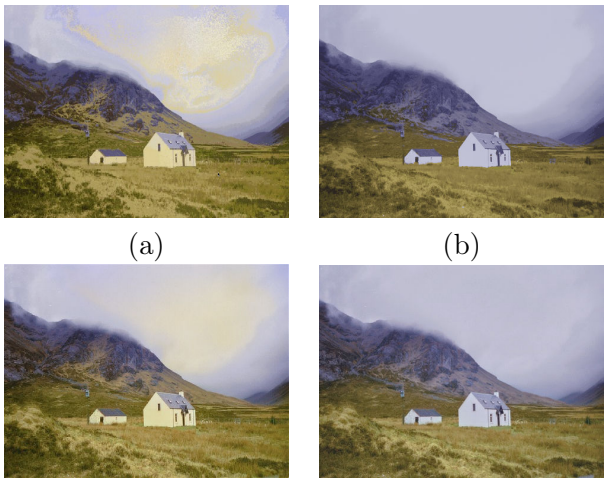


Figure 4.10: Color fusion framework results. (a) (Giraud et al., 2017) color fusion result. (b) Our color fusion result. (c) (Giraud et al., 2017) result + regrain. (d) Our result + regrain. The regrain algorithm is from (Pitié et al., 2005).

To build feature maps, we rely on a modified pre-trained VGG-19 (Simonyan et Zisserman, 2015) as our texture and color characteristics extractor, due to its simplicity and its 95.24% classification accuracy on the ImageNet Top-5 classes. The main modification was the removal of max-pooling layers from the first three levels (see Figure 4.8) as it highly improves matching results compared to using upsampling on max-pooled feature maps at Conv2_2 and Conv3_4 from the baseline VGG-19. Figure 4.11 exemplifies that matching low-resolution features does not preserve details nor retains color coherence, especially when going deeper into the architecture. Also, note that our approach can work with other types of CNN architectures regardless of their features dimensions.

In order to choose an optimal temperature τ value, we experimented on different images at distinct temperatures. Empirically, we obtain satisfying results using $\tau = 0.015$ and $\omega = 1$. In addition, all experiments have been run with $\delta_s = 10$ and $\delta_c = 0.1$, as recommended by (Giraud et al., 2017) to favor spatial consistency.

4.6.3 Analysis on different layers

Our SFE and SFM blocks support any CNN features map dimensions, so choosing to work with one or coupling many of these features maps depends mostly on the application. In this experiment, we analyze the effects of using separately each of the first three feature map outputs for the color transfer application.

From the different columns of Figure 4.12 we can retain that each layer focuses on different aspects of the image, resulting in color variations of the same target image. Specifically, in the first row (house image), we can see that the deeper layer (layer 3) focuses on the grass color while the second layer focuses on the mountain color. In the second and third rows, layers 2 and 3 bring stronger colors from the reference image, but the results are still unrealistic. In the first layer, the recovered image seems more natural; however, most of

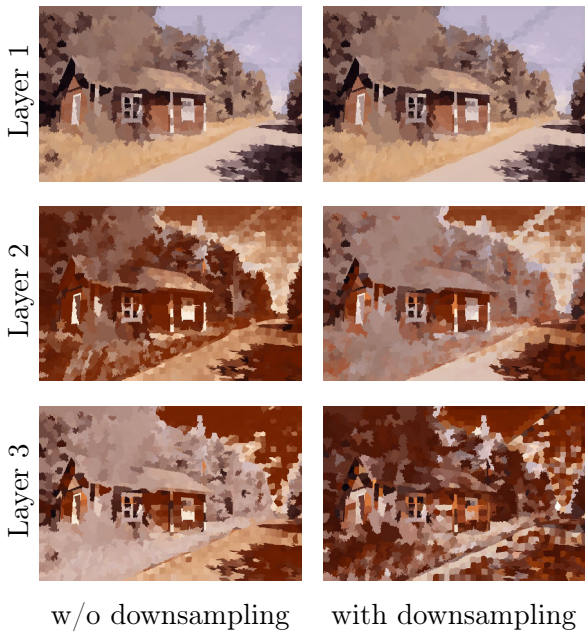


Figure 4.11: Effect on direct super-features matching (*i.e.,* before color fusion) using high-resolution feature maps (without downsampling) and using low-resolution feature maps (with downsampling). Target and reference images are presented in Figure 4.12.

the colors transferred from the reference image are opaque or not transferred.

Finally, we decided to combine all three layers as each of them brings important feature information to achieve pleasant and realistic images for this color transfer application.

4.7 Evaluation

We compare our method against three approaches: (Pitié et al., 2007) which proposes an automated color transfer based on color distributions; (Lee et al., 2020b) which implements a color transfer approach based on color histogram analogy using a deep neural network; and (Giraud et al., 2017) which implements the color fusion framework by leveraging on its proper superpixel decomposition. All three mentioned approaches have been considered state-of-the-art in color transfer, and have open-source codes for a fair comparison. Each method has been run with its default parameters.

Results comparing the three methods are shown in Figures 4.13, 4.14 and 4.15. Overall, our results (last column) have more visually pleasant colors and consistency in image texture, providing more realistic color transfers with respect to the other methods. Figure 4.13 shows that our approach correctly matches and transfers natural colors from indoor images, avoiding color bleeding (blue color on the wall) as shown in (Giraud et al., 2017), (Lee et al., 2020b) and partially in (Pitié et Kokaram, 2007) results. For outdoor images shown in Figure 4.14, we observe that (Pitié et al., 2007) and (Lee et al., 2020b) can suffer from over-saturation of the illumination on some of their results (first, fourth, sixth images). Although this problem does not appear in (Giraud et al., 2017) some of its

Figure 4.12: Results of our method using each of the three layers separately with $\tau = 0.015$.

results present visible unnatural effects on sky colors such as a halo effect (first image) and yellowish marks (fifth, sixth images). Our results for outdoor images overcome these issues thanks to the robust matching on high-resolution superpixel deep features, which ensures color consistency and spatially coherent colors across the resulting images. Figure 4.15 presents images with no background (studio shooting like images). In this case, results by (Pitié et al., 2007) and (Lee et al., 2020b) show unnatural color effects around the bottle and background. On the other hand, (Giraud et al., 2017) approach and ours achieve pleasant color results without over-saturation nor artifacts on the resulting image. Lastly, our method correctly transfers colors to the statue image resulting in the most visually satisfying and realistic results with regards to all compared methods.

4.8 Conclusion and future works

In this chapter, we proposed the novel super-features encoding block (SFE) and the super-features matching (SFM) block that successfully encodes and matches high-resolution deep learning features from different images using superpixel decomposition. We validate these two blocks on the problem of color transfer; for doing that, we update the color fusion framework initially proposed by (Giraud et al., 2017) to consider our attention map, which provides texture and color knowledge from the reference image onto the final color transfer step. Finally, our method achieves more visually consistent and realistic results in comparison to the three state-of-the-art methods considered.

In the next Chapter 5, we will extend and reuse most of the super-attention block methodology in a learnable exemplar-based colorization framework.

Figure 4.13: Comparison of color transfer results on indoor images. We compare our method with three different state-of-the-art approaches: (Pitié et al., 2007) color distribution grading, (Giraud et al., 2017) color fusion based on superpixel representation and, (Lee et al., 2020b) deep learning-based color histogram analogy.

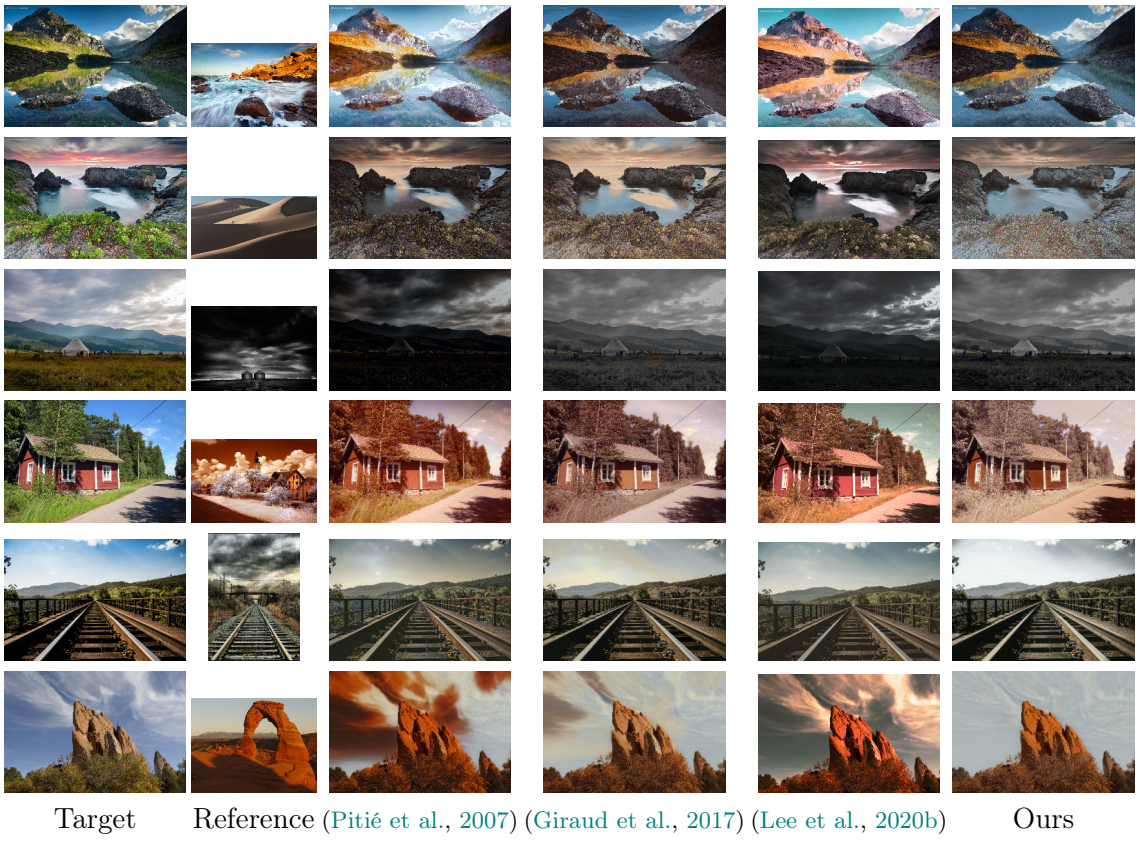


Figure 4.14: Comparison of color transfer results on outdoor images.

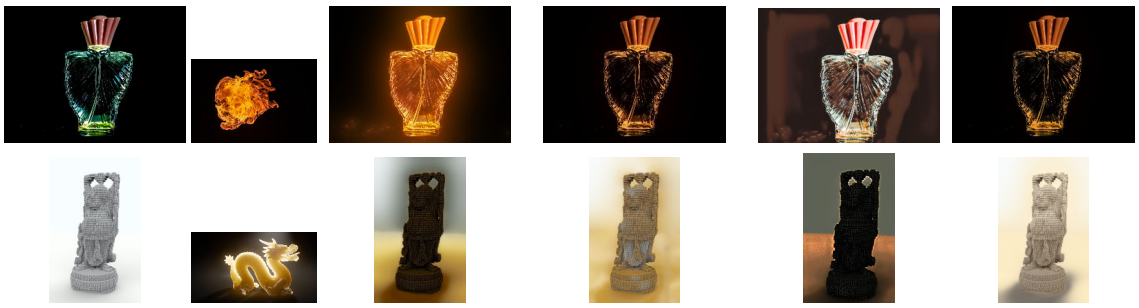


Figure 4.15: Comparison of color transfer results on images with no background.

Chapter 5

Super-Attention for exemplar image colorization

Table of contents

Summary

In image colorization, exemplar-based methods use a reference color image to guide the colorization of a target grayscale image. In this chapter, we present a deep learning framework for exemplar-based image colorization which relies on attention layers to capture robust correspondences between high-resolution deep features from pairs of images. Super-attention blocks introduced in Chapter 4 can learn to transfer semantically related color characteristics from a reference image at different scales of a deep network. Our experimental validations highlight the interest of this approach for exemplar-based colorization. We obtain promising results, achieving visually appealing colorization and outperforming state-of-the-art methods on different quantitative metrics.

Related publications

[Carrillo *et al.* 2022] **H. Carrillo,** M. Clément, A. Bugeau. "Super-attention for exemplar-based image colorization." *Asian Conference on Computer Vision (ACCV)*, 2022.

Contributions

The main contributions in this chapter are:

- A new end-to-end deep learning architecture for exemplar-based colorization improving results over state-of-the-art methods.
- A multiscale attention mechanism based on superpixels features for reference-based colorization.
- A strategy for choosing relevant target/reference pairs for the training phase.

5.1 Introduction

In this chapter, we extend the super-attention mechanism presented in Chapter 4 to an exemplar image colorization framework using a deep learning architecture. As detailed in Chapter 1, colorization is the process of adding plausible color information to grayscale images. Ideally, the result must reach a visually pleasant image, avoiding possible artifacts or improper colors. However, colorization is an inherently ill-posed problem, as multiple suitable colors might exist for a single grayscale pixel, making it challenging. To solve this issue, exemplar-based methods consider a reference color image from which colors can be transferred to a target grayscale image. Nevertheless, when semantic similarities do not exist between the reference image and input image, the efficacy of these exemplar-based methods highly decreases. Another category of methods, named colorization by learning, brings fairly good colors to grayscale images by leveraging color priors learned from large scale datasets. Nonetheless, this type of method lacks user's decision. Therefore, there is an interest in coupling information from large scale datasets and associating it with semantic information of a reference image to no longer depend on naive pixel-wise matching.

To address the challenge mentioned above, (Zhang et al., 2019) proposed an attention mechanism for image colorization, mainly by calculating non-local similarities between different feature maps (input and reference images). However, attention mechanisms come with a complexity problem, namely a quadratic scaling problem, due to its non-local operation. This is why it has to be applied to features with low dimensions. On the other hand, low-resolution features lack detailed information for calculating precise and robust pixel-wise similarities. For instance, low-resolution deep features mainly carry high-level semantic information related to a specific application (*i.e.,* segmentation, classification) that can be less relevant for high-resolution similarity calculation or matching purposes. In addition, this high-resolution information is essential to retrieve colors of small specific objects.

This chapter proposes a model that enables coupling exemplar-based approaches and colorization by learning. Our proposal relies on similarity calculation between high resolution deep features. These features contain rich low-level characteristics, which are important in the colorization task. To overcome the complexity issue, we extend to the colorization task the super-attention block presented in Chapter 4 that performs non-local matching over high-resolution features based on superpixels.

The rest of this chapter is as follows: in Section 5.2, we examine how the attention mechanism has been used in exemplar-based image colorization. Then, in Section 5.3, we introduce our general architecture and analyze how to use the super-attention block in different parts of the architecture. Next, in Section 5.4, we present a new way to find target-reference pairs to train our network. In Section 5.6, we compare our approach to state-of-the-art methods on various quantitative metrics, and our results show that our approach produces visually appealing colorizations. Finally, in the appendix B.1, we provide more details of our architecture, along with additional comparison results to non-learning methods as well as results on archive images. Additionally, we introduce another variant of our proposal which incorporate the histogram loss on our final method.

5.2 Attention in exemplar image colorization

As presented in Section 2.6, attention mechanisms have become a popular topic for computer vision tasks, especially within transformer architectures. These networks use attention layers as their main block. These attention layers learn to compute similarities, called attention maps, between input features using a non-local matching operation.

Recent works in exemplar-based image colorization using attention mechanisms have been shown to be particularly effective for this task by improving visual coherence between grayscale images and colored examples. In (Zhang et al., 2019), they proposed a framework that consists of two sub-modules, the correspondence subnet, and the colorization subnet. The first receives features for both the input image and the reference image. By means of a cross-attention mechanism, it captures long-range similarities between the two different features. This allows the network to selectively focus on the most relevant features for aligning the input and reference images. As a result, the first subnet outputs a pre-colored version of the grayscale image. The second sub-module uses the pre-colored image and the similarity map in order to guide the final generation of the colors. However, this type of dual architecture framework is highly complex in training. In addition, the resulting colors mainly depend on the correct alignment of the pre-colored images. Lately, (Yin et al., 2021) proposed an approach that utilizes three types of color inputs: semantic colors associated with objects in the reference image, the global color distribution such as tones of the reference image, and color information from the database. To address the challenge of considering these three types of colors simultaneously, the approach relies on the attention operation to constrain the searching process in the database by using the color distribution of the reference image prior to the network. This idea ensures that the colorized image is forced to have the colors from the reference image and, at the same time, keep a natural look, even when the reference image and the grayscale image are semantic independent. However, it is important to notice that this approach relies on the global histogram of the reference image, which takes colors that might not be pleasant for the final colorization. Finally, (Blanch et al., 2021) presents a fast-end-to-end architecture for exemplar colorization that improves existing methods and significantly decreases the complexity of the attention mechanism. The method implements the axial attention mechanism to guide the transfer of the color's characteristics. In the axial attention mechanism, instead of calculating the non-local operation for the whole feature map, the axial applies the attention sequentially, first on the rows and then on the columns, to reduce the complexity of the task. Even though the method achieves a lower complexity on the attention block, it is still not possible to do it on high-resolution features. This complexity issue constrains the attention mechanism to be only performed at low-resolution features. In Section 5.3.2, we extend the super-attention mechanism presented in Chapter 4 into the exemplar colorization task, this is in order to not only be able to handle features at low resolutions but also at high resolutions in a deep neural network architecture.

5.3 Colorization framework

Our objective is to colorize a target grayscale image T by taking into consideration the characteristics of a color reference image R. Let $T_L \in R^{H\times W\times 1}$ be the luminance component of the target, specifically the channel L from the Lab color space, and $R_{Lab} \in R^{H\times W\times 3}$

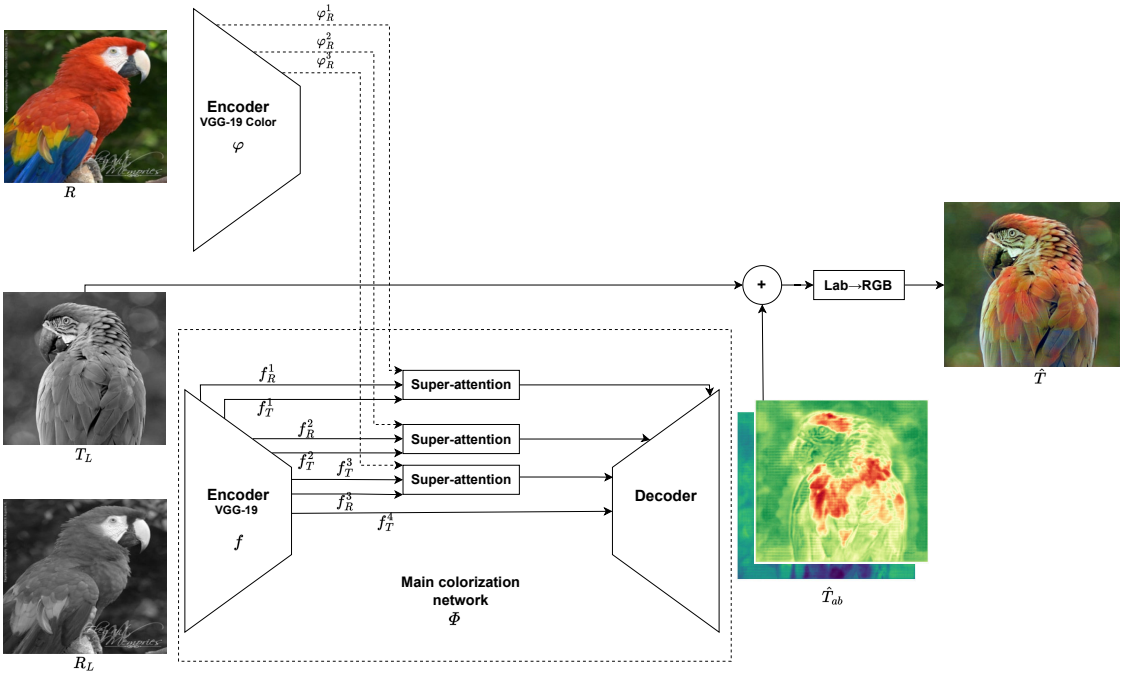


Figure 5.1: Diagram of our proposal for exemplar-based image colorization. The framework consists of two main parts: 1) the color feature extractor φ extracts multi-level feature maps φ_R^ℓ from a color reference image R, and 2) the main colorization network Φ, which learns to map a luminance channel image T_L to its chrominance channels $\hat{T}_{ab}$ given a reference color image R. This colorization guidance is done by super-attention modules, which learn superpixel-based attention maps from the target and reference feature maps from distinct levels f_T^ℓ and f_R^ℓ respectively.

the color reference image in CIELAB color space. In this work, we choose to work with the luminance-chrominance CIELAB color, as it is perceptually more uniform than other color spaces (Connolly et Fleiss, 1997).

In order to colorize the input image, we train a deep learning model Φ that learns to map a grayscale image to its chrominance channels (ab) given a reference color image:

$$\hat{T}_{ab} = \Phi(T_L \,|\, R). \tag{5.1}$$

Our proposed colorization framework is composed of two main parts: an external feature extractor φ for color images, and the main colorization network Φ, which relies on super-attention blocks applied at different levels (see Figure 5.1). The main colorization network Φ is based on a classical Unet-like encoder-decoder architecture (Ronneberger et al., 2015), with the addition of the super-attention blocks (see Chapter 4), which allows transferring color hints from the color reference image to the main colorization network. Next, this network predicts the target's chrominance channels $\hat{T}_{ab}$. And, as a final stage, target luminance and chrominance channels are concatenated into $\hat{T}_{Lab}$ and then converted to the RGB color space $\hat{T}$ using Kornia (Riba et al., 2020).

5.3.1 Main colorization network

The main colorization network Φ aims to colorize a target grayscale image based on a reference image when semantic-related content appears, or pulling back to the learned model when this relation is not present in certain objects or regions between the images. The colorization network receives target T_L and reference R_L as input images to obtain deep learning feature maps f_T^ℓ, f_R^ℓ from the ℓ^{th} level of the architecture. In the same sense, the reference color image R is fed to the color feature extractor, which is a frozen pre-trained VGG19 encoder that retrieves multiscale feature maps φ_R^ℓ. Specifically, feature maps are extracted from the first three levels of the encoder. Then, all extracted features are fed to the super-attention blocks, where a correlation is computed between target and reference encoded features. Next, the content is transferred from the reference features to the target by multiplying the super-attention map and the color reference features. Then, the color features coming from the super-attention modules are transferred to the future prediction by concatenating them to the decoder features. Finally, the decoder predicts the two (ab) chrominance channels $\hat{T}_{ab}$ that are then concatenated to the target luminance channel T_L, then the prediction is converted from CIELAB color space to RGB color space to provide the final RGB image $\hat{T}$. However, this conversion between color spaces arises clipping problems on RGB values as in certain cases the combination of predicted Lab values can be outside the conversion range.

5.3.2 Super-attention as color reference prior

The super-attention block injects colors priors from a reference image R to the main colorization network Φ. This block relies on multi-level deep features to calculate robust correspondence matching between the target and reference images. Specifically, the super-attention block is divided into two parts: the super-features encoding layer and the super-features matching layer. The super-features encoding layer provides a compact representation of high-resolution deep features using superpixels. For the colorization application, we focus on feature maps extracted at the first three levels of the architecture, as they provide a long range of high and low-level features that suit content and style transfer applications as mentioned in Chapter 4. Figure 5.2 depicts the diagram of the super-attention block where f_T^ℓ, f_R^ℓ and φ_R^ℓ are feature maps from the encoder f and the encoder φ at level ℓ of T_L, R_L and R respectively. In summary, this encoding part makes possible operations such as correlation between high-resolution features in our colorization framework. After the encoding block, we have a layer that computes the correlation between encoded high-resolution deep-learning features, which we call the super-features matching (SFM). This layer is inspired by the classic attention mechanism (Zhang et al., 2019) on images to achieve a robust matching between target and reference super-features. However, contrary to our based attention block proposed in Chapter 4, our attention map is learned by the model. The attention map is then calculated using the Equation (4.3) at each at layer ℓ. Then, the final output (f_{TR}) of the super-attention block is calculated as follows:

$$f_{TR}^\ell = unpooling(\mathcal{A}^\ell \cdot \Phi_R^\ell), \tag{5.2}$$

where f_{TR}^ℓ is the weighted feature map with the same size of f_{T_L} and which includes color characteristics from the reference images.

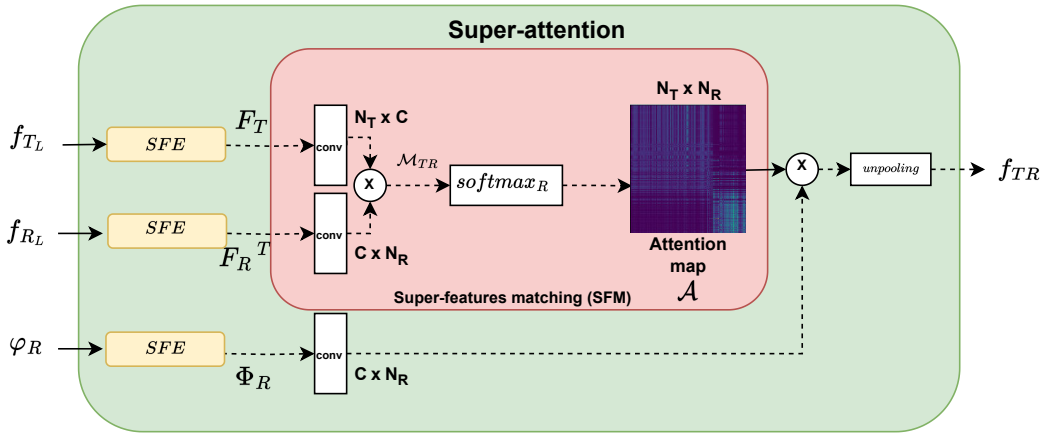


Figure 5.2: Diagram of our super-attention block. This layer takes a reference luminance feature map f_R, reference color feature map φ_R and a target luminance feature map f_T, as an input, and learns an attention map at superpixel level by means of a robust matching between high-resolution encoded feature maps.

In terms of complexity, the super-feature encoding approach allows to overcome the quadratic complexity problem in the computation of standard attention maps. Indeed, instead of computing attention maps of quadratic size in the number of pixels, our super-attention maps are computed on a much smaller number of superpixels (*i.e.,* linear in the number of pixels). More details about this complexity reduction can be found in Chapter 4.

To illustrate the use of this block between target and reference images, Figure 5.3 shows some examples of matching using the super-attention block at the first level of the architecture. Mainly, it shows that for one superpixel from the target feature maps, the learned attention map successfully looks for superpixels on the reference feature maps with similar characteristics to the reference image.

5.4 Training framework

5.4.1 Training losses

Designing the loss function in any deep learning model is one of the key parts of the training strategy. In the classical case of automatic colorization, one would like to predict $\hat{T}_{ab}$ by reconstructing the colors from the ground-truth image T_{ab}. But, this idea could not work within exemplar-based colorization as the predicted $\hat{T}_{ab}$ colors should take into account color's characteristics from a reference image $\hat{T}_{ab} = \phi(T_L \,|\, R)$. Then, the goal is to guarantee an accurate and well-grounded transfer of color characteristics to the target from the reference.

In this work, we propose a coupled strategy of two loss terms, L1 smooth and LPIPS. The first encourages pixel similarities between the ground-truth chrominances (T_{ab}) and the predicted chrominances ($\hat{T}_{ab}$). The second loss helps to encourage the perceptual similarity between the features from the ground-truth target image (T) and features from the predicted one ($\hat{T}$). More details on both losses can be found in Section 2.8. Finally, these two loss terms, $L_{1_{smooth}}$ and L_{LPIPS}, are summed by means of different fixed weights

Target superpixel | Reference superpixel guidance | Attention map

Figure 5.3: Example of guidance maps from the super attention mechanism. First column highlights one superpixel in the target; second column, the reference in which the lightness of the superpixels are scaled according to the computed attention map; third column: the attention map with in red the row corresponding to the target superpixel and that is used to generate the second column.

which allow to balance the total loss. The joint total loss used on the training phase is then:

$$L_{total} = \lambda_1 L_{1_{smooth}} + \lambda_2 L_{LPIPS}, \tag{5.3}$$

where λ_1, λ_2 are fixed weights for each of the individual losses.

Notice that some previous exemplar-based colorization methods proposed to add an additional histogram loss to favor color transfer (Lu et al., 2020; Yin et al., 2021; Blanch et al., 2021). However, as we do not want to enforce a complete transfer of all colors from the reference, but only the ones that are relevant to the source image, we have decided not to use it in our final model. We provide additional experiments using the histogram loss in the supplementary material B.1.

5.4.2 Implementation details

For the training phase, we set the weights of two terms of the loss to $\lambda_1 = 10$, $\lambda_2 = 0.1$. These values were chosen empirically to obtain a good balance between the $L_{1_{smooth}}$ reconstruction term and the LPIPS semantic term. We train the network with a batch size of 8 and for 20 epochs. We use the Adam optimizer with a learning rate of 10^{-5} and $\beta_1 = 0.9$, $\beta_2 = 0.99$. Finally, our model is trained with a single GPU NVIDIA RTX 2080 Ti and using PyTorch 1.10.0.

For the super-attention modules, superpixel segmentations are calculated on the fly using the SLIC algorithm (Achanta et al., 2012). Multiscale superpixels grid are calculated using downsampled versions of the grayscale target T_L and reference R_L images. Note

that different superpixel algorithms could be used. These downsamplings are performed to match the size of the feature maps from the first three levels of the encoder f.

5.5 Dataset and references selection

We train the proposed colorization network using the COCO dataset (Lin et al., 2014). Unlike ImageNet, this dataset proposes a smaller quantity of images but with more complex scene structures, depicting a wider diversity of objects classes. Additionally, the dataset provides the segmentation of objects, which we rely on for our target and reference images pairs strategy. In detail, this dataset is composed of 100k images for training and 5k images for validation. For the training procedure, we resize the images to the size of 224×224 pixels.

Another key aspect of the training strategy in exemplar-based methods is to find a suitable semantic reference to the target image. To build the target and reference pairs of images, we took inspiration from (He et al., 2018) to design our ranking of reference images. There, they proposed a correspondence recommendation pipeline based on grayscale images. Here, our approach focuses on searching target-reference matches from a wide variety of segmented objects as well as natural scenes images. Our proposal ranks four images semantically related to each target image. First, to increase the variety of reference images within a category, we extract each meaningful object whose size is greater than 10% of the size of the current image. To retrieve the image objects, we use the segmentations provided by the dataset. Second, we compute semantically rich features from the fifth level of a pre-trained VGG-19 (Simonyan et Zisserman, 2015) encoder φ_T^5 and φ_R^5. Next, for each target, reference images are ranked based on the L_2 distance between these pre-computed features. Finally, during training, target-reference pairs of images are sampled using a uniform distribution with a weight of 0.25 by randomly choosing either itself (*i.e.,* the ground-truth target image is used as the reference) or the other top-4 semantically closest references.

Figure 5.4 shows examples of target-reference matching based on our proposal. The target images are presented in the first column, and the following columns represent its corresponding reference based on the nearest L_2 distance between feature maps. The second column (Top 1) shows the references most semantically-related to the target, while the last column (Top 4) shows the references the least semantically-related to the target.

5.6 Evaluation

In this section, we present quantitative and qualitative experimental results to illustrate the interest of our proposed approach. First, we propose an analysis of our method with an ablation study comparing different architectural choices and training strategies. Then, we compare our results to three state-of-the-art exemplar-based colorization approaches.

5.6.1 Analysis of the method

We start by analyzing quantitatively and qualitatively certain variants of our proposed colorization framework. Within this ablation study, we analyze two variants and our final

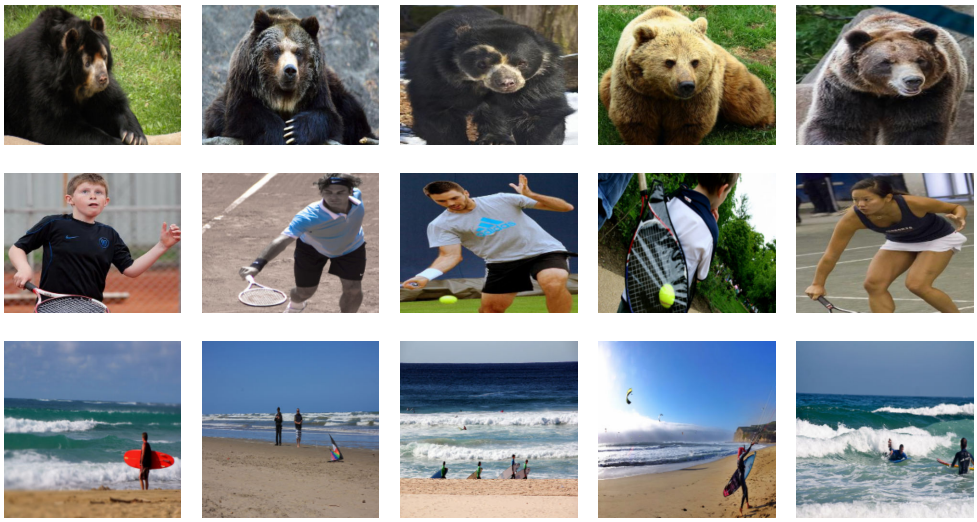


Figure 5.4: Illustration of our reference selection method. The first column shows example target images, and the next four columns show the closest references in the dataset, in decreasing order of similarity.

proposal. The first one is a baseline colorization model without using references. It uses the same generator architecture without any attention layer. The second variant is our framework using a standard attention layer in the bottleneck of the architecture instead of our super-attention blocks. Finally, the third model is our proposed exemplar-based colorization framework, which includes the use of references with super-attention blocks in the top three levels of the network.

Evaluation metrics. To evaluate quantitatively the results of these different methods, we used three metrics. Two metrics compare the result with the ground-truth color image, and a third metric compares the prediction of colors with respect to the reference color image (see more details in Section 2.7). The first one is the structural similarity (SSIM) metric (Wang et al., 2004), which analyzes the ability of the model to reconstruct the original image content. The second one is the learned perceptual image patch similarity (LPIPS) metric (Zhang et al., 2018b) which correlates better with the human perceptual similarity. These first two metrics (SSIM and LPIPS) evaluate the quality of the output colorization compared to the ground-truth. The third metric employed is the histogram intersection similarity (HIS) (Isola et al., 2017) which is computed between the predicted colorization and the reference image. The goal of this third metric is to evaluate if the colors from the reference have been correctly transferred to the prediction. However, unless the ground-truth and reference share the same color distribution, these metrics are inherently contradictory (*i.e.,* good HIS would lead to bad SSIM/LPIPS). In this work, we do not necessarily want to fully transfer the colors from the reference image. Instead, we view the reference as colors hints that can be used by our network to predict a more plausible

Figure 5.5: Colorization results obtained using different variants of our colorization framework. In the first two lines, the chosen color reference is different from the target to be colorized (but semantically similar). In the last two lines, the reference correspond to the actual color version of the target image. These results allow to assess the ability of our method to transfer relevant colors from different types of reference images.

colorization.

Results and ablation study. The results displayed in Table 5.1 are the averages of the metrics calculated using the evaluation set of 5000 pairs of target-reference images from the COCO validation set. From this Table, we can observe that our full colorization framework achieves the best SSIM and LPIPS scores in comparison with the other variants of the model, suggesting that the use of reference images with super-attention blocks helps in getting better colorization results. We can also notice that the full model achieves a better ΔHIS than the standard attention variant. This suggests that, instead of forcing a global transfer of colors from the references, our model is capable of picking specific and plausible colors from the reference images to generate better colorization results. Note that the ΔHIS is not reported for the first line, as this variant does not use any reference image.

In addition to this quantitative evaluation, in Figure 5.5 we present a qualitative comparison of our method and its ablation variants. From the results of the first and second row, we can see that the method with standard attention proposes a more global transfer of colors, leading to the appearance of brighter colors related to the reference. However, this also causes unnatural colorization around the carriage in the first example (*i.e.,* green, orange) and on the hand of the baseball player in the second example. On the other hand,

Table 5.1: Quantitative analysis of our model. SSIM and LPIPS metrics are calculated with respect to the target ground-truth image. ΔHIS is the absolute difference between the histogram intersection of the ground-truth target and the reference and the histogram intersection of the predicted target and the reference.

Method	**SSIM↑**	**LPIPS↓**	**ΔHIS↓**
Ours without reference	0.920	0.164	-
Ours with standard attention	0.921	0.172	0.257
Ours	**0.925**	**0.160**	**0.234**

Table 5.2: Quantitative comparison with three state-of-the-art exemplar based-colorization methods. Ours correspond to our full model with references and super-attention blocks.

Method	**SSIM ↑**	**LPIPS ↓**	**ΔHIS↓**
XCNET (Blanch et al., 2021)	0.867	0.270	0.276
Deep Exemplar (He et al., 2018)	0.894	0.241	0.242
Just Attention (Yin et al., 2021)	0.896	0.239	0.262
Ours	**0.925**	**0.160**	**0.234**

our method with super-attention block overcomes this issue and provides a more natural colorization, using specific colors from the reference. The last two lines show the effectiveness of our method when the reference is semantically identical to the target. Indeed, it shows that color characteristics from the reference image are passed with high fidelity, as compared to the variant without reference where it results in opaque colors.

Overall, our framework achieves visually pleasant colorization results, and the transfer of color characteristics from the reference is done in specific areas of the target image, thanks to our super-attention blocks. When the color reference and the super-attention modules are not included in the framework, we observe a decrease of quality in the resulting colors (*i.e.,* averaging of colors and/or wrong colorization).

5.6.2 Comparison with other exemplar-based colorization methods

To evaluate the performance of our exemplar-based colorization framework, we compare our results quantitatively and qualitatively with three other state-of-the-art exemplar-based image colorization approaches (He et al., 2018; Yin et al., 2021; Blanch et al., 2021). In order to fairly compare these methods, we run their available codes for the three approaches using the same experimental protocol and the same evaluation set as in the previous section.

A quantitative evaluation is proposed to compare the three state-of-the-art methods and our framework. For comparing the methods, we again use SSIM, LPIPS, and ΔHIS. As shown in Table 5.2 our colorization framework preserves stronger perceptual structural information from the original target image with respect to all three state-of-the-art methods. LPIPS score measures the perceptual similarity between colorized results and target ground-truth. Our method surpasses (Blanch et al., 2021), (Yin et al., 2021), (He et al., 2018) significantly. Finally, our method achieves a smaller ΔHIS with respect to all compared state-of-the-art methods.

Figure 5.6 shows colorization results from (Blanch et al., 2021), (Yin et al., 2021), (He et al., 2018) and our approach. For the first two images, our proposal produces a more

visually pleasant colorization and provides more natural colors than the other methods. In contrast, the results for the first two images of (Yin et al., 2021) and (Blanch et al., 2021) shows a high amount of color bleeding, mainly on top of the baseball player and on the background, as their approach captured global color distribution from the reference image. For the fourth image, methods (Yin et al., 2021) and (Blanch et al., 2021) failed to transfer the color information from the red jacket. Conversely, (He et al., 2018) and our approach did transfer the jacket's color. Next, in the results for the fifth image, we observe unnatural colors as the green water for methods (He et al., 2018) and (Blanch et al., 2021). In contrast, our method and (Yin et al., 2021) achieves a natural colorization by transferring colors from the reference. Finally, for the results of the last image, (He et al., 2018) encourages the transfer of the colors from the reference image better than the other methods; however, the final colorization results seem unnatural. For the same image, the colorization results on method (Blanch et al., 2021) and ours seem to be the right balance between color transfer and naturalness from the learned colorization model.

5.7 Conclusion and future works

In this chapter, we have proposed a novel end-to-end exemplar-based colorization framework. Our framework uses a multiscale super-attention mechanism applied on high-resolution features. This method learns to transfer color characteristics from a reference color image, while reducing the computational complexity compared to a standard attention block. In this way, we coupled into one network both semantic similarities from a reference and the learned automatic colorization process from a large dataset. Our method outperforms quantitatively state-of-the-art methods, and achieves qualitatively competitive colorization results in comparison with the state-of-the-art methods.

In the exemplar-based methods, the correct transfer of plausible colors plays a key role in guiding the final colorization results. In our method, we introduce semantically-related colors characteristics from the reference by means of the super-attention block. This block let us find correspondences at different levels of the architecture, resulting in rich low-level and high-level information. While our attention mechanism successfully retrieves relevant color features, simply concatenating them may not effectively enforce strong guidance from the reference color. Further research should focus on developing a more refined transfer scheme to enhance the model's ability to incorporate reference color information. For that, in Chapter 6, we propose two solutions; first, we will consider adding the super-attention blocks in the encoder part of the architecture instead of the skip connections. Second, we will ensure the transfer of specific objects by allowing the user to provide segmentation masks inside our model.

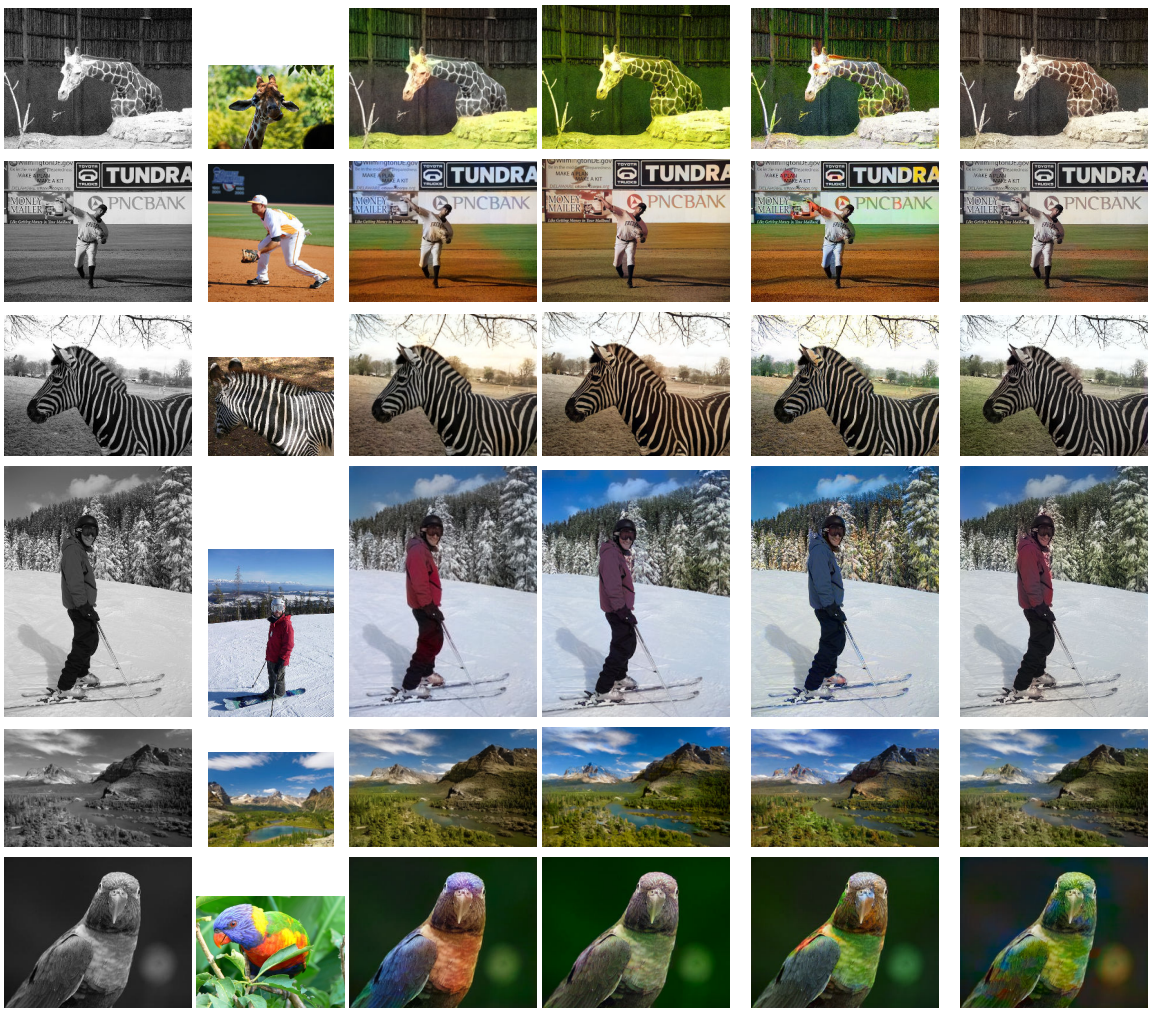


Figure 5.6: Comparison of our proposed method with different reference-based colorization methods: Deep Exemplar (He et al., 2018), Just Attention (Yin et al., 2021) and XCNET (Blanch et al., 2021).

Chapter 6

Masked super-attention for object guided image colorization

Table of contents

Summary

In this chapter, we propose a novel end-to-end deep learning framework for exemplar-based colorization that integrates user-provided object masks. We aim to guide the colorization on specific and meaningful objects rather than a full reference image. Our framework consists of an encoder-decoder generator architecture. The core module of the encoder is our proposed masked super-attention. This multiscale object-specific attention mechanism improves the ability to transfer color characteristics from the user's selected objects. In addition, we introduce a strategic method for selecting pertinent target/reference image pairs at the object-level. To comprehensively evaluate the effectiveness of our proposed approach, we conduct a complete evaluation of both full-level and object-level images. Finally, our framework achieves colorful and visually pleasant colorization and surpasses state-of-the-art methods on different quantitative metrics.

Related publications

[Carrillo *et al.* 2023] **H. Carrillo,** M. Clément, A. Bugeau. "Exemplar-based image colorization using object-guided attention" *Preprint submitted to: International Journal of Computer Vision (IJCV)*, 2023.

Contributions

Our main contributions are as follows:

- We develop a new end-to-end deep learning framework for exemplar-based colorization capable of incorporating object masks provided by the user.
- We integrate the super-attention block within the encoder of the network architecture instead of within the skip connections as in Chapter 5.
- We propose a multiscale object-specific attention mechanism that utilizes masked superpixel features for reference-based colorization, called masked super-attention.
- We leverage a strategy for selecting relevant object pairs in target/reference images.
- We conduct a complete and comprehensive evaluation of our approach at both the full-image level and object-level images, comparing it to state-of-the-art methods.

6.1 Introduction

Image colorization results from previous methods can be unsatisfactory for real-world applications. One of the reasons is that they are inefficient at exploiting semantic color information, mainly when two or more objects are presented in the target or reference images.

To address the weaknesses identified in the previous methods, and keeping the same idea from Chapter 5. We propose a guided attention mechanism using a segmentation map with an exemplar-based colorization method to better guide the colorization on specific, meaningful objects rather than a whole reference image. We suggest using segmentation masks to enhance the quality of image colorization in the following ways. First, the segmentation masks identify visually significant regions within the image. This allows the colorization framework to prioritize important objects rather than, for example, the background. Typically, objects of interest are more colorful, while backgrounds tend to be dominated by green and blue hues, such as sky, trees, and water. Our approach decreases the probability that the framework is biased toward the background colors. Secondly, segmentation masks help localize specific objects, highlighting semantically relevant regions with distinct boundaries. The previous is advantageous for colorization networks as it reduces color bleeding artifacts. Finally, adding segmentation as input is relatively easy for the user.

The rest of this chapter is organized as follows: in Section 6.2 we review how segmentation is used in the image colorization task. Next, in Section 6.3 we present our proposal called *Masked super-attention* which provides the user object-specific interaction within the colorization process. Then in Section 6.4 we show the training process of our proposal as well as dataset selection. Finally, in section 6.5 we present a comprehensive evaluation of our proposed approach, for that, we conduct a complete evaluation of both full-level and object-level images.

6.2 Segmentation in image colorization

Image segmentation involves dividing an image into multiple regions or segments, each of which corresponds to a specific object. These approaches range from classical methods such as thresholding (Sezgin et Sankur, 2004; Anjos et Shahbazkia, 2008), clustering (Achanta et al., 2012), and edge detection (Lindeberg et Li, 1997) to deep learning-based methods. They have been successfully applied to image generation (Singh et al., 2019), image-to-image translation (Shen et al., 2019; Ma et al., 2018), and semantic image synthesis (Wang et al., 2018a). Segmentation has been used for the cartoon colorization task, starting by Sykora et al.(Sýkora et al., 2004), which presents an exemplar-based colorization technique that uses unsupervised image segmentation joined with patch-based sampling to transfer colors from a reference colorized image. Extensions have been made to natural image colorization. Irony et al. (Irony et al., 2005) proposed a method for colorizing grayscale images using a segmented reference image. It considers the higher-level context of each pixel, resulting in colorization with a higher degree of spatial consistency through the mean-shift segmentation algorithm (Comaniciu et Meer, 2002). Later, Gupta et al. (Gupta et al., 2012) used superpixels to improve the colorization process by speeding up the task and increasing spatial coherence, which was further improved in (Li et al., 2017b) by taking

into account intensity, texture, and semantic features. Recent approaches such as Zhao et al. (Zhao et al., 2018) coupled neural networks and pixel-level object semantics to guide colorization and mitigate the context confusion issues. Recently, Su et al. (Su et al., 2020) proposed a method to improve image colorization with multiple objects, it uses an object detector to extract object instances, then employs a neural network to capture object-level features for later combining them with full-image features using a fusion module to predict accurate colors. Previous methods leverage fully automatic colorization methods with segmentation, meaning human intervention is unavailable for the colorization or segmentation tasks. The previous leads to issues where the automatic segmentation mask is inaccurate, or none of the objects were identified correctly, causing visible artifacts such as washed-out colors or bleeding across object boundaries.

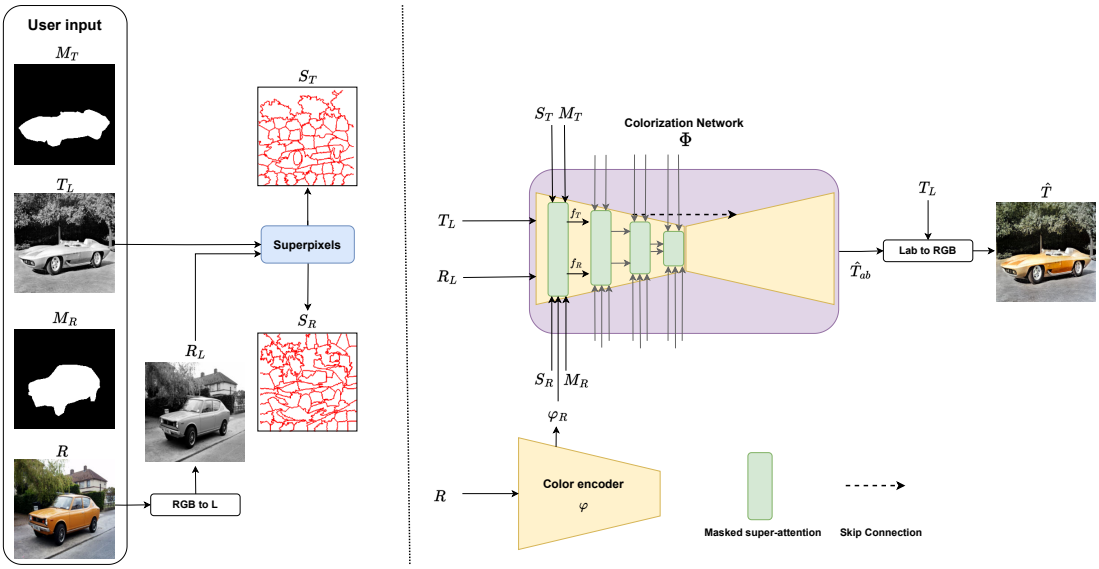


Figure 6.1: Our proposal. The colorization framework includes two main parts that jointly handle the semantic correspondence and chromatic propagation between the input images. First, The color feature extractor φ extracts multi-level feature maps from a color reference image R. The main colorization network Φ that learns to map a luminance channel image T_L to its chrominance channels $\hat{T}_{ab}$ given color characteristics from image R. The colorization guidance is done by the masked super-attention modules. These attention layers receive feature maps from distinct levels f_T, f_R and their respective superpixel grids S and a segmentation mask M from the target and reference images.

6.3 Object specific interaction using masked super-attention in exemplar image colorization

Our objective is to add feasible colors to a grayscale image using a color reference image. We aim to apply reference colors to semantically related content in the target image while creating a plausible colorization for regions or objects without such relationships. This goal poses two challenges. First, measuring the semantic connection between reference and target images is particularly challenging when the reference and the target images are

partly semantically different. Secondly, even if we have good similarity metrics, selecting appropriate reference colors and effectively propagating them through the target image remains a difficult task.

We propose an end-to-end colorization network framework to address the previous two challenges. This framework includes two main parts that jointly handle the semantic correspondence and chromatic propagation between the input images. By doing so, we can break down the colorization task into two distinct subproblems instead of a highly complex one. Then, an external feature extractor is designed to extract color features from the reference color image. The main colorization network uses the original super-attention modules (Chapter 5) in combination with our proposal on masked features at various levels of the encoder to guide the final colorization. The main colorization network uses a traditional encoder-decoder architecture similar to U-net, incorporating our proposed superpixel-level masked attention blocks. These blocks enable the transfer of color characteristics from the reference image to the main colorization network, allowing a more accurate and robust colorization. An overview of our proposal is depicted in Figure 6.1.

For training, we use a two-phase sequential approach that involves first, training the framework without any segmentation (*i.e.,* just pairs of full target-reference images), and then, pre-loading weights from the previous step and re-training with the masked super-attention block.

Masked super-attention. In addition to colorizing grayscale images from full reference images, our colorization framework can also be used to colorize specific objects within an image. This is done using a segmentation mask M_s to identify the object of interest. Once the object of interest has been specified, a super-attention mechanism (proposed in Chapter 5) is modified to be applied only on the superpixels of the mask. Mainly, this masked super-attention mechanism, learns to find similar object-to-object characteristics between a reference and a target image.

The masked super-attention block has two parts: the masked super-features encoding layer (MSFE) and the super-features matching layer (SFM). The MSFE creates a compact representation of deep features using superpixels constrained to a segmentation mask. The SFM layer then matches these compact representations to find the most similar features between the target and reference object images. In the MSFE, we use features from all four levels of the architecture, as these features provide a broad range of high-level and low-level characteristics that are well-suited for content and style applications as presented in Chapter 4. Figure 6.2 exemplifies the encoding process of the MSFE block where superpixels are used to represent the target and reference images into smaller regions. Each of these smaller regions inside the mask M_s contains N_T and N_R superpixels, respectively, with P_i pixels each, where i is the superpixel index. We apply a channel-wise masked pooling operation on the chosen superpixels to perform the encoding. This results in super-features F with dimensions $C \times N$, where N is significantly smaller than $H \times W$. Our masked-super-attention block is inspired by the super-attention module presented in Chapter 5. Figure 6.3 shows the diagram of our masked super-attention block where f_T^ℓ, f_R^ℓ and φ_R^ℓ are feature maps from the encoder f and the encoder φ at level ℓ of T_L, R_L and R respectively. This module guides the colorization, considering the global context of a full reference image. However, our masked super-attention module focuses on specific object-to-object feature maps, helping the network guide the colorization to a particular

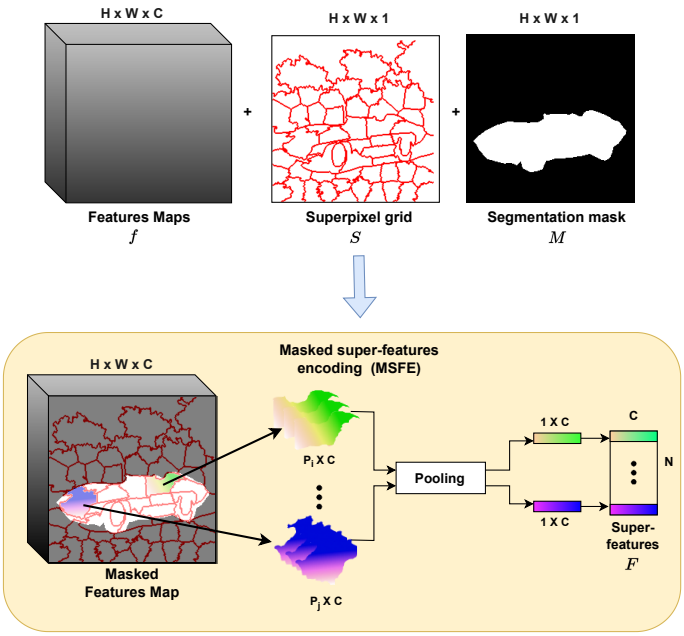


Figure 6.2: Diagram of our masked super-features encoding proposal (MSFE). This encoding block takes as input a feature map, a segmentation mask and a superpixel grid. In the feature maps each superpixel that belongs inside the mask M_s is extracted and encoded in vectors of size $C \times P_i$ pixels. Afterward, the vectors are pooled channel-wise and, finally, stacked in the super-features matrix F with size $C \times N$ number of superpixels.

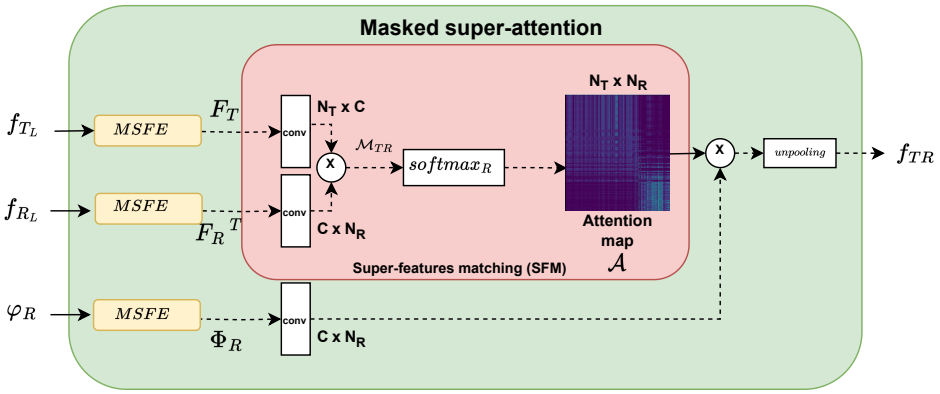


Figure 6.3: Overview of our masked super-attention layer. Given a reference luminance feature map, denoted as f_R, a reference color feature map represented as φ_R, and a target luminance feature map called f_T with their respective reference mask M_R and target mask M_T.

structure the user decides.

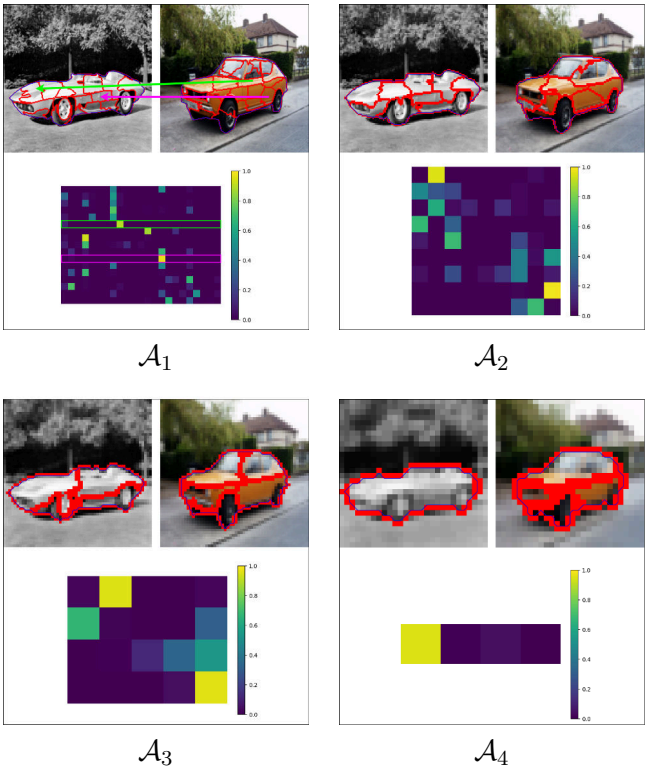


Figure 6.4: Example of masked super-attention mechanism guidance. Each image presents a target and a reference image with a superpose grid of superpixel using MaskSLIC (Irving, 2016) and its respective attention map $\mathcal{A}_\ell$. In each image $\mathcal{A}_\ell$ two arrows connect similar superpixels from the reference image to the target image. The similarity score between these superpixels is shown in the attention map, where each row represents a target superpixel and each column represents a reference superpixel.

Figure 6.4 illustrates attention maps $\mathcal{A}_\ell$ at all four levels ℓ of the architecture encoder. These attention maps depict the similarity between specific characteristics of an object in the target image and another in the reference image. This shows that the learned masked attention map can find relevant superpixels in the reference feature maps with similar characteristics to the target superpixel.

Masked super-attention vs. original super-attention. Masked super-attention can be seen as a generalization of the super-attention from (Chapter 5). To retrieve this global attention, we can simply apply the block without an object mask.

6.4 Training framework

6.4.1 Dataset and reference selection

Our framework was trained on COCO dataset (Lin et al., 2014). This dataset exhibits images with complex scene structures and diverse object classes. Additionally, it provides object segmentation information, which we later use in our strategy of pairing object-specific target and reference images. For our training, we use two different splits of the dataset. First, a full image-level split, which consists of 100k images for training and 5k images for validation. Second, an object-level split consists of 25k object images and their segmentation mask for training and 1k images for testing. We resized the images to a standardized size of 224×224 pixels during the training process.

Another essential aspect of the training strategy of exemplar-based methods is the identification of an appropriate semantic reference for the target image. For searching pairs between target and reference images in the full image-level split, we use the super-attention approach (Chapter 5) to match several reference images with each target one. In this approach, five reference images are ranked regarding semantic similarity using a pre-trained VGG-19 and a L2 distance. However, we found that after top-3, images do not convey significant semantic relevance with respect to the target image. We therefore keep only top-3 target images and complete this set with two additional pseudo-synthetic reference images. These two images are obtained by appearance and spatial transformation on the current target image using the Thin Plate Spline (TPS) (Bookstein, 1989; Lee et al., 2020a), a non-linear spatial transformation operator.

For the second split, we search pairs at the object level between target and reference images. First, we do a local search in each class to find meaningful objects whose size is larger than a percentage of the actual image. This is because image features are downsampled at each of the four levels of the architecture, and then small objects will not introduce meaningful characteristics to the attention calculation. For this, we set the percentage empirically to 30% of the image size. This is because doing superpixels on a smaller threshold (smaller objects) would not represent the actual object well in the architecture lower levels. Knowing the object class, we randomly sample three reference images from this class, and additionally, we apply TPS transformation on the target object to finish a top-5 reference object images.

Finally, during training and for both splits, target-reference pairs of images were sampled using a uniform distribution with a weight of 0.25. This was accomplished by randomly selecting either the three semantically closest reference images or the two synthetic references.

6.4.2 Implementation details

In this paper, we implement an U-net like generator architecture for our main colorization network Φ where, for each of the levels in the encoder, we introduce our masked super-attention block. Both the main model Φ and color encoder φ are jointly trained. We employed the Adam optimizer to optimize both networks with a learning rate of 10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. The training was conducted in two phases. First was a full image-level phase where we used the original super-attention without providing any segmentation

mask for 40 epochs. The second phase uses our masked super-attention (with the same weights as the first phase) to introduce object-specific characteristics to the network for 10 epochs. Throughout both phases, a batch size of 8 was used. In order to balance the losses, we set the coefficients for each loss function as follows: $\lambda_1 = 2$ and $\lambda_2 = 0.15$. The training process was carried out on a single NVIDIA RTX 2080 Ti using PyTorch 1.30.0 as the programming framework.

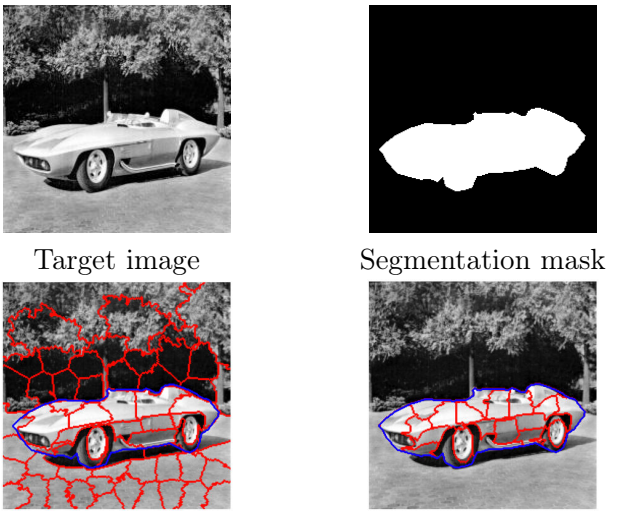


Figure 6.5: Example of superpixel algorithm on a grayscale image. The superpixel grids are generated using the SLIC and the MaskSLIC algorithm, which is a region-based image segmentation algorithm. The SLIC algorithm divides the full image into a set of small, non-overlapping regions. On the other hand, the MaskSLIC divides only the regions inside a segmentation mask.

We took inspiration from the original super-attention module (Chapter 5) when developing our masked super-attention approach. The classic super-attention module employs the SLIC algorithm (Achanta et al., 2012) to calculate superpixel segmentation on the full image level. However, our masked super-attention leverages an object-specific segmentation mask for the calculation. We then employ MaskSLIC (Irving, 2016) to compute both the target and reference masked superpixel grid. Figure 6.5 shows an example of the difference between SLIC and MaskSLIC. Note that any superpixel segmentation method could be used.

6.5 Evaluation

This section discusses the evaluation metrics and the qualitative and quantitative analysis between different variants of our proposed method and several state-of-the-art approaches. To begin, Subsection 6.5.1 introduces the evaluation metrics used for the quantitative analyses presented in the next subsections. Following that, Subsection 6.5.2 analyzes and compares the impact of incorporating the super-attention blocks into the skip-connections (Chapter 5) versus integrating them into the encoder of the architecture (Ours

w/o seg.). Next, Subsection 6.5.3 examines the advantages of applying either the first training phase (full image training) or both training phases (full image training + object image fine-tuning). For this comparison, we use two variants of our model: the first trained only using the first phase and subsequently performing inference using the mask super-attention (without prior object training), and the second variant, which represents our full proposal, involves fine-tuning the weights obtained from the first variant using the second training phase. In the last subsection 6.5.4, we will compare our full proposal with respect to four state-of-the-art exemplar-based colorization methods.

In detail, for the evaluation metrics, we group metrics into two groups; the first group uses metrics that compare the predicted results with the ground-truth image ($\hat{T}$ - GT target). and the second group compares the predicted image with the reference image ($\hat{T}$ - Reference). In addition, to comprehensively analyze our proposal, we perform quantitative analysis on two different test splits: the *full image split* and the *object image split*. The full image split evaluates the different metrics on the entire predicted image (classic approach), while the object image split evaluates the metrics directly on a specific object within the full image. For the latter, we cropped the object from where the color transfer was desired and measured each metric on this object image. By employing these two test splits, we gain a better understanding of how global and specific color characteristics are been transferred to the target image.

6.5.1 Metrics details

To quantitatively evaluate the results, we used six metrics. Four of the metrics compare the results with the ground-truth color image (SSIM, LPIPS, FID, and FID_∞), while the two other metrics compare the prediction of colors with respect to the reference color image ($LPIPS_R$ and ΔHIS). Following is a small recapitulation of each of of the metrics:

Structural similarity (SSIM) (Wang et al., 2004). This metric analyzes the ability of the model to reconstruct the original image color and texture.

Learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018b). The goal is to measure the perceptual similarity between the predicted colorized image and ground-truth image.

Learned perceptual image patch similarity w.r.t reference (LPIPS_R) (Mechrez et al., 2018). This metric measures the perceptual similarity between non-aligned images, in this case, the predicted colorized image and the reference image.

Fréchet Inception Distance (FID) (Heusel et al., 2017). This metric measures the similarity between the distribution of features extracted from a set of predicted images and the distribution of features extracted from a set of ground-truth images.

Fréchet Inception Distance infinity (FID_∞) (Chong et Forsyth, 2020). This metric is particularly useful for comparing our test set at the object level, especially when our current split comprises only $N = 1k$ images and is susceptible to this bias issue. In detail, we let $k = 15$ as the default value in their metric. In addition, we choose to calculate FID_{300} and FID_{600} as they are sufficient to know the true tendency of the metric. Finally, to ensure robustness and reliability in our results, since the metric relies on randomly sampled intervals from the test set, we evaluate FID_{300} and FID_{600} ten times. The final results are obtained by calculating the average and standard deviation across these ten evaluations.

Histogram intersection similarity (HIS) (Isola et al., 2017). This metric evaluates the similarity of the global color distributions of the two images. For this evaluation, we choose to calculate the difference in HIS score (Δ HIS) between the ground-truth target images and the reference images and the predicted colorized target image and the reference.

The final results for these six metrics are the averages calculated using either the full-image evaluation set from the COCO validation set or the subset of object-level within the same dataset. These six metrics provide a comprehensive measurement of the quality of the output colorization at the full-level image and the object-level image. For more details regarding the previous evaluation metrics, see Section 2.7.

6.5.2 Analysis on super-attention in the encoder

We conduct qualitative and quantitative evaluations to inspect the super-attention effectiveness in the encoder rather than in the skip-connections, as presented in Chapter 5. For this analysis, we have two variants of the same architecture with super-attention blocks: in the first variant, the super-attention mechanism is applied only within the encoder part of the architecture (ours without segmentation). And for the second variant, it incorporates super-attention into the skip-connections (Chapter 5).

In terms of quantitative results, Table 6.1 shows that our proposal without segmentation, retrieves a better FID score than using the original super-attention in the skip-connections. and achieve comparable results in the other two metrics that compare the target with respect to the ground-truth (LPIPS and SSIM). For the metrics comparing the results with respect to the reference image, our proposal without segmentation achieves better results in LPIPS_R and ΔHIS, which means that results using super-attention in the encoder enforce the transfer of more semantics characteristics from the reference image, as well as better similarity on the global tone from the reference image than the super-attention approach.

Table 6.1: Quantitative analysis between super-attention in the skip-connections (Chapter 5) and super-attention in the encoder at full image level.

	Full image split				
	$\hat{T}$ - GT target			$\hat{T}$ - Reference	
Method	**SSIM↑**	**LPIPS↓**	**FID↓**	**LPIPS_R↓**	**Δ HIS↓**
Super-attention (Chapter 5)	**0.92**	**0.14**	11.24	2.14	0.23
Ours w/o seg.	0.91	**0.14**	**9.20**	**2.01**	**0.18**

Figure 6.6 compares these two versions qualitatively. In the case of the super-attention in the skip-connections (Chapter 5), it produces quite opaque colors, which means that colors expected from the reference images are not been completely transferred but are average to the target image. On the other hand, placing the super-attention in the encoder forces a more vivid transfer of color characteristics between reference and target images.

Figure 6.6: Results of using the original super-attention (Chapter 5) on skip-connections and our implementation with super-attention module in the encoder. Our proposal is more effective at transferring color characteristics between the reference and the target, particularly for specific objects.

6.5.3 Fine-tuning at object-level

This subsection evaluates the benefits of training our proposal with object-level images. We conducted experiments on two versions of our model using the same architecture and pre-trained at the full-image level (first training phase). The first version uses the masked super-attention module without further fine-tuning or training on object-specific images (just using the full image split). The second version is our full proposal, using the weights from the first version, we fine-tune using the object level split (second training phase). We compared the two models quantitatively and qualitatively.

Table 6.2 demonstrates that even without fine-tuning, our method achieves promising results, particularly in terms of LPIPS_R where it surpasses our full proposal, indicating that the masked super-attention effectively transfers meaningful semantic features from the reference object to the target object. However, fine-tuning our proposal (ours) improves its performance across the rest of the metrics. Our full proposal achieved better LPIPS and FID results, meaning that stronger perceptual similarities are retained between the colorized results and the target ground-truth. Finally, our full method achieves smaller ΔHIS, meaning that the global tone of the reference object is also well transferred. In addition to the quantitative evaluation, Figure 6.7 provides a qualitative comparison of the two approaches. In the first row, the goal is to colorize the woman's sweater in the target image with the color characteristics of the pink pyjamas in the reference image while ensuring that the rest of the image is properly colorized. The result achieved by our proposal without leaning at the object level shows a correct transfer of color within the sweater; however, the face shows a not visually pleasant grayish color with slight color bleeding on the wall. Our full proposal overcomes previous aspects; however, we got a yellowish tonality in the back wall. For the last image, the goal is to transfer the bird's blue in the reference images to the bird in the image in the grayscale. The result of our

proposal, without fine-tuning at the object level, presents a fairly good transfer of colors from the global tones within the object, as we can see that a mix of dark blue is transferred to the target image. However, our full proposal shows a more colorful colorization with a brighter blue, nearly as the one in the reference mask image.

Table 6.2: Quantitative results on fine-tuning at object-level. Ours w/o fine-tune. corresponds to our proposal without fine-tuning on object images but using masked super-attention. Ours corresponds to our full model with masked super-attention module and after fine-tuning with object-related images.

	Object level split					
	$\hat{T}$ - GT target				$\hat{T}$ - Reference	
Method	**LPIPS↓**	**FID↓**	**FID$_{\infty_300}$↓**	**FID$_{\infty_600}$↓**	**LPIPS$_R$↓**	**Δ HIS↓**
Ours w/o fine-tune.	0.17	32.80	10.87 ± 0.29	4.02 ± 0.33	**1.92**	0.18
Ours	**0.15**	**30.45**	**6.80 ± 0.28**	**2.67 ± 0.32**	2.04	**0.17**

The results showed that our full proposal achieved the best performance in terms of both quantitative and qualitative metrics with respect to our proposal without fine-tuning object-specific images. This suggests that by fine-tuning the masked super-attention results gains more spatial consistency in colors between the object and the image background, resulting in more naturalness in the image.

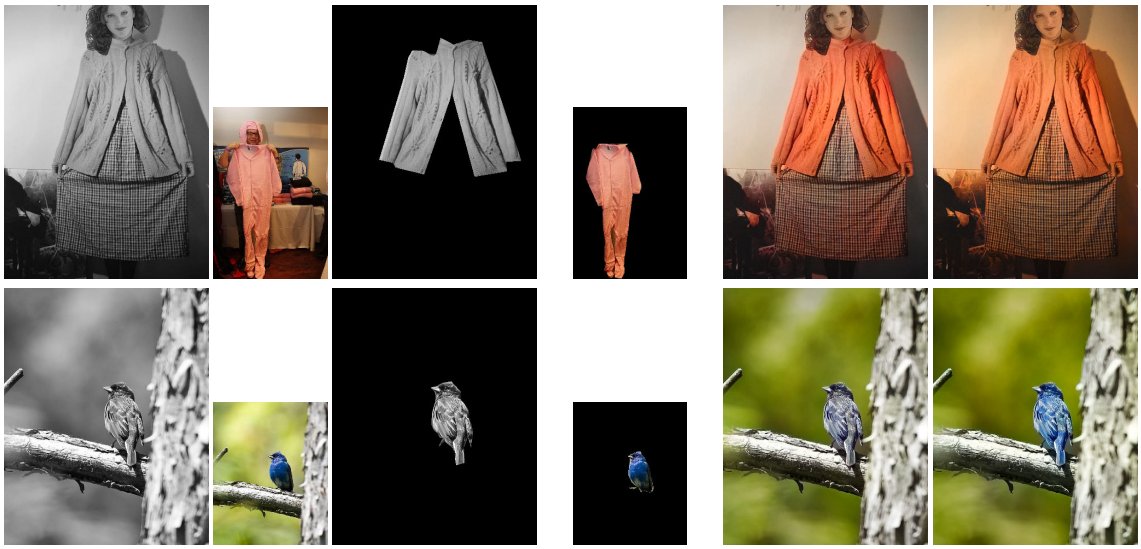


Figure 6.7: Comparison of the proposal with and without fine-tuning at object-level. The first four columns show the inputs to our framework, which the user provides. The fifth column shows the results of our framework without object-level learning. This means we only use our masked super-attention module without additional training on object-specific images. The last column shows the results of our full method, which includes both the masked super-attention module and fine-tuning on object-level images.

Table 6.3: Comparative evaluation at full image level.

	Full image split				
	$\hat{T}$ - GT target			$\hat{T}$ - Reference	
Method	**SSIM↑**	**LPIPS↓**	**FID↓**	**LPIPS$_R$↓**	**Δ HIS↓**
Gray2ColorNet (Lu et al., 2020)	0.89	0.24	22.04	2.17	0.34
Just attention (Yin et al., 2021)	0.90	0.23	16.80	2.23	0.26
Super-attention (Chapter 5)	**0.92**	**0.14**	11.24	2.14	0.23
Unicolor (Huang et al., 2022)	0.88	0.22	9.70	2.11	0.21
Ours w/o seg.	0.91	**0.14**	**9.20**	**2.01**	**0.18**

Table 6.4: Comparative evaluation at object-level.

	Object level split					
	$\hat{T}$ - GT target				$\hat{T}$ - Reference	
Method ↑	**LPIPS↓**	**FID↓**	**FID$_{\infty_300}$ ↓**	**FID$_{\infty_600}$ ↓**	**LPIPS$_R$↓**	**Δ HIS↓**
Gray2ColorNet (Lu et al., 2020)	0.18	35.85	19.04 ± 0.31	18.40 ± 0.26	2.88	0.21
Just attention (Yin et al., 2021)	0.19	39.65	15.34 ± 0.23	14.76 ± 0.45	2.58	0.18
Super-attention (Chapter 5)	0.17	32.61	8.22 ± 0.28	4.28 ± 0.38	2.14	0.20
Unicolor (Huang et al., 2022)	0.23	32.40	7.64 ± 0.15	3.57 ± 0.32	**1.87**	**0.16**
Ours	**0.15**	**30.45**	**6.80 ± 0.28**	**2.67 ± 0.32**	2.04	**0.16**

6.5.4 Comparison with state-of-the-art

To evaluate the performance of our framework, we compared quantitatively and qualitatively our results to four other state-of-the-art exemplar-based image colorization approaches: Gray2colorNet (Lu et al., 2020), Just attention (Yin et al., 2021), Super-attention (Chapter 5) and Unicolor (Huang et al., 2022). To ensure a fair comparison, we ran the available codes for the four approaches using the same experimental protocol and the same evaluation set for all the methods.

Color transfer at full image level. As shown in Table 6.3 our proposal without object-specific segmentation obtains the best LPIPS, FID, and LPIPS$_R$. The latter means that our framework retains strong perceptual information not only from the original target image but as well as from the reference color image. For the SSIM metric, ours achieve competitive results with respect to super-attention (Chapter 5) and surpasses all four other methods. Finally, our method achieves a smaller ΔHIS with respect to all compared state-of-the-art methods. This indicates that rather than forcing to transfer all colors from the reference images, our model has the ability to selectively choose specific colors from the references. As a result, it can generate natural colorization results.

Color transfer over object. Table 6.4 shows the comparison of the four evaluation metrics for each of the different methods. It is important to note that these metrics were calculated on object-specific images. Instead of doing calculations on the entire predicted image, we cropped the specific object for which color transfer was desired and measured each of the metrics on this object image. In terms of metrics, our full proposal retrieves more perceptually semantic characteristics at the object level than the other four methods. From all the variants of FID calculations, our method manages to well retain similar characteristics distribution from the ground-truth images. LPIPS$_R$ metric measures how well the model transfers perceptual characteristics from the reference image to the target one. In this case, (Huang et al., 2022) achieves better results. Finally, in terms of ΔHIS, our full method achieves comparable results to (Huang et al., 2022). This demonstrates

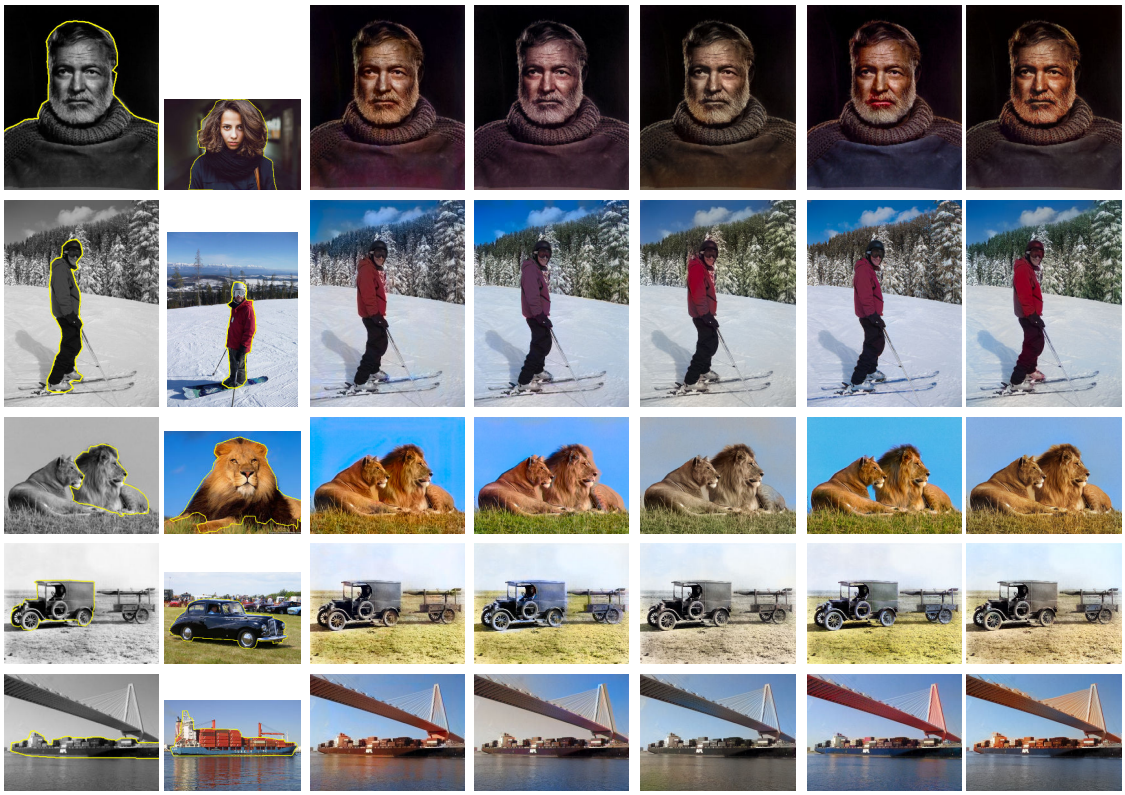


Figure 6.8: Comparison of our proposed method with different reference-based colorization methods. Target and reference images are the only input to the SOTA methods: Gray2colorNet (Lu et al., 2020), Just Attention (Yin et al., 2021), Super-attention (Chapter 5), Unicolor (Huang et al., 2022). For our full method, in addition to the target and reference images, users can also provide an object segmentation mask. The yellow contours in both columns of images indicate the object segmentation mask.

the capability to transfer color characteristics from an object reference to specific regions on the target object.

Figure 6.8 shows results of image colorization from state-of-the-art methods and ours: (Lu et al., 2020), (Yin et al., 2021), (Chapter 5), (Huang et al., 2022), and our full method. For the first two images, our proposal produces more visually pleasant and natural colorization than the other four methods. The results from (Lu et al., 2020), (Yin et al., 2021), and (Chapter 5) fail to transfer the blue from the woman's sweater. Additionally, (Yin et al., 2021) and (Chapter 5) produce washed-out colors, making the head and clothes having the same color in the first image. For the fourth image, (Lu et al., 2020) and (Huang et al., 2022) show a high amount of color bleeding, mainly on the car. This color bleeding also appears on the small trailer and in its background. In contrast, our proposal and (Lu et al., 2020) shows the right balance between transferring colors between objects and colorizing the background. Finally, for the last image, (Yin et al., 2021), (Chapter 5), and (Huang et al., 2022) struggle to find the correct colors to transfer. The first method transfers a red

color, the second method transfers a blue color, and the third method transfers an average of colors. Our proposal and (Huang et al., 2022) correctly transfer vivid colors from the reference image, especially from the ship in the reference image. Our proposal and (Huang et al., 2022) shows the right balance between a colorful colorization and the naturalness from the learned colorization model.

6.6 Conclusion and future works

In this chapter, we have introduced a novel end-to-end deep learning framework for exemplar-based colorization, which stands out for its ability to incorporate user-provided object masks. Our proposed masked super-attention is an extension of the previous super-attention presented in Chapter 5, it successfully guides the colorization based on specific objects within the image. The addition of this new block results in visually pleasant and spatially consistent colorized images with vivid colors. We performed a comprehensive evaluation, which includes both full-image and object-level metrics, outperforming quantitatively state-of-the-art methods in three of the metrics. However, we believe there is room for improvement, particularly regarding the low-resolution layers where the masked super-attention module is applied. This is because features, specially from small objects in these layers, often lack from sufficient details and as a result it can encountered limited amount of superpixels to provide a useful matching. A possible solution involves calculating the attention maps on upsampled low-level features and re-weighing them with all attention maps.

Another future line of work is to study the clipping problem arising from passing from Lab to RGB spaces when the combination of predicted Lab values falls outside the conversion range. One solution could be using an oblique projection (Pierre et al., 2015b) in the final part of our model. Finally, another perspective of research could be to study combining other user interactions, such as color scribbles (Heu et al., 2009; Zhang et al., 2021) with our reference-based colorization framework. For the next Chapter 7, we will study the interaction with color scribbles on the application of line art colorization. We will propose a method to introduce these user color hints into a new family of architectures called probabilistic diffusion models.

Chapter 7

Conditioning diffusion models with user colors for line art colorization

Table of contents

Summary

Colorization of line art drawings is an important task in illustration and animation workflows. However, this highly laborious process is mainly done manually, limiting the creative productivity. In this chapter, we present a novel interactive approach for line art colorization using conditional Diffusion Probabilistic Models (DPMs). In our proposed approach, the user provides initial color strokes for colorizing the line art. The strokes are then integrated into the conditional DPM-based colorization process by means of a coupled implicit and explicit conditioning strategy to generate diverse and high-quality colorized images. We evaluate our proposal and show it outperforms existing state-of-the-art approaches using the FID, LPIPS, and SSIM metrics. This work was achieved during a 2-month research stay at Waseda university in Tokyo, japan.

Related publications

[Carrillo *et al.* 2023] **H. Carrillo,** E. Simo-Serra, M. Clément, A. Bugeau,. "Diffusart: Enhancing Line Art Colorization with Conditional Diffusion Models." *Conference on Computer Vision and Pattern Recognition Workshops (CPRW)*, 2023.

Contributions

In this chapter, the main contributions are:

- We introduce a novel approach for user-guided line art colorization using conditional Diffusion Models.
- We exploit coupled implicit and explicit conditioning strategy for line-art colorization task.

7.1 Introduction

As mentioned in Chapter 1, automatic colorization has been used in a wide range of sectors, such as cultural heritage, broadcasting, post-production, digital art, and animation. In previous Chapters 3, 5 and 6, we analyzed and developed colorization tools using natural images covering the cultural heritage, broadcasting, and part of the post-production areas. However, colorization also extends to the digital art, and animation areas where other types of images are used, such as line art or sketches. Line-art is the most basic form of drawing; it mainly uses straight or curved lines to create an image. This type of drawing does not use color or shading to represent the object within the image. Line art colorization then, involves adding color to black-and-white sketches to make them visually appealing and expressive. In this task, the artist usually follows some steps to colorize an image. Figure 7.1 shows the results of several steps in the line-art colorization process. In brief, the artists first apply flat coloring, by adding the base colors (without shading or highlighting) into the corresponding areas or objects. Second, the artist adds the sensation of depth and dimension by incorporating highlighting and shading details and, finally last step is, detailing and texturing; this addition can include more complex textural details such as gradients of colors, patterns, etc. However, this process is laborious as it is typically carried out manually for traditional animation, mainly using software illustration tools such as Photoshop, ClipStudio, and Krita (see Figure 7.2). Automated colorization has the potential to significantly enhance an artist's workflow.

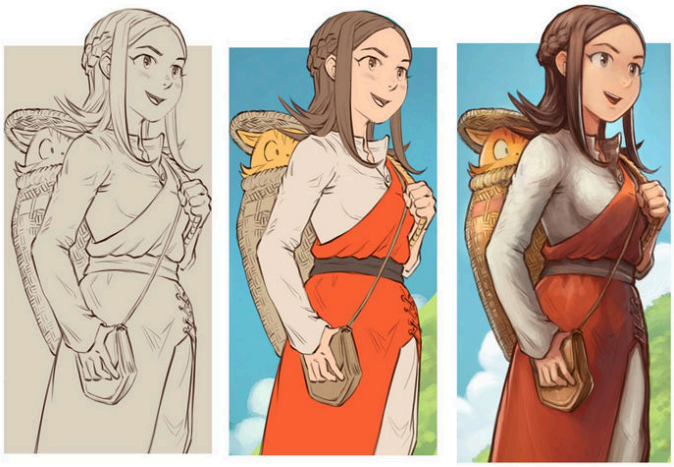


Figure 7.1: Example of colonization steps in line art from (Revoy, 2022). The process begins with the first image, which shows a digital line-art with some smooth traces to guide the user's next steps. The second image presents the same line art, now filled with basic, flat colors. Finally, the third image shows the final result of the line-art colorization with color, shadow, and highlight details.

In recent years, various methods have explored user-guided automatic line art colorization using deep learning. In particular, Generative Adversarial Network (GANs) architectures have been proposed (Ci et al., 2018; Sangkloy et al., 2017; Yliess et al., 2019; Yuan et Simo-Serra, 2021; Zhang et al., 2021). These methods couple color hints as input with learned color priors from large-scale datasets to colorize the line art images. GAN archi-

Figure 7.2: Example of flat coloring in line art colorization (Revoy, 2022). The first image shows the user's color scribbles on top of the line-art. The second image shows the result of the flat coloring.

tectures can achieve impressive and high-quality outputs. However, certain issues remain problematic, for example, as shows in Figure 7.3 ensuring color consistency with user color inputs and reaching color harmony between small image regions. In addition, GAN architectures can be challenging to train due to instabilities (Arjovsky et al., 2017; Gulrajani et al., 2017). To overcome these issues, Diffusion Probabilistic Models (DPMs) (Ho et al., 2020; Song et Ermon, 2019) propose a framework capable of generating high-fidelity images by training a U-Net (Ronneberger et al., 2015) like generator architecture and sampling from a Markov chain. These methods have been applied to various computer vision problems such as image synthesis (Dhariwal et Nichol, 2021), super-resolution (Saharia et al., 2022b; Rombach et al., 2022), and automatic image colorization (Saharia et al., 2022a) achieving better qualitative and quantitative results than previous state-of-the-art GANs architectures. In this chapter, we propose a method for line-art colorization using the user's color scribbles. This novel colorization framework is based on a conditional diffusion model that achieves better results than state-of-the-art methods. Figure 7.4 shows an example of the inputs and the colorized result of our proposal.

The rest of the chapter is organized as follows: in Section 7.2, we review the existing literature on line-art colorization. Then, in Section 7.3 we explain the bases of diffusion models in an unconditional manner for image generation. Furthermore in Section 7.5 we present the process of creation of our synthetic dataset and the simulation of scribbles. Next, in Section 7.4, we propose a method for conditioning a diffusion model to colorize line art. Finally, in Section 7.6, we evaluate the proposed method against state-of-the-art approaches using both quantitative and qualitative metrics.

Context of this chapter. The work in this chapter arose during my stay at the Simo-Serra laboratory at Waseda university in Tokyo, Japan where I stayed during two months and a half (from October to December 2022). In the Simo-Serra Laboratory, they specialize in the application of machine learning, computer graphics, and computer vision techniques to solve task such as: augmenting creative processes with applications to illustration, design, and fabrication. The experience I gained there was incredibly enriching, both professionally and personally. I had the opportunity to work in a relatively different research subject, and work environment with different colleagues that allowed me to expand my skills and knowledge as a researcher. I worked directly with Dr. Simo-Serra on the line-art colorization task and on the new Diffusion Probabilistic Models (DPMs), where, thanks to his guidance, I substantially learned about these subjects. One of the

Figure 7.3: Result of line-art colorization using (Yliess et al., 2019). The first column is the line-art image to colorize. The second is the mask of color scribbles provided by the user. And finally, is the resulting colorized line-art. In the last image, we can observe color bleeding on the right hand and in the hair.

highlights of this mobility was our proposal of a line-art colorization framework called Diffussart (See Figure 7.4), which was published in the workshop on Fashion, Art, and Design at the CVPR conference.

Figure 7.4: Diffusart enables the colorization of line arts l created by an artist (left) using color scribbles s (center). On the right is the result of our proposal $\hat{x}_0$.

7.2 Related work

Colorizing line art is a different challenge compared to coloring grayscale natural images. In line art, the drawn images are usually in black and white. This changes from classic image colorization, where it has a valuable and wide range of gray tones information. Therefore, when coloring line art images, the task is to creatively fill in the white spaces with an unlimited variety of colors and shades. In this process, the most useful pieces of information are the edges, the size of the areas, and the specific position of each pixel. This makes line art colorization a unique and interesting problem, where the objective is to make the most out of unlimited color information and user input to bring the artwork to life. Many works on automatic methods have been proposed over the years. The first

automatic methods for line art colorization have been primarily classical image processing optimization-based, as described in (Sýkora et al., 2009; Qu et al., 2006). These approaches typically involve using image features such as pattern and intensity continuity to propagate color hints over regions. However, their effectiveness is limited when applied to complex line drawings where a high amount of user color hints are needed.

Deep learning methods, and in particular GAN architectures (Zhang et al., 2018a; Liu et al., 2018; Yuan et Simo-Serra, 2021), have been used for user-guided line art colorization where color scribbles are propagated based on neural networks trained with large amounts of data. In (Ci et al., 2018), the authors propose a method to enhance the generalization capability of the neural network by introducing a local features network independent of synthetic data. The method trains a conditional GAN on a join loss using Wasserstein distance loss and contextual loss in order to close the gap between the input pairs, line art, and randomly sampled color points to the ground-truth. However, this method uses a small dataset of 23k images for training, which, in the end, impacts the generalization to other types of line-art images. In (Yliess et al., 2019) they improve the training process by using stroke simulation as a substitute for random pixel sampling for the synthetic color hints generation. They also ameliorate the visual fidelity of results by using a cross-domain double-generator approach. The first generator is capable of generating a fake colored illustration from the given line art and color hints (result), on the other hand, the second generator is responsible for generating a synthetic line art out of the fake illustration inferred by the first generator. Other methods explore the use of reference color images to transfer a particular artist style (Furusawa et al., 2017; Lee et al., 2020a; Li et al., 2022b). Although GANs can produce high-fidelity images, these architectures can be unstable to train which could produce perceptually unsatisfying results.

Diffusion Probabilistic Models (DPMs) (Ho et al., 2020; Song et Ermon, 2019) seem to have the potential to overcome GANs issues in many applications. These methods have recently emerged as a class of generative models for high-dimensional data such as images and audio. DPMs transform the input data through a series of controlled noising/denoising steps to predict new data distributions. These methods have achieved state-of-the-art results in various image-to-image generation tasks, including image synthesis (Dhariwal et Nichol, 2021), superresolution (Li et al., 2022a; Saharia et al., 2022b), and colorization(Saharia et al., 2022a). Inspired by these new methods, we propose a novel conditional diffusion model for line art colorization that can be guided by user color scribbles.

7.3 Unconditional diffusion models

Diffusion models, which have been used in various image generation applications, convert standard Gaussian distribution samples to empirical data distribution samples by employing a Markov chain denoising process of T steps. Given an initial image x_0 from the real distribution, the diffusion process successively adds Gaussian noise with variance β_t and mean $\mu_t = \sqrt{1-\beta_t}x_{t-1}$ to obtain intermediate noisy image x_t, this is called forward process. That is:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon. \tag{7.1}$$

By replacing $\alpha_t = \prod_{i=1}^{T}(1-\beta_i)$ in Equation (7.1), we can then apply a parametrization

trick, where in DPMs we can express the input image x_t in relation to the initial image x_0:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, \tag{7.2}$$

where t goes from 0 to T and determines the current step of the noise, which means $\alpha_0 = 1$ clear image and $\alpha_T = 0$ pure noise.

In the reverse process, the idea is to learn the reverse chain. For that, we use a neural denoising model ϵ_θ, which approximates the noise added at an exact given timestep t. Those unconditional diffusion models are trained using the following loss function:

$$L_p = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),t}|\epsilon - \epsilon_\theta(x_t, t)|_p, \tag{7.3}$$

where p could be 1 or 2 making reference to the L_1 or L_2 losses.

Figure 7.5 shows the noise and denoised steps. For the inference step, the idea is to approximate the unknown conditional distribution $p_\theta(x_{t-1} \mid x_t)$ based on the learned parametric approximation. This is done by a stochastic iterative refinement process and calculated using:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_{t-1} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_{t-1}, t-1\right)\right) + \sigma_t\mathbf{z} \tag{7.4}$$

During inference, the aim is to reverse the Gaussian diffusion process through a reverse Markov chain according to a learned transition distribution p_θ:

$$p_\theta\left(x_{0:T}\right) = p_\theta\left(x_T\right)\prod_{t=1}^{T} p_\theta\left(x_{t-1} \mid x_t\right). \tag{7.5}$$

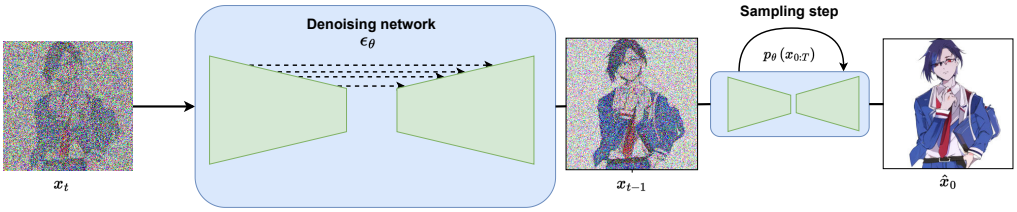


Figure 7.5: Diagram of unconditional diffusion model for line art generation.

7.4 Conditional diffusion models for line art colorization

Diffusion models such as those proposed by (Ho et al., 2020; Song et Ermon, 2019) operate in a non-conditional setting as in Equation (7.5). Instead, we jointly use two approaches to condition our diffusion model. First, as in (Rombach et al., 2022), we jointly train an application-specific encoder g_θ, which extracts semantic features from color scribbles and line art images. These features are then introduced to the denoising model ϵ_θ by means of cross-attention mechanism (Vaswani et al., 2017). For the second approach, and inspired by (Chen et al., 2021; Saharia et al., 2022a), we explicitly condition the predicted

distribution on the denoising neural network ϵ_θ by directly concatenating line art image s to the noisy input x_t. Finally, by joining both conditioning in the inference process (7.5) changes to

$$p_\theta\left(x_{0:T} \mid (l,s)\right) = p_\theta\left(x_T\right) \prod_{t=1}^{T} p_\theta\left(x_{t-1} \mid x_t, (l,s)\right). \tag{7.6}$$

Given (7.6), the training process of our proposal on conditioned image pairs is trained using the L_1 loss as

$$L_1 = \mathbb{E}_{l,\epsilon\sim\mathcal{N}(0,1),t} |\epsilon - \epsilon_\theta\left(l, x_t, t, g_\theta(l,s)\right)|_1, \tag{7.7}$$

where both the denoising model ϵ_θ and the encoder g_θ are jointly trained.

The objective is to colorize a grayscale line art image from user color scribbles. Our proposal uses a diffusion model, which learns to generate a colorized line art image $\hat{x}_0$ given grayscale line art l and color scribbles s (see Figure 7.4).

Our framework is composed of two main parts: a denoising model ϵ_θ from the main denoising pipeline and an application-specific encoder g_θ for extracting color scribbles information (see Figure 7.6). The first learns to denoise noisy images from an unknown distribution conditioned to a line art image l. The second part, g_θ, extracts color features previously introduced by the user to guide the line art colorization. Finally, the predicted image $\hat{x}_0$ is retrieved using the DDPM sampling algorithm (Ho et al., 2020).

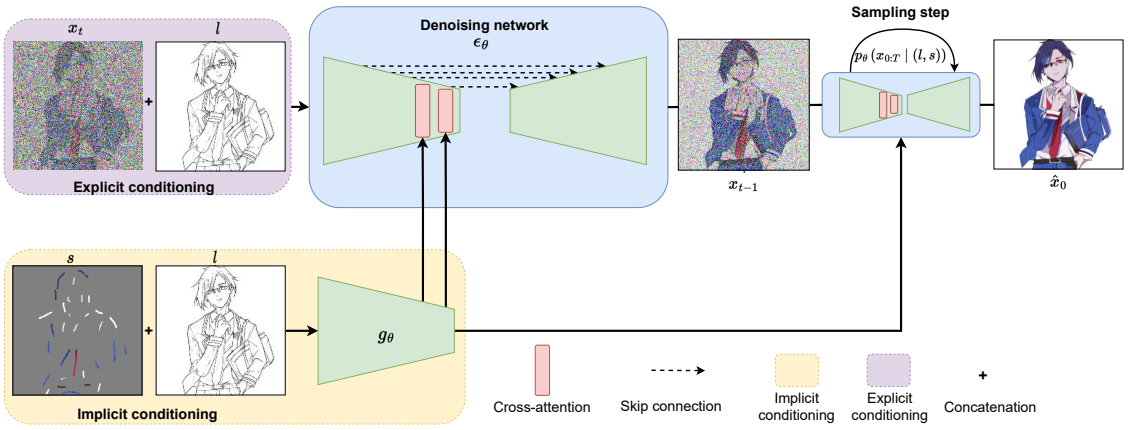


Figure 7.6: Overview of our proposed user-guided line art colorization. The framework is composed of two main components: a denoising model ϵ_θ, which learns to generate a denoised image, and an application-specific encoder g_θ for extracting user color scribbles information.

7.5 Dataset preparation

Synthetic Line Art. We conducted experiments using a subsample of color illustrations from the dataset safe Danbooru2021 (DanbooruCommunity, 2021). To filter out grayscale images, we utilized tags such as *grayscale* and *monochrome*. We use 200k training images

and 13k images for test. For creating synthetic line art, we rely on two extraction methods: SketchKeras (lllyasviel, 2017) and Sketch simplification (Simo-Serra et al., 2018). Figure 7.7 depicts an example of synthetic data generated by previously mentioned methods. Finally, the type of sketch at training time is randomly sample by a uniform distribution with a 50% probability of choosing SketchKeras or the Sketch simplification methods.

Color image SketchKeras Sketch Simplification

Figure 7.7: Example of synthetic line art of a color image generated using SketchKeras (lllyasviel, 2017), and Sketch Simplification (Simo-Serra et al., 2018).

Simulated Color Scribbles. To achieve a model that can handle user color inputs, we simulate human scribbles by randomly sampling vertical, horizontal, and diagonal lines. We use three parameters: the number of scribbles sampled from the uniform distribution $\mathcal{U}(4, 25)$, scribble thickness sampled from $\mathcal{U}(1, 4)$ pixels, and scribble length sampled from $\mathcal{U}(5, 30)$ pixels. Additionally, as a high amount of illustrations in the dataset present white backgrounds, there is a high probability that the synthetic scribbles would bias the model toward the color white. Therefore, we only use the sampled synthetic scribbles when they contain less than 60% white pixels.

7.6 Experimental validation

7.6.1 Implementation details

Our implementation was inspired by (Ho et al., 2020). To reduce computational cost, we only use self-attention and cross-attention mechanisms on the bottleneck of the denoising model ϵ_θ. We use the Adam optimizer with a learning rate of $2e^{-5}$, a cosine warm-up schedule for 5k training steps, and a batch size of 40. We introduce a color feature extraction encoder g_θ that uses the same encoder architecture as ϵ_θ with only one residual block per layer instead of two. Both the denoising model ϵ_θ and encoder g_θ are jointly trained from scratch. All line art and color scribble images are fixed to the resolution 256×256, and values are normalized to the range $[-1, 1]$. Our final method was trained for 80 epochs with 10 NVIDIA RTX 2080Ti GPUs.

To evaluate the effectiveness of our line art colorization framework, we compare our results quantitatively and qualitatively with two other state-of-the-art user-guided line art colorization approaches (Ci et al., 2018; Yliess et al., 2019). In order to do a fair comparison, we retrained all the methods with the same dataset and used the default parameters presented in their methods.

7.6.2 Quantitative Evaluation.

We use three metrics to compare different methods quantitatively. The first metric is the Structural Similarity (SSIM) (Wang et al., 2004), which examines the model's ability to reconstruct the content of the original image. The second metric is the Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018b), which measures perceptual similarities between two images and correlates strongly with human perception. The last metric, the Fréchet Inception Distance (FID) (Heusel et al., 2017), is used to measure a perceptual similarity between two sets of images. All three metrics are calculated on 13k test images, between the color illustration image as ground-truth and generated images with fixed color hints from the different methods.

Table 7.1: Quantitative comparison with state-of-the-art user-guided line art colorization methods.

Method	**SSIM↑**	**LPIPS ↓**	**FID↓**
AlacGAN (Ci et al., 2018)	0.66	0.26	13.27
PaintsTorch (Yliess et al., 2019)	0.79	**0.14**	8.79
Ours (w/o explicit cond.)	0.77	0.15	7.91
Ours (full)	**0.81**	**0.14**	**6.15**

As shown in Table 7.1 our results retain 15% and 2% more structural information than the other two state-of-the-art methods. For the LPIPS metric, our method surpasses (Ci et al., 2018) and achieves comparable results to (Yliess et al., 2019). Finally, on the FID metric, we outperform both methods. In addition, using only implicit conditioning reduces the performance compared to our full method.

Qualitative Evaluation. Figure 7.8 shows the results from the two state-of-the-art methods (Ci et al., 2018), (Yliess et al., 2019), and ours. We can see that our qualitative results are consistent with quantitative scores, showing more high-quality details, visually appealing colorization, and a more accurate representation of color shades compared to the other two methods.

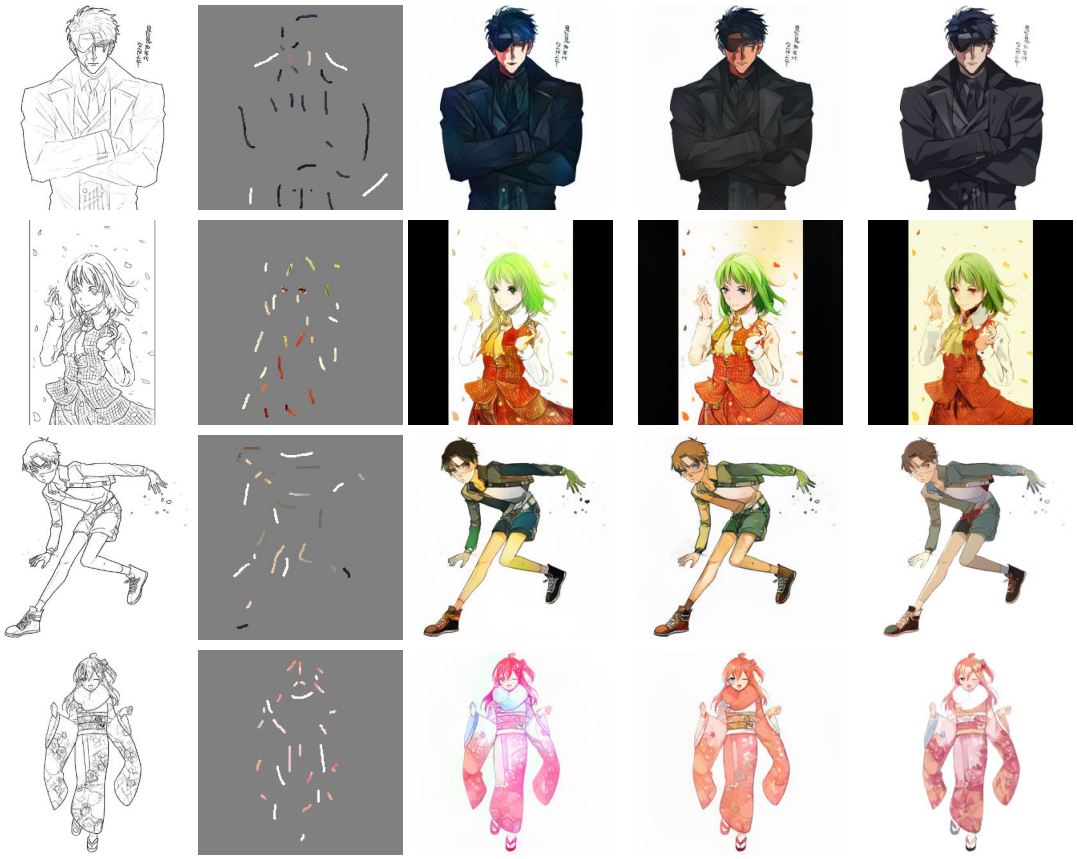


Figure 7.8: Comparison of our proposed method with different user-guided line art colorization methods: AlacGAN (Ci et al., 2018) and PaintsTorch (Yliess et al., 2019).

7.7 Conclusion and future works

In this chapter, we introduced a novel approach for user-guided line art colorization using conditional Diffusion Models. Our proposal exploits a coupled implicit and explicit conditioning strategy that ensures a robust structural generation of details and an accurate representation of user colors. Experimental evaluation on a large-scale dataset shows that our method outperforms existing techniques both quantitatively and qualitatively.

In terms of future work, one particularly interesting area is enhancing the inference speed of image generation in diffusion models. In this context, the use of latent embedding presents an interesting line of work. Specifically, in (Rombach et al., 2022), they propose a method where instead of doing the noise and denoising steps in the image space, they train application specifics encoder-decoder to represent the image into the latent space, which could significantly accelerate the inference time. Another line of work could be exploring recent approaches for creating synthetic color scribbles. Current techniques typically rely on sequential algorithms, which can be highly time-intensive. Exploring and developing

new approaches in this area could lead to more efficient processes, potentially creating a formal way that color scribbles could be generated.

Chapter 8

General conclusions and future works

8.1 General conclusions

In this thesis, we proposed several contributions to the field of automatic image colorization, which is an area of growing importance in image editing, post-production, digital art, and animation. In Chapter 2, we established a comprehensive understanding of the current landscape, reviewing state-of-the-art colorization methods, color spaces, and evaluation techniques. This in-depth exploration revealed that while deep learning approaches are the current state-of-the-art methods in this task, their complex architectures and reliance on multiple joint losses make it difficult to comprehend each novelty of each method.

To address these challenges, in Chapter 3, we leveraged a simple encoder-decoder deep learning architecture for automatic image colorization. Here, we demonstrated that selecting the appropriate color space and loss functions is not a straightforward process and may be influenced by various factors, such as the network architecture or the type of images. However, for our architecture, the models that include the VGG-based LPIPS loss function generally produces slightly better results, especially for the perceptual metrics LPIPS and FID.

Our second major contribution was presented in Chapter 4, which was a novel attention layer that we called *Super-attention*. This proposal relies on superpixels; this layer addresses the quadratic complexity issues inherent to traditional attention mechanisms and effectively mitigates common colorization artifacts such as color bleeding. We then used it in the color transfer application, achieving impressive and more natural qualitative results in comparison with state-of-the-art methods.

In Chapter 5 and Chapter 6 respectively, we proposed two semi-automatic exemplar-based colorization frameworks. The first relied on the super-attention block as a color prior to capturing multi-level correspondences between high-resolution deep features from pairs of images. Our second framework presents the *Masked super-attention*, which successfully guides the colorization on specific and meaningful objects rather than a full reference image. Finally, both colorization frameworks achieve colorful and pleasant color generations and state-of-the-art results in different metrics.

Finally, our final contribution was proposed in Chapter 7, where we developed a conditional diffusion model for coloring line art. This model integrates artist guidance through color scribbles by successfully merging explicit user color inputs with the implicit generative knowledge within the diffusion model. We demonstrated the capacity of our model to generate diverse and high-quality line-art colorizations.

8.2 General future works

Multi-modal colorization framework As introduced in section 2.6, multi-modal colorization methods involve combining different user inputs alongside the grayscale image, such as:

- *Text descriptors:* This type of conditioning has been accurately and highly utilized by diffusion model architectures (Rombach et al., 2022). This type of conditioning in a colorization framework proposes a quick and coherent way to add colors to the image to colorize.

- *Color hints:* This conditioning combines human creativity's strengths and the deep learning models' precision. This allows users to input color hints or scribbles to guide the colorization more effectively and precisely.
- *Semantic cues:* Using semantic cues, such as segmentation masks or labels, might help to identify precisely an object and specific regions within an image. This conditioning could reduce color bleeding and identify the object by applying a more realistic color. In addition, regarding video colorization, the segmentation cues could help retain objects between frames.
- *Reference images:* This conditioning helps in particular where specific color patterns are desired or when dealing with complex scenes containing multiple objects and textures. For instance, colorizing a historical black-and-white photo of a city can be more accurately achieved by providing an actual colored picture of the same location as a reference image.

During this thesis we developed colorization frameworks that let the user interact with some of the previous mentioned inputs for example, using color scribles(Chapter 7), using reference image(Chapter 5) or combination between reference images and segmentation cues(Chapter 6). However, none of then can be consider a multi-modal colorization framework. Recently, efforts have been made in (Huang et al., 2022) where users can guide the colorization with text, reference images, color points, and their combinations (see Figure 8.1). However, in several cases, the method generated a mixture of colors due to the variety of input conditions. In addition, it is computationally expensive due to the two-stage framework and requires a large and varied number of datasets to train in a multi-modal manner.

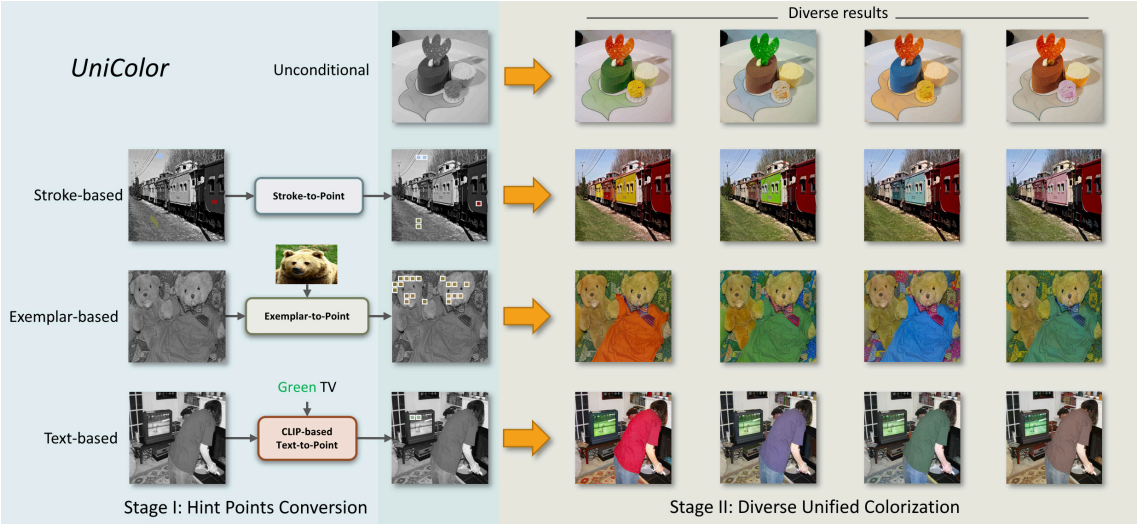


Figure 8.1: The multi-modal colorization framework from (Huang et al., 2022) consists of two stages. First, the method lets the user introduce combinations of color strokes, reference images, and text. Then, in the second stage, diverse results are generated automatically.

A solution for the mix of colors presented in (Huang et al., 2022) could be the addition of a segmentation mask to delimit the conflict of colors where two or more modalities are

presented. Finally, for the two-stage issue, we could jointly train an encoder that projects the conditioning (text, segmentation, strokes, etc.) to an intermediate latent representation and then introduce it into the main colorization network through attention mechanisms as done in (Rombach et al., 2022) and in (Zhang et al., 2023) for the task of image synthesis.

Video colorization using deep learning The video colorization process involves adding color to black and white videos/frames, traditionally requiring high manual intervention and effort by artists. This task combines the challenges of understanding the video sequence content and temporal coherence. Some works have attacked these challenges using deep learning methods. For example, (Zhang et al., 2019) proposes an exemplar-based colorization method based on attention mechanisms for transferring colors and retaining temporal coherence between adjacent frames. For that, several prior keyframes are colored beforehand and then used as a reference for colorizing similar frames. However, the calculation complexity of using attention blocks on videos is high; in addition, the results on old videos or documentaries are full of artifacts. Therefore in (Iizuka et Simo-Serra, 2019), the authors propose a fully convolutional framework capable of remastering and colorizing old videos. In this method, they first create a synthetic dataset of common artifacts presented in old films by emulating scratches, holes, and noise (See Figure 8.2). Second, the framework uses, at its core, combinations of temporal and spatial attention blocks to ensure the corresponding coherence. Despite the results being interesting, there is work to be done for this method to be used in a professional workplace.

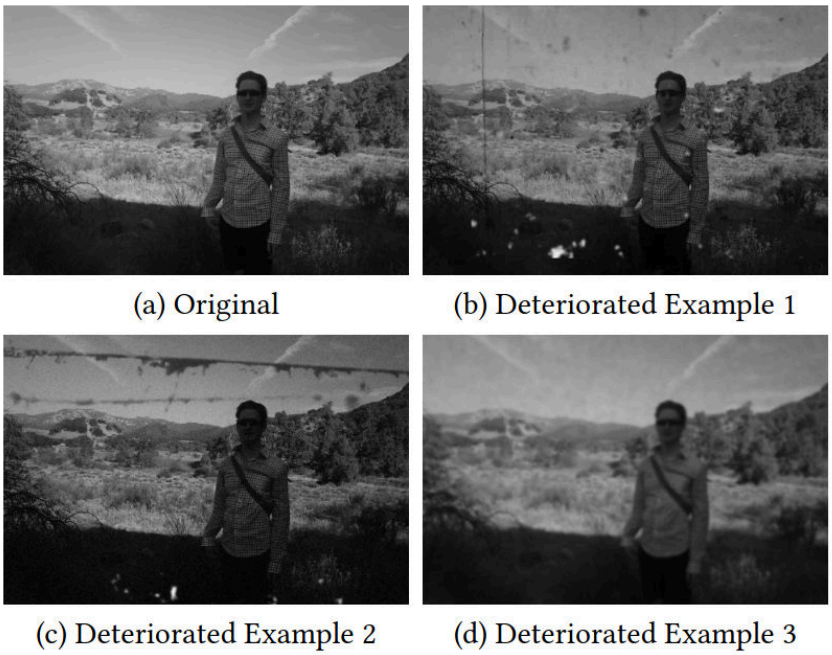


Figure 8.2: (a) Original image, (b-d) several artifacts effects are randomly added using image processing techniques, such as JPEG compression artifacts and film scratches.

As a future suggestion for this perspective, our super-attention method proposed in Chapter 4, and its extension to exemplar based-colorization in Chapter 5 and Chapter 6

could be extended to exemplar video colorization. In order to keep the temporal coherence of superpixels, we could use our super-attention jointly with the temporal superpixels method (Chang et al., 2013). This method is a generative probabilistic framework that explicitly models the motion flow between frames. As a result, they can track with the same superpixel object parts in adjacent frames. In addition, this could help with color bleeding artifacts with the advantage of less calculation complexity.

Inference time on diffusion models In Chapter 7 of this thesis, we introduced a novel method employing diffusion models for line-art image colorization. Our proposal showed promising results in terms of high quality and detail in the generated colorized line art. However, a significant limitation of our proposed approach is the inference time. In our experiments, generating a colorized image of size 256×256 pixels required approximately 25 seconds. This duration is substantially higher than other deep learning architectures for image generation. This longer inference time is usually attributed to the inherent calculation complexity of diffusion models and the series of forward and reverse diffusion steps for iteratively refining the image. However, the recent work in (Rombach et al., 2022) has improved the inference time by doing the diffusion process on the latent space instead of pixel space. This is done by using a pre-trained autoencoder to embed the input images into low-resolution features. This addition has been shown to enhance the efficacity of the models and reduce the time taken for image generation with a slight compromise in the output quality, reducing generation time to a couple of seconds for an image of size 512×512 pixels. Figure 8.3 shows the details of the architecture used in (Rombach et al., 2022).

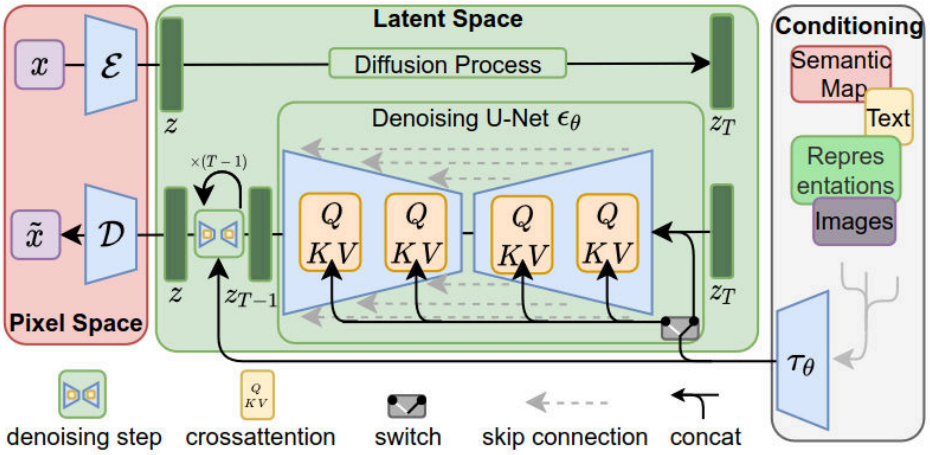


Figure 8.3: The method proposed by (Rombach et al., 2022) proposed an autoencoder to decrease the complexity by working on a reduced latent space instead of the pixel space. This addition reduces the impact of the inference time in the diffusion process on high-resolution images.

This approach could be a potential path for optimizing our proposed method for line art colorization, decreasing inference time, and, in addition, generating higher-resolution images. Finally, this addition could help us to extend our proposal using diffusion models to natural image colorization.

Bibliography

[Achanta et al. 2012] Achanta, Radhakrishna ; Shaji, Appu ; Smith, Kevin ; Lucchi, Aurelien ; Fua, Pascal ; Süsstrunk, Sabine : SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012), n. 11, pp. 2274–2282 viii, 28, 68, 69, 70, 75, 88, 97, 103

[Afifi et al. 2021] Afifi, Mahmoud ; Brubaker, Marcus A. ; Brown, Michael S. : HistoGAN: Controlling Colors of GAN-Generated and Real Images via Color Histograms. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7941–7950 62

[Agustsson et Timofte 2017] Agustsson, Eirikur ; Timofte, Radu : Ntire 2017 challenge on single image super-resolution: Dataset and study. In : *Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135 36, 38

[Anjos et Shahbazkia 2008] Anjos, António ; Shahbazkia, Hamid : Bi-Level Image Thresholding - A Fast Method. In : *International Conference on Bio-inspired Systems and Signal Processing*, 2008 97

[Antic 2019] Antic, Jason : *DeOldify*. `https://github.com/jantic/DeOldify`. 2019 26, 30, 35, 37, 46

[Anwar et al. 2020] Anwar, Saeed ; Tahir, Muhammad ; Li, Chongyi ; Mian, Ajmal ; Khan, Fahad S. ; Muzaffar, Abdul W. : Image colorization: A survey and dataset. In : *arXiv preprint arXiv:2008.10774* (2020) 37

[Arbelaez et al. 2009] Arbelaez, Pablo ; Maire, Michael ; Fowlkes, Charless ; Malik, Jitendra : From contours to regions: An empirical evaluation. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2009 68

[Arbelot et al. 2017] Arbelot, B. ; Vergne, R. ; Hurtut, T. ; Thollot, J. : Local texture-based color transfer and colorization. In : *Computers & Graphics* 62 (2017), pp. 15–27 70

[Arjovsky et al. 2017] Arjovsky, Martin ; Chintala, Soumith ; Bottou, Léon : Wasserstein Generative Adversarial Networks. In : *International Conference on Machine Learning* 70, 2017, pp. 214–223 44, 114

[Bahng et al. 2018] Bahng, Hyojin ; Yoo, Seungjoo ; Cho, Wonwoong ; Park, David K. ; Wu, Ziming ; Ma, Xiaojuan ; Choo, Jaegul : Coloring with words: Guiding image colorization through text-based palette generation. In : *European Conference on Computer Vision*, 2018, pp. 431–447 34, 41

[Barnes et al. 2009] Barnes, Connelly ; Shechtman, Eli ; Finkelstein, Adam ; Goldman, Dan B. : *PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing.* In : *ACM Trans. Graph.* 28, Association for Computing Machinery, 2009 32, 33

[Van den Bergh et al. 2015] Bergh, Michael Van den ; Boix, Xavier ; Roig, Gemma ; Van Gool, Luc : SEEDS: Superpixels Extracted via Energy-Driven Sampling. In : *International Journal of Computer Vision* 111 (2015), n. 3, pp. 298–314 68

[Blanch et al. 2021] Blanch, Marc G. ; Khalifeh, Issa ; Smeaton, Alan ; Connor, Noel E. ; Mrak, Marta : Attention-based Stylisation for Exemplar Image Colourisation. In : *IEEE International Workshop on Multimedia Signal Processing*, 2021, pp. 1–6 26, 33, 35, 37, 41, 43, 46, 49, 84, 88, 92, 93, 94, 149, 151

[Bookstein 1989] Bookstein, F.L. : Principal warps: thin-plate splines and the decomposition of deformations. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1989), pp. 567–585 102

[Buades et al. 2005] Buades, A. ; Coll, B. ; Morel, J.-M. : A non-local algorithm for image denoising. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 60–65 vol. 2 33, 65, 67

[Bugeau et Ta 2012] Bugeau, Aurélie ; Ta, Vinh-Thong : Patch-based image colorization. In : *International Conference on Pattern Recognition*, 2012, pp. 3058–3061 28

[Bugeau et al. 2014] Bugeau, Aurélie ; Ta, Vinh-Thong ; Papadakis, Nicolas : Variational Exemplar-Based Image Colorization. In : *IEEE Transactions on Image Processing* 23 (2014), n. 1, pp. 298–307 23, 29

[Cao et al. 2017] Cao, Yun ; Zhou, Zhiming ; Zhang, Weinan ; Yu, Yong : Unsupervised diverse colorization via generative adversarial networks. In : *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 151–166 26, 30, 35, 37, 41

[Chang et al. 2013] Chang, Jason ; Wei, Donglai ; Fisher III, John W. : A Video Representation Using Temporal Superpixels. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2051–2058 128

[Chang et al. 2023] Chang, Zheng ; Weng, Shuchen ; Zhang, Peixuan ; Li, Yu ; Li, Si ; Shi, Boxin : L-CoIns: Language-Based Colorization With Instance Awareness. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19221–19230 vii, 26, 34, 35, 36, 37, 41, 46, 49

[Charpiat et al. 2008] Charpiat, G. ; Hofmann, M. ; Schölkopf, B. : Automatic Image Colorization via Multimodal Predictions. In : *European Conference on Computer Vision*, 2008, pp. 126–139 29, 30

[Chen et al. 2021] Chen, Nanxin ; Zhang, Yu ; Zen, Heiga ; Weiss, Ron J. ; Norouzi, Mohammad ; Chan, William : WaveGrad: Estimating Gradients for Waveform Generation. In : *International Conference on Learning Representations*, 2021 117

[Cheng et al. 2015] Cheng, Zezhou ; Yang, Qingxiong ; Sheng, Bin : Deep colorization. In : *IEEE International Conference on Computer Vision*, 2015, pp. 415–423 3, 14, 26, 29, 35, 37, 41, 42, 46

[Cherel et al. 2024] Cherel, Nicolas ; Almansa, Andrés ; Gousseau, Yann ; Newson, Alasdair : Patch-based stochastic attention for image editing. In : *Computer Vision and Image Understanding* 238 (2024), pp. 103866 33

[Chia et al. 2011] Chia, Alex Yong-Sang ; Zhuo, Shaojie ; Gupta, Raj K. ; Tai, Yu-Wing ; Cho, Siu-Yeung ; Tan, Ping ; Lin, Stephen : Semantic Colorization with Internet Images. In : *ACM SIGGRAPH ASIA*, 2011, pp. 1–8 23, 28, 29

[Chong et Forsyth 2020] Chong, Min J. ; Forsyth, David : Effectively Unbiased FID and Inception Score and Where to Find Them. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6069–6078 39, 104

[Ci et al. 2018] Ci, Yuanzheng ; Ma, Xinzhu ; Wang, Zhihui ; Li, Haojie ; Luo, Zhongxuan : User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks. In : *ACM International Conference on Multimedia*, 2018 ix, 15, 113, 116, 120, 121

[CIE 1998] CIE, S : CIE standard illuminants for colorimetry. (1998) 25

[Comaniciu et Meer 2002] Comaniciu, D. ; Meer, P. : Mean shift: a robust approach toward feature space analysis. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002) 97

[Connolly et Fleiss 1997] Connolly, C. ; Fleiss, T. : A study of efficiency and accuracy in the transformation from RGB to CIELAB color space. In : *IEEE Transactions on Image Processing* (1997) 24, 85

[Dalal et Triggs 2005] Dalal, N. ; Triggs, B. : Histograms of oriented gradients for human detection. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2005 69

[DanbooruCommunity 2021] DanbooruCommunity : *Danbooru2021: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset.* 2021. – URL https://gwern.net/danbooru2021 118

[Deng et al. 2009] Deng, Jia ; Dong, Wei ; Socher, Richard ; Li, Li-Jia ; Li, Kai ; Fei-Fei, Li : Imagenet: A large-scale hierarchical image database. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255 36, 38

[Deshpande et al. 2017] Deshpande, Aditya ; Lu, Jiajun ; Yeh, Mao-Chuang ; Jin Chong, Min ; Forsyth, David : Learning diverse image colorization. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6837–6845 23, 26, 31, 35, 37, 41, 46

[Deshpande et al. 2015] Deshpande, Aditya ; Rock, Jason ; Forsyth, David : Learning Large-Scale Automatic Image Colorization. In : *IEEE International Conference on Computer Vision*, 2015, pp. 567–575 29

[Dhariwal et Nichol 2021] Dhariwal, Prafulla ; Nichol, Alexander : Diffusion models beat gans on image synthesis. In : *Advances in Neural Information Processing Systems* (2021) 114, 116

[Di Blasi et Reforgiato 2003] Di Blasi, Gianpiero ; Reforgiato, Diego : Fast colorization of gray images. In : *Eurographics Italian* (2003) 28

[Ding et al. 2021] Ding, Keyan ; Ma, Kede ; Wang, Shiqi ; Simoncelli, Eero P. : Comparison of full-reference image quality models for optimization of image processing systems. In : *International Journal of Computer Vision* 129 (2021), n. 4, pp. 1258–1281 53

[Ding et al. 2012] Ding, Xiaowei ; Xu, Yi ; Deng, Lei ; Yang, Xiaokang : Colorization Using Quaternion Algebra with Automatic Scribble Generation. In : *Advances in Multimedia Modeling*, 2012, pp. 103–114 27

[Dosovitskiy et al. 2021] Dosovitskiy, Alexey ; Beyer, Lucas ; Kolesnikov, Alexander ; Weissenborn, Dirk ; Zhai, Xiaohua ; Unterthiner, Thomas ; Dehghani, Mostafa ; Minderer, Matthias ; Heigold, Georg ; Gelly, Sylvain ; Uszkoreit, Jakob ; Houlsby, Neil : An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In : *International Conference on Learning Representations*, 2021 34

[Dowson et Landau 1982] Dowson, D.C ; Landau, B. V. : The Fréchet distance between multivariate normal distributions. In : *Journal of Multivariate Analysis* 12 (1982), pp. 450–455 39, 53

[Drew et Finlayson 2011] Drew, Mark S. ; Finlayson, Graham D. : Improvement of Colorization Realism via the Structure Tensor. In : *International Journal on Image Graphics* 11 (2011), n. 4, pp. 589–609 27

[Efros et Leung 1999] Efros, A. ; Leung, T. : Texture Synthesis by Non-parametric Sampling. In : *IEEE International Conference on Computer Vision*, 1999, pp. 1033–1038 27

[Fang et al. 2019] Fang, Faming ; Wang, Tingting ; Zeng, Tieyong ; Zhang, Guixu : A superpixel-based variational model for image colorization. In : *IEEE Transactions on Visualization and Computer Graphics* 26 (2019), n. 10, pp. 2931–2943 29

[Ferradans et al. 2013] Ferradans, Sira ; Papadakis, Nicolas ; Rabin, Julien ; Peyré, Gabriel ; Aujol, Jean-François : Regularized Discrete Optimal Transport. In : *Scale Space and Variational Methods in Computer Vision*, 2013 70

[Frigo et al. 2015] Frigo, Oriel ; Sabater, Neus ; Demoulin, Vincent ; Hellier, Pierre : Optimal Transportation for Example-Guided Color Transfer. In : *Asian Conference on Computer Vision*, 2015 70

[Fulkerson et al. 2009] Fulkerson, Brian ; Vedaldi, Andrea ; Soatto, Stefano : Class segmentation and object localization with superpixel neighborhoods. In : *IEEE International Conference on Computer Vision*, 2009 68

[Furusawa et al. 2017] Furusawa, Chie ; Hiroshiba, Kazuyuki ; Ogaki, Keisuke ; Odagiri, Yuri : Comicolorization: semi-automatic manga colorization. In : *International Conference on Computer Graphics and Interactive Techniques*, 2017, pp. 1–4 116

[Gatys et al. 2016] Gatys, Leon A. ; Ecker, Alexander S. ; Bethge, Matthias : A Neural Algorithm of Artistic Style. In : *Journal of Vision* 16 (2016), September, n. 12, pp. 326 42

[Giraud et al. 2017] Giraud, Rémi ; Ta, Vinh-Thong ; Papadakis, Nicolas : Superpixel-based color transfer. In : *IEEE International Conference on Image Processing*, 2017, pp. 700–704 viii, 7, 18, 64, 65, 70, 75, 76, 77, 78, 79, 80

[Glasner et al. 2009] Glasner, Daniel ; Bagon, Shai ; Irani, Michal : Super-resolution from a single image. In : *IEEE International Conference on Computer Vision*, 2009 66

[Goodfellow et al. 2014] Goodfellow, Ian ; Pouget-Abadie, Jean ; Mirza, Mehdi ; Xu, Bing ; Warde-Farley, David ; Ozair, Sherjil ; Courville, Aaron ; Bengio, Yoshua : Generative Adversarial Nets. In : *Advances in Neural Information Processing Systems*, 2014 30, 43

[Gu et al. 2019] Gu, Shuhang ; Timofte, Radu ; Zhang, Richard : NTIRE 2019 Challenge on Image Colorization: Report. In : *Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 2233–2240 26, 35, 37, 41, 46

[Guadarrama et al. 2017] Guadarrama, Sergio ; Dahl, Ryan ; Bieber, David ; Norouzi, Mohammad ; Shlens, Jonathon ; Murphy, Kevin : Pixcolor: Pixel recursive colorization. In : *British Machine Vision Conference*, 2017 26, 31, 35, 37, 41, 46, 144

[Gulrajani et al. 2017] Gulrajani, Ishaan ; Ahmed, Faruk ; Arjovsky, Martin ; Dumoulin, Vincent ; Courville, Aaron : Improved Training of Wasserstein GANs. In : *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779 44, 114

[Gupta et al. 2012] Gupta, Raj K. ; Chia, Alex Yong-Sang ; Rajan, Deepu ; Ng, Ee S. ; Zhiyong, Huang : Image colorization using similar images. In : *ACM International Conference on Multimedia*, 2012, pp. 369–378 28, 97

[He et al. 2017] He, Kaiming ; Gkioxari, Georgia ; Dollár, Piotr ; Girshick, Ross : Mask R-CNN. In : *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969 31

[He et al. 2018] He, Mingming ; Chen, Dongdong ; Liao, Jing ; Sander, Pedro V. ; Yuan, Lu : Deep exemplar-based colorization. In : *ACM Transactions on Graphics* 37 (2018), n. 4, pp. 1–16 vii, 3, 14, 26, 32, 33, 35, 37, 41, 42, 46, 49, 54, 89, 92, 93, 94, 149, 151

[He et al. 2019] He, Mingming ; Liao, Jing ; Chen, Dongdong ; Yuan, Lu ; Sander, Pedro V. : Progressive Color Transfer With Dense Semantic Correspondences. In : *ACM Transactions on Graphics* 38 (2019), n. 2 65, 71

[He et al. 2015] He, Shengfeng ; Lau, Rynson ; Liu, Wenxi ; Huang, Zhe ; Yang, Qingxiong : SuperCNN: A superpixelwise convolutional neural network for salient object detection. In : *International Journal of Computer Vision* 115 (2015), pp. 330–344 69

[Heu et al. 2009] Heu, Junhee ; Hyun, Dae-Young ; Kim, Chang-Su ; Lee, Sang-Uk : Image and video colorization based on prioritized source propagation. In : *IEEE International Conference on Image Processing*, 2009, pp. 465–468 27, 110

[Heusel et al. 2017] Heusel, Martin ; Ramsauer, Hubert ; Unterthiner, Thomas ; Nessler, Bernhard ; Hochreiter, Sepp : GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In : *Advances in Neural Information Processing Systems* 30 (2017) 39, 104, 120

[Ho et al. 2020] Ho, Jonathan ; Jain, Ajay ; Abbeel, Pieter : Denoising Diffusion Probabilistic Models. In : *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020 114, 116, 117, 118, 119

[Ho et al. 2019] Ho, Jonathan ; Kalchbrenner, Nal ; Weissenborn, Dirk ; Salimans, Tim : Axial attention in multidimensional transformers. In : *arXiv preprint arXiv:1912.12180* (2019) 31

[Huang et Belongie 2017] Huang, Xun ; Belongie, Serge : Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In : *IEEE International Conference on Computer Vision*, 2017, pp. 1510–1519 32

[Huang et al. 2005] Huang, Yi-Chin ; Tung, Yi-Shin ; Chen, Jun-Cheng ; Wang, Sung-Wen ; Wu, Ja-Ling : An adaptive edge detection based colorization algorithm and its applications. In : *ACM international conference on Multimedia*, 2005, pp. 351–354 23, 27

[Huang et al. 2022] Huang, Zhitong ; Zhao, Nanxuan ; Liao, Jing : UniColor: A Unified Framework for Multi-Modal Colorization with Transformer. In : *ACM Transactions on Graphics* 41 (2022) ix, 2, 3, 13, 14, 24, 26, 34, 35, 37, 41, 42, 46, 49, 108, 109, 110, 126

[Ihsan et al. 2020] Ihsan, A. ; Chu Kiong, Loo ; Naji, Sinan ; Seera, Manjeevan : Superpixels Features Extractor Network (SP-FEN) for Clothing Parsing Enhancement. In : *Neural Processing Letters* 51 (2020), pp. 2245–2263 69

[Iizuka et Simo-Serra 2019] Iizuka, Satoshi ; Simo-Serra, Edgar : DeepRemaster: Temporal Source-Reference Attention Networks for Comprehensive Video Enhancement. In : *International Conference on Computer Graphics and Interactive Techniques* (2019), pp. 1–13 ix, 127

[Iizuka et al. 2016] Iizuka, Satoshi ; Simo-Serra, Edgar ; Ishikawa, Hiroshi : Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. In : *ACM Transactions on Graphics* 35 (2016), n. 4 3, 14, 26, 30, 35, 37, 41, 46, 51, 144

[Irony et al. 2005] Irony, Revital ; Cohen-Or, Daniel ; Lischinski, Dani : Colorization by example. In : *Eurographics conference on Rendering Techniques*, 2005, pp. 201–210 28, 97

[Irving 2016] Irving, Benjamin : maskSLIC: regional superpixel generation with application to local pathology characterisation in medical images. In : *arXiv preprint arXiv:1606.09518* (2016) 101, 103

[Isola et al. 2017] Isola, Phillip ; Zhu, Jun-Yan ; Zhou, Tinghui ; Efros, Alexei A. : Image-to-Image Translation with Conditional Adversarial Networks. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134 30, 40, 90, 105

[Jin et al. 2021] Jin, Xin ; Li, Zhonglan ; Liu, Ke ; Zou, Dongqing ; Li, Xiaodong ; Xingfan Zhu, Ziyin Z. ; Sun, Qilong ; Liu, Qingyu : Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies. In : *ACM International Conference on Multimedia*, 2021 36

[Johnson et al. 2016] Johnson, Justin ; Alahi, Alexandre ; Fei-Fei, Li : Perceptual losses for real-time style transfer and super-resolution. In : *European Conference on Computer Vision*, 2016, pp. 694–711 42, 53

[Kawulok et al. 2012] Kawulok, Michal ; Kawulok, Jolanta ; Smolka, Bogdan : Discriminative Textural Features for Image and Video Colorization. In : *IEICE Transaction on Information and Systems* 95-D (2012), n. 7, pp. 1722–1730 27

[Kong et al. 2021] Kong, Guangqian ; Tian, Huan ; Duan, Xun ; Long, Huiyun : Adversarial Edge-Aware Image Colorization With Semantic Segmentation. In : *IEEE Access* 9 (2021), pp. 28194–28203 26, 31, 37, 41, 46, 55

[Krizhevsky et al. 2009] Krizhevsky, Alex ; Hinton, Geoffrey et al. : Learning multiple layers of features from tiny images / University of Toronto. 2009. – Forschungsbericht 36

[Kumar et al. 2021] Kumar, Manoj ; Weissenborn, Dirk ; Kalchbrenner, Nal : Colorization Transformer. In : *International Conference on Learning Representations*, 2021 26, 31, 35, 37, 41, 46, 144

[Lagodzinski et Smolka 2008] Lagodzinski, Przemyslaw ; Smolka, Bogdan : Digital image colorization based on probabilistic distance transformation. In : *50th International Symposium ELMAR* 2, Sept 2008, pp. 495–498 27

[Larsson et al. 2016] Larsson, Gustav ; Maire, Michael ; Shakhnarovich, Gregory : Learning representations for automatic colorization. In : *European Conference on Computer Vision*, 2016, pp. 577–593 3, 14, 26, 30, 35, 37, 41, 46, 144

[Lee et al. 2020a] Lee, Junsoo ; Kim, Eungyeup ; Lee, Yunsung ; Kim, Dongjun ; Chang, Jaehyuk ; Choo, Jaegul : Reference-Based Sketch Image Colorization Using Augmented-Self Reference and Dense Semantic Correspondence. In : *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020 102, 116

[Lee et al. 2020b] Lee, Junyong ; Son, Hyeongseok ; Lee, Gunhee ; Lee, Jonghyeop ; Cho, Sunghyun ; Lee, Seungyong : Deep Color Transfer using Histogram Analogy. In : *The Visual Computer* 36 (2020), n. 10, pp. 2129–2143 65, 70, 77, 78, 79, 80

[Levin et al. 2004] Levin, Anat ; Lischinski, Dani ; Weiss, Yair : Colorization using optimization. In : *ACM Transactions on Graphics* 23 (2004), n. 3, pp. 689–694 vii, 4, 15, 23, 27, 28

[Lézoray et al. 2008] Lézoray, Olivier ; Ta, Vinh-Thong ; Elmoataz, Abderrahim : Nonlocal graph regularization for image colorization. In : *International Conference on Pattern Recognition*, 2008, pp. 1–4 27

[Li et al. 2019] Li, Bo ; Lai, Yu-Kun ; John, Matthew ; Rosin, Paul L. : Automatic example-based image colorization using location-aware cross-scale matching. In : *IEEE Transactions on Image Processing* 28 (2019), n. 9, pp. 4606–4619 29

[Li et al. 2017a] Li, Bo ; Lai, Yu-Kun ; Rosin, Paul L. : Example-based image colorization via automatic feature selection and fusion. In : *Neurocomputing* 266 (2017), pp. 687–698 29

[Li et al. 2017b] Li, Bo ; Zhao, Fuchen ; Su, Zhuo ; Liang, Xiangguo ; Lai, Yu-Kun ; Rosin, Paul L. : Example-based image colorization using locality consistent sparse representation. In : *IEEE Transactions on Image Processing* 26 (2017), n. 11, pp. 5188–5202 28, 97

[Li et al. 2022a] Li, Haoying ; Yang, Yifan ; Chang, Meng ; Chen, Shiqi ; Feng, Huajun ; Xu, Zhihai ; Li, Qi ; Chen, Yueting : SRDiff: Single image super-resolution with diffusion probabilistic models. In : *Neurocomputing* (2022), pp. 47–59 116

[Li et al. 2022b] Li, Zekun ; Geng, Zhengyang ; Kang, Zhao ; Chen, Wenyu ; Yang, Yibo : Eliminating Gradient Conflict in Reference-based Line-Art Colorization. In : *European Conference on Computer Vision*, 2022, pp. 579–596 116

[Lin et al. 2014] Lin, Tsung-Yi ; Maire, Michael ; Belongie, Serge ; Hays, James ; Perona, Pietro ; Ramanan, Deva ; Dollár, Piotr ; Zitnick, C L. : Microsoft COCO: Common objects in context. In : *European Conference on Computer Vision*, 2014, pp. 740–755 36, 38, 51, 89, 102

[Lindeberg et Li 1997] Lindeberg, Tony ; Li, Meng-Xiang : Segmentation and classification of edges using minimum description length approximation and complementary junction cues. In : *Computer vision and image understanding* (1997) 97

[Ling et al. 2015] Ling, Yonggen ; Au, Oscar C. ; Pang, Jiahao ; Zeng, Jin ; Yuan, Yuan ; Zheng, Amin : Image colorization via color propagation and rank minimization. In : *IEEE International Conference on Image Processing*, 2015, pp. 4228–4232 27

[Liu et al. 2016] Liu, Jiaying ; Yang, Wenhan ; Sun, Xiaoyan ; Zeng, Wenjun : Photo Stylistic Brush: Robust Style Transfer via Superpixel-Based Bipartite Graph. In : *IEEE International Conference on Multimedia and Expo*, 2016 70

[Liu et Zhang 2012] Liu, Shiguang ; Zhang, Xiang : Automatic grayscale image colorization using histogram regression. In : *Pattern Recognition Letters* 33 (2012), n. 13, pp. 1673–1681 28

[Liu et al. 2018] Liu, Yifan ; Qin, Zengchang ; Wan, Tao ; Luo, Zhenbo : Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks. In : *Neurocomputing* 311 (2018), pp. 78–87 116

[lllyasviel 2017] lllyasviel : *sketchKeras*. 2017. – URL `https://github.com/lllyasviel/sketchKeras` ix, 119

[Lowe 2004] Lowe, David G. : Distinctive image features from scale-invariant keypoints. In : *International Journal of Computer Vision* 60 (2004), n. 2, pp. 91–110 69

[Lu et al. 2020] Lu, Peng ; Yu, Jinbei ; Peng, Xujun ; Zhao, Zhaoran ; Wang, Xiaojie : Gray2ColorNet: Transfer More Colors from Reference Image. In : *ACM International Conference on Multimedia*, 2020, pp. 3210–3218 26, 33, 46, 88, 108, 109, 149

[Luan et al. 2007] Luan, Qing ; Wen, Fang ; Cohen-Or, Daniel ; Liang, Lin ; Xu, Ying-Qing ; Shum, Heung-Yeung : Natural image colorization. In : *Eurographics conference on Rendering Techniques*, 2007, pp. 309–320 27

[Ma et al. 2018] Ma, Shuang ; Fu, Jianlong ; Chen, Chang W. ; Mei, Tao : Da-gan: Instance-level image translation by deep attention generative adversarial networks. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2018 97

[MacAdam 1937] MacAdam, David L. : Projective Transformations of I. C. I. Color Specifications. In : *J. Opt. Soc. Am.* (1937), pp. 294–299 25

[Manjunatha et al. 2018] Manjunatha, Varun ; Iyyer, Mohit ; Boyd-Graber, Jordan ; Davis, Larry : Learning to Color from Language. In : *North American Chapter of the Association for Computational Linguistics*, 2018 34

[Mechrez et al. 2018] Mechrez, Roey ; Talmi, Itamar ; Zelnik-Manor, Lihi : The contextual loss for image transformation with non-aligned data. In : *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 768–783 40, 104

[Mouzon et al. 2019] Mouzon, Thomas ; Pierre, Fabien ; Berger, Marie-Odile : Joint CNN and variational model for fully-automatic image colorization. In : *Scale Space and Variational Methods in Computer Vision*, 2019, pp. 535–546 26, 30, 35, 46

[Murray et al. 2012] Murray, Naila ; Skaff, Sandra ; Marchesotti, Luca ; Perronnin, Florent : Toward automatic and flexible concept transfer. In : *Computers & Graphics* 36 (2012), n. 6, pp. 622–634 70

[Nazeri et al. 2018] Nazeri, Kamyar ; Ng, Eric ; Ebrahimi, Mehran : Image colorization using generative adversarial networks. In : *International Conference on Articulated Motion and Deformable Objects*, 2018, pp. 85–94 30, 37, 41, 43, 46

[Oord et al. 2016] Oord, Aaron van d. ; Kalchbrenner, Nal ; Vinyals, Oriol ; Espeholt, Lasse ; Graves, Alex ; Kavukcuoglu, Koray : Conditional image generation with PixelCNN decoders. In : *Advances in Neural Information Processing Systems* (2016) 31

[Pang et al. 2013] Pang, Jiahao ; Au, Oscar C. ; Tang, Ketan ; Guo, Yuanfang : Image colorization using sparse representation. In : *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 1578–1582 28

[Pierre et al. 2015a] Pierre, F. ; Aujol, J.-F. ; Bugeau, A. ; Papadakis, N. ; Ta, V.-T. : Luminance-Chrominance Model for Image Colorization. In : *SIAM Journal on Imaging Sciences* 8 (2015), jan, n. 1, pp. 536–563 ix, 148, 149, 150, 151

[Pierre et al. 2014] Pierre, Fabien ; Aujol, Jean-François ; Bugeau, Aurélie ; Ta, Vinh-Thong : A Unified Model for Image Colorization. In : *European Conference on Computer Vision Workshops*, 2014, pp. 297–308 29, 49

[Pierre et al. 2015b] Pierre, Fabien ; Aujol, Jean-François ; Bugeau, Aurélie ; Ta, Vinh-Thong : Luminance-Hue Specification in the RGB Space. In : *Scale Space and Variational Methods in Computer Vision*, 2015, pp. 413–424 62, 110

[Pitié et Kokaram 2007] Pitié, F. ; Kokaram, A. : The linear Monge-Kantorovitch linear colour mapping for example-based colour transfer. In : *European Conference on Visual Media Production*, 2007 69, 77

[Pitié et al. 2005] Pitié, F. ; Kokaram, A.C. ; Dahyot, R. : Towards automated colour grading. In : *IEEE European Conference on Visual Media Production*, 2005. – ISSN 0537-9989 75, 76

[Pitié 2020] Pitié, Francois : Advances in colour transfer. In : *IET Computer Vision* 14 (2020), pp. 304–322 70

[Pitié et al. 2007] Pitié, François ; Kokaram, Anil C. ; Dahyot, Rozenn : Automated colour grading using colour distribution transfer. In : *Computer vision and image understanding* 107 (2007), n. 1–2, pp. 123–137 77, 78, 79, 80

[Pucci et al. 2021] Pucci, Rita ; Micheloni, Christian ; Martinel, Niki : Collaborative Image and Object Level Features for Image Colourisation. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2160–2169 26, 31, 35, 37, 41, 46, 49

[Puzicha et al. 1997] Puzicha, J. ; Hofmann, T. ; Buhmann, J.M. : Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In : *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 267–272 40

[Qu et al. 2006] Qu, Yingge ; Wong, Tien-Tsin ; Heng, Pheng-Ann : Manga Colorization. In : *ACM Transactions on Graphics* 25 (2006), n. 3. – ISSN 0730-0301 23, 116

[Reinhard et al. 2001] Reinhard, Erik ; Ashikhmin, Michael ; Gooch, Bruce ; Shirley, Peter : Color Transfer between Images. In : *ACM Transactions on Graphics* 21 (2001), n. 5, pp. 34–41 69

[Ren et Malik 2003] Ren, Xiaofeng ; Malik, Jitendra : Learning a classification model for segmentation. In : *IEEE International Conference on Computer Vision*, 2003, pp. 10–17 28

[Reso et al. 2013] Reso, Matthias ; Jachalsky, Jörn ; Rosenhahn, Bodo ; Ostermann, Jörn : Temporally Consistent Superpixels, 2013 68

[Revoy 2022] Revoy, David : *Episode 37 Production report, part 2.* 2022. – URL `https://www.davidrevoy.com/` viii, 113, 114

[Riba et al. 2020] Riba, Edgar ; Mishkin, Dmytro ; Ponsa, Daniel ; Rublee, Ethan ; Bradski, Gary : Kornia: An open source differentiable computer vision library for PyTorch. In : *Winter Conference on Applications of Computer Vision*, 2020, pp. 3674–3683 26, 53, 59, 60, 85

[Rombach et al. 2022] Rombach, Robin ; Blattmann, Andreas ; Lorenz, Dominik ; Esser, Patrick ; Ommer, Björn : High-resolution image synthesis with latent diffusion models. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2022 ix, 114, 117, 121, 125, 127, 128

[Ronneberger et al. 2015] Ronneberger, Olaf ; Fischer, Philipp ; Brox, Thomas : U-net: Convolutional networks for biomedical image segmentation. In : *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241 85, 114, 148

[Royer et al. 2017] Royer, Amelie ; Kolesnikov, Alexander ; Lampert, Christoph H. : Probabilistic image colorization. In : *British Machine Vision Conference* (2017) 26, 37, 41, 144, 145

[Saharia et al. 2022a] Saharia, Chitwan ; Chan, William ; Chang, Huiwen ; Lee, Chris ; Ho, Jonathan ; Salimans, Tim ; Fleet, David ; Norouzi, Mohammad : Palette: Image-to-Image Diffusion Models. In : *ACM International Conference on Multimedia*, 2022 114, 116, 117

[Saharia et al. 2022b] Saharia, Chitwan ; Ho, Jonathan ; Chan, William ; Salimans, Tim ; Fleet, David J. ; Norouzi, Mohammad : Image super-resolution via iterative refinement. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022) 114, 116

[Sangkloy et al. 2017] Sangkloy, Patsorn ; Lu, Jingwan ; Fang, Chen ; Yu, Fisher ; Hays, James : Scribbler: Controlling deep image synthesis with sketch and color. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2017 113

[Sezgin et Sankur 2004] Sezgin, M. ; Sankur, Bulent : Survey over image thresholding techniques and quantitative performance evaluation. In : *Journal of Electronic Imaging* (2004) 97

[Shen et al. 2019] Shen, Zhiqiang ; Huang, Mingyang ; Shi, Jianping ; Xue, Xiangyang ; Huang, Thomas S. : Towards instance-level image-to-image translation. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2019 97

[Simo-Serra et al. 2018] Simo-Serra, Edgar ; Iizuka, Satoshi ; Ishikawa, Hiroshi : Mastering Sketching: Adversarial Augmentation for Structured Prediction. In : *Transactions on Graphics* (2018) ix, 119

[Simonyan et Zisserman 2015] Simonyan, Karen ; Zisserman, Andrew : Very Deep Convolutional Networks for Large-Scale Image Recognition. In : *International Conference on Learning Representations*, 2015 50, 76, 89

[Singh et al. 2019] Singh, Krishna K. ; Ojha, Utkarsh ; Lee, Yong J. : Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2019 97

[Song et Ermon 2019] Song, Yang ; Ermon, Stefano : Generative Modeling by Estimating Gradients of the Data Distribution. In : *Advances in Neural Information Processing Systems*, 2019 114, 116, 117

[Su et al. 2020] Su, Jheng-Wei ; Chu, Hung-Kuo ; Huang, Jia-Bin : Instance-aware image colorization. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7968–7977 vii, 26, 31, 32, 35, 37, 41, 42, 46, 55, 60, 98

[Sýkora et al. 2004] Sýkora, Daniel ; Buriánek, Jan ; Žára, Jiří : Unsupervised Colorization of Black-and-White Cartoons. In : *International symposium on Non-photorealistic animation and rendering*, 2004 97

[Sýkora et al. 2009] Sýkora, Daniel ; Dingliana, John ; Collins, Steven : LazyBrush: Flexible Painting Tool for Hand-drawn Cartoons. In : *Computer Graphics Forum* 28 (2009) 116

[Tai et al. 2005] Tai, Yu-Wing ; Jia, Jiaya ; Tang, Chi-Keung : Local color transfer via probabilistic segmentation by expectation-maximization. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 747–754 28, 70

[Tighe et Lazebnik 2010] Tighe, Joseph ; Lazebnik, Svetlana : SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. In : *European Conference on Computer Vision*, 2010 68

[Vaswani et al. 2017] Vaswani, Ashish ; Shazeer, Noam ; Parmar, Niki ; Uszkoreit, Jakob ; Jones, Llion ; Gomez, Aidan N. ; Kaiser, undefinedukasz ; Polosukhin, Illia : Attention is All You Need. In : *Advances in Neural Information Processing Systems*, 2017 32, 66, 67, 117

[Vitoria et al. 2020] Vitoria, Patricia ; Raad, Lara ; Ballester, Coloma : ChromaGAN: Adversarial picture colorization with semantic class distribution. In : *Winter Conference on Applications of Computer Vision*, 2020, pp. 2445–2454 vii, 3, 14, 23, 26, 30, 35, 37, 41, 42, 43, 46, 49, 50, 55, 60, 144

[Wan et al. 2020] Wan, Shaohua ; Xia, Yu ; Qi, Lianyong ; Yang, Yee-Hong ; Atiquzzaman, Mohammed : Automated colorization of a grayscale image with seed points propagation. In : *IEEE Transactions on Multimedia* 22 (2020), n. 7, pp. 1756–1768 29, 35

[Wang et Zhang 2012] Wang, Shusen ; Zhang, Zhihua : Colorization by matrix completion. In : *AAAI Conference on Artificial Intelligence*, 2012 27

[Wang et al. 2018a] Wang, Ting-Chun ; Liu, Ming-Yu ; Zhu, Jun-Yan ; Tao, Andrew ; Kautz, Jan ; Catanzaro, Bryan : High-resolution image synthesis and semantic manipulation with conditional gans. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2018 97

[Wang et al. 2018b] Wang, Xiaolong ; Girshick, Ross ; Gupta, Abhinav ; He, Kaiming : Non-local Neural Networks. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2018 viii, 33, 66, 67

[Wang et al. 2004] Wang, Zhou ; Bovik, Alan C. ; Sheikh, Hamid R. ; Simoncelli, Eero P. : Image quality assessment: from error visibility to structural similarity. In : *IEEE Transactions on Image Processing* 13 (2004), n. 4, pp. 600–612 53, 90, 104, 120

[Welsh et al. 2002] Welsh, Tomihisa ; Ashikhmin, Michael ; Mueller, Klaus : Transferring color to greyscale images. In : *ACM Transactions on Graphics* 21 (2002), n. 3, pp. 277–280 ix, 23, 27, 148, 149, 150, 151

[Wexler et al. 2004] Wexler, Y. ; Shechtman, E. ; Irani, M. : Space-time video completion. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2004 66

[Xiao et al. 2010] Xiao, Jianxiong ; Hays, James ; Ehinger, Krista A. ; Oliva, Aude ; Torralba, Antonio : Sun database: Large-scale scene recognition from abbey to zoo. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3485–3492 36

[Xiao et Ma 2006] Xiao, Xuezhong ; Ma, Lizhuang : Color transfer in correlated color space. In : *International Conference on Virtual Reality Continuum and its Applications*, 2006 69

[Xu et al. 2023] Xu, R. ; Tu, Z. ; Du, Y. ; Dong, X. ; Li, J. ; Meng, Z. ; Ma, J. ; Bovik, A. ; Yu, H. : Pik-Fix: Restoring and Colorizing Old Photos. In : *Winter Conference on Applications of Computer Vision*, 2023, pp. 1724–1734 36

[Xu et al. 2020] Xu, Zhongyou ; Wang, Tingting ; Fang, Faming ; Sheng, Yun ; Zhang, Guixu : Stylization-Based Architecture for Fast Deep Exemplar Colorization. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9360–9369 32

[Yang et al. 2010] Yang, Yi ; Hallman, Sam ; Ramanan, Deva ; Fowlkes, Charless : Layered object detection for multi-class segmentation. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2010 68

[Yao et James 2015] Yao, Quanming ; James, T K. : Colorization by patch-based local low-rank matrix completion. In : *AAAI Conference on Artificial Intelligence*, 2015, pp. 1959–1965 27

[Yatziv et Sapiro 2006] Yatziv, Liron ; Sapiro, Guillermo : Fast image and video colorization using chrominance blending. In : *IEEE Transactions on Image Processing* 15 (2006), n. 5, pp. 1120–1129 27

[Yin et al. 2021] Yin, Wang ; Lu, Peng ; Zhao, Zhaoran ; Peng, Xujun : Yes, "Attention Is All You Need", for Exemplar Based Colorization. In : *ACM International Conference on Multimedia*, 2021, pp. 2243–2251 26, 33, 35, 37, 41, 42, 43, 46, 54, 84, 88, 92, 93, 94, 108, 109, 144, 149, 151

[Yliess et al. 2019] Yliess, Hati ; Jouet, Gergor ; Rousseaux, Francis ; Duhrat, Clement : PaintsTorch: A User-Guided Anime Line Art Colorization Tool with Double Generator Conditional Adversarial Network. In : *ACM Eur. Conf. Visual Media Production*, 2019 viii, ix, 113, 115, 116, 120, 121

[Yoo et al. 2019] Yoo, Seungjoo ; Bahng, Hyojin ; Chung, Sunghyo ; Lee, Junsoo ; Chang, Jaehyuk ; Choo, Jaegul : Coloring with limited data: Few-shot colorization via memory augmented networks. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11283–11292 26, 37, 46

[Yu et al. 2015] Yu, Fisher ; Seff, Ari ; Zhang, Yinda ; Song, Shuran ; Funkhouser, Thomas ; Xiao, Jianxiong : LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. In : *arXiv preprint arXiv:1506.03365* (2015) 36

[Yuan et Simo-Serra 2021] Yuan, Mingcheng ; Simo-Serra, Edgar : Line Art Colorization with Concatenated Spatial Attention. In : *Conference on Computer Vision and Pattern Recognition Workshops*, 2021 113, 116

[Yun et al. 2023] Yun, Jooyeol ; Lee, Sanghyeon ; Park, Minho ; Choo, Jaegul : iColoriT: Towards Propagating Local Hints to the Right Region in Interactive Colorization by Leveraging Vision Transformer. In : *IEEE Conference on Computer Vision and Pattern Recognition*, January 2023, pp. 1787–1796 4, 15, 26, 34, 35, 37, 41, 46, 49

[Zhang et al. 2019] Zhang, Bo ; He, Mingming ; Liao, Jing ; Sander, Pedro V. ; Yuan, Lu ; Bermak, Amine ; Chen, Dong : Deep exemplar-based video colorization. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2019 67, 72, 73, 83, 84, 86, 127

[Zhang et al. 2021] Zhang, Lvmin ; Li, Chengze ; Simo-Serra, Edgar ; Ji, Yi ; Wong, Tien-Tsin ; Liu, Chunping : User-Guided Line Art Flat Filling with Split Filling Mechanism. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2021 110, 113

[Zhang et al. 2018a] Zhang, Lvmin ; Li, Chengze ; Wong, Tien-Tsin ; Ji, Yi ; Liu, Chunping : Two-stage sketch colorization. In : *ACM Transactions on Graphics* (2018) 116

[Zhang et al. 2023] Zhang, Lvmin ; Rao, Anyi ; Agrawala, Maneesh : Adding Conditional Control to Text-to-Image Diffusion Models. In : *IEEE International Conference on Computer Vision*, 2023 127

[Zhang et al. 2016] Zhang, Richard ; Isola, Phillip ; Efros, Alexei A. : Colorful image colorization. In : *European Conference on Computer Vision*, 2016, pp. 649–666 vii, 23, 26, 30, 31, 35, 37, 41, 42, 46, 144

[Zhang et al. 2018b] Zhang, Richard ; Isola, Phillip ; Efros, Alexei A. ; Shechtman, Eli ; Wang, Oliver : The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In : *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595 40, 53, 90, 104, 120

[Zhang et al. 2017] Zhang, Richard ; Zhu, Jun-Yan ; Isola, Phillip ; Geng, Xinyang ; Lin, Angela S. ; Yu, Tianhe ; Efros, Alexei A. : Real-time user-guided image colorization with learned deep priors. In : *ACM Transactions on Graphics* (2017) 3, 6, 14, 15, 16, 26, 34, 35, 37, 41, 46

[Zhao et al. 2018] Zhao, Jiaojiao ; Liu, Li ; Snoek, Cees G. M. ; Han, Jungong ; Shao, Ling : *Pixel-level Semantics Guided Image Colorization*. 2018 98

[Zhou et al. 2017] Zhou, Bolei ; Lapedriza, Agata ; Khosla, Aditya ; Oliva, Aude ; Torralba, Antonio : Places: A 10 million image database for scene recognition. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2017), n. 6, pp. 1452–1464 36

Appendix A

Appendix

A.1 Supplementary details of Chapter 2

As mentioned in Section 2.6, some authors colorize an image after learning a certain probability distribution such as, for instance, a color probability distribution (Larsson et al., 2016; Zhang et al., 2016; Royer et al., 2017; Yin et al., 2021), or a distribution of semantic classes (Vitoria et al., 2020), or directly using it for classification purposes (Iizuka et al., 2016). The remaining of this section describes the corresponding measures of the difference between two probability distributions that have been used in the mentioned related work (see also Table 2.6).

Kullback–Leibler loss. The *Kullback–Leibler* (KL) loss is the directed divergence between two probability densities ρ and $\widehat{\rho}$ defined in the same space $\mathcal{Y}$. It is defined as the relative entropy from $\widehat{\rho}$ to ρ which, for discrete probability densities, is given by

$$KL(\rho||\widehat{\rho}) = \sum_{y\in\mathcal{Y}} \rho(T) \log \frac{\rho(T)}{\widehat{\rho}(\hat{T})}. \tag{A.1}$$

Here, ρ is usually taken as the ground truth density (sometimes as a Dirac delta or a one-hot vector on the ground truth value, or a regularized one) and $\widehat{\rho}$ the predicted one. In the image colorization task, this loss measures the dissimilarity between the predicted color distribution $\widehat{\rho}$ and the ground truth color distribution ρ of an image.

Cross-Entropy Loss. Cross-Entropy loss is used for classification problems and it is sometimes referred to as logistic loss. For discrete densities, it is defined as

$$CE(\rho, \widehat{\rho}) = -\sum_{y\in\mathcal{Y}} \rho(y) \log \widehat{\rho}(y), \tag{A.2}$$

where, again, ρ is usually taken as the ground truth density and $\widehat{\rho}$ the predicted one. In the classification context, ρ is often a one-hot vector equal to 1 on the ground truth class, or a regularized version of it. Let us also note, from Equation (A.1) and Equation (A.2), that there is a relationship between the Kullback–Leibler and the Cross-Entropy losses given by

$$CE(\rho, \widehat{\rho}) = E(\rho) + KL(\rho||\widehat{\rho}), \tag{A.3}$$

where $E(\rho)$ denotes the entropy of ρ. Specifically, CE in the colorization task is usually applied to color distributions. This approach treats the colorization problem as multinomial classification by learning a mapping from the input grayscale image to a probability distribution over possible discrete chrominance values. In this context, CE compares the estimated distribution with the one of the ground truth.

Log-likelihood Maximization for Diversity. Some works in automatic colorization propose to generate multiple possible colorizations (Guadarrama et al., 2017; Royer et al., 2017; Kumar et al., 2021). for the same input grayscale image by sampling over possible color distributions that are often learned by maximizing the log likelihood conditioned to the grayscale image. These methods are motivated by the observation that there are multiple plausible colorizations for a given grayscale image, and that sampling from a distribution of colorizations can produce more diverse and creative results. In this approach a color probability distribution $p(T)$ can be in principle, learned by choosing an order of the data variables $T = (T_1, T_2, \ldots, T_n) \in \mathcal{X}$, associated to the color values of a discrete

color image and its n pixels (where $\mathcal{X}$ denotes the space of discrete color images), and exploiting the fact that the joint distribution can be decomposed as

$$p(T) = p(T_1, T_2, \ldots, T_n) = p(T_1) \prod_{i=2}^{n} p(T_i | T_1, \ldots, T_{i-1}). \tag{A.4}$$

This approach can be applied to, for example, a deep-learning network to predict for each pixel a probability distribution over all possible chrominances conditioned to the luminance (as in (Royer et al., 2017)). Specifically, we can leverage the Lab color space where $p(T_{ab}|T_L)$ can be seen as the product of terms of the form $p(T_{ab_i}|T_{ab_1}, \ldots, T_{ab_{i-1}}, T_L)$, that is learned on a set of training images D by minimizing negative log-likelihood of the chrominance channels in the training data

$$\arg\min - \sum_{T \in D} \log p(T_{ab}|T_L). \tag{A.5}$$

T_L and T_{ab} denote the luminance and chrominance channels, respectively.

Appendix B

Appendix

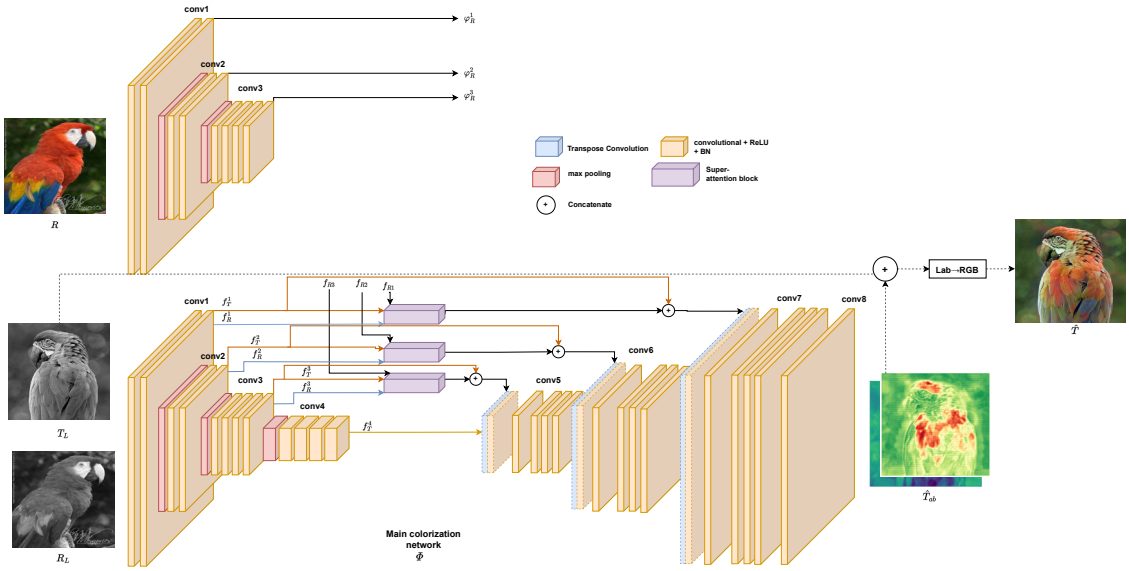


Figure B.1: Detailed architecture of our colorization pipeline.

B.1 Supplementary details of Chapter 5

In this supplementary material of our examplar-based colorization proposal presented in Chapter 5, we show more in-depth details of the architecture, also other comparison results with respect to non-learning methods as well as results on archive images. In addition, we developed another variant of our proposal as mentioned in Section 5.4 where we include the histogram loss on our final method.

B.1.1 Architecture details

Figure B.1 details the complete architecture. Our implementation is based on a U-net like encoder-decoder architecture (Ronneberger et al., 2015), the encoder part is a pre-trained VGG19 without the final dense layers, and the decoder is the mirror architecture of the encoder but without pre-train. Each level has two to three convolutional layers with ReLU as its activation function and a batch normalization layer. The architecture uses max-pooling as a downsampling operator and transposed convolution for upsampling. For each skip connection, we add the super-attention block between the encoder and the decoder for the first three levels. For detailed information about size of output resolution of our framework, see Table B.1 and Table B.2.

B.1.2 Comparisons with non-learning based methods

This section presents additional experimental results. Figure B.2 shows the comparison between our method and two state-of-the-art non-learning based methods: (Welsh et al., 2002) and (Pierre et al., 2015a). In general, (Welsh et al., 2002) and (Pierre et al., 2015a) present more unrealistic colorization results and, in certain cases, evident color bleeding

Table B.1: Detailed architecture and output resolution for each block.

Layer type	Output resolution
Input	3 x H x W
Conv1 + Max-pooling	64 x H/2 x W/2
Conv2 + Max-pooling	128 x H/4 x W/4
Conv3 + Max-pooling	256 x H/8 x W/8
Conv4 + Conv. Transpose (I)	512 x H/4 x W/4
Conv5 + Conv. Transpose (II)	256 x H/2 x W/2
Conv6 + Conv. Transpose (III)	64 x H x W
Conv8	C x H x W

Table B.2: Detailed architecture and output resolution for super-attention blocks.

Layer type	Output resolution
Super-attention 1	64 x H x W
Super-attention 2	128 x H/2 x W/2
Super-attention 3	256 x H/4 x W/4

over the images. The reason is that both methods rely on patch matching, and semantic characteristics of images are not taken into account. For instance, no good patch correspondences are found in the reference image for the giraffe from the first image or the coat from the fourth image. Conversely, our method uses semantic features, which let us retrieve content not presented in the reference image, such as the color of the egg from the last image.

B.1.3 Results on archive images

Colorizing archive images is still a challenging task for all methods because of the difference in quality between legacy black and white images and modern images. In Figure B.3 we show a comparison between state-of-the-art-methods, the previously presented two methods (Welsh et al., 2002) and (Pierre et al., 2015a), and three deep-learning exemplar-based methods (He et al., 2018), (Yin et al., 2021) and (Blanch et al., 2021) on archive black and white images.

B.1.4 Adding histogram loss

A way to help reference-based colorization is to favor the transfer of color histogram from the reference to the target image. While we do not want a complete histogram transfer, as reference and target images are usually not similar, encouraging histogram transfer may ensure the presence of most colors from the reference into the final color image.

In the deep learning literature, (Lu et al., 2020) proposed to rely on an additional histogram loss L_{hist} to train their model. This loss forces the predicted color histogram to be similar to the one of the reference image. In (Lu et al., 2020), the authors proposed a differentiable way to compute the color histogram from the output chrominance channels. This method relies on a set of discretized chrominance bins and bilinear interpolation. We experimented the same approach to compute color histograms. The histogram loss is then

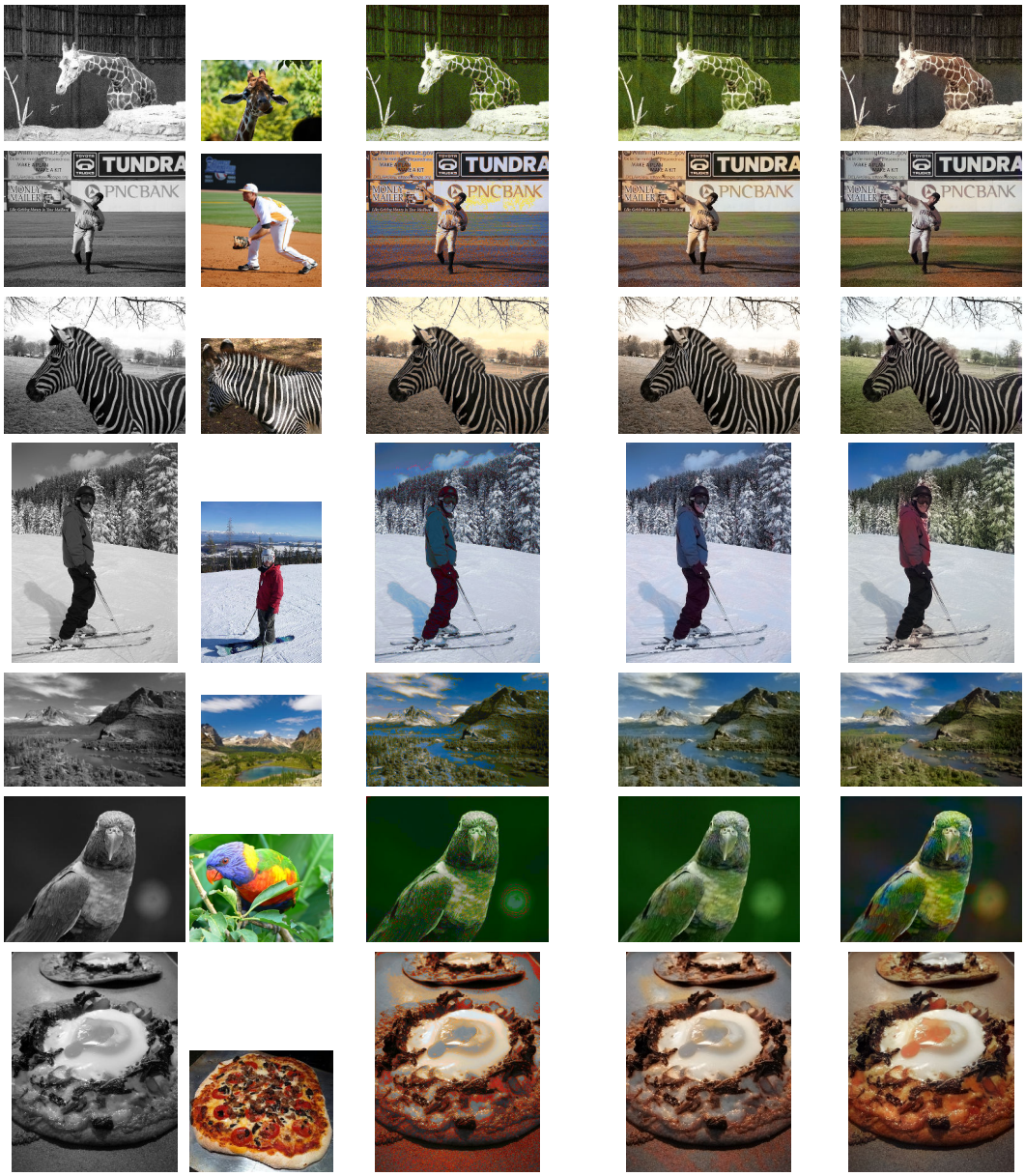


Figure B.2: Comparison of our proposed method with two non-learning reference-based Welsh *et al.* (Welsh et al., 2002) and Pierre *et al.* (Pierre et al., 2015a).

defined as the symmetric $\mathcal{X}^2$ distance, between the reference color histogram and the one from our prediction:

$$L_{hist} = 2 \sum_{q=1}^{Q} \frac{\left(\hat{T}_H(q) - R_H(q)\right)^2}{\hat{T}_H(q) + R_H(q) + \epsilon}, \tag{B.1}$$

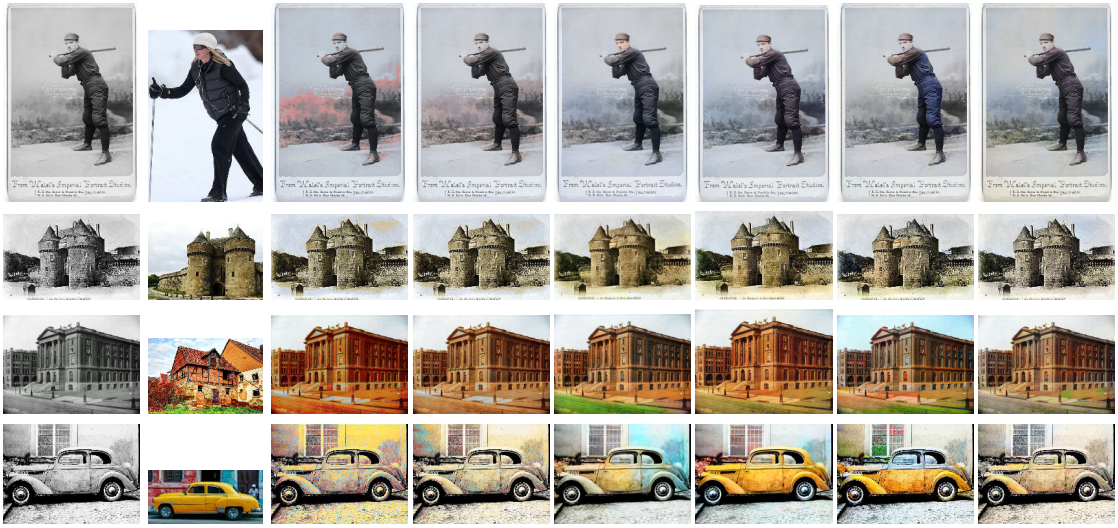


Figure B.3: Comparison of our proposed method on archive images with: two non-learning reference-based, (a) (Welsh et al., 2002) and (b) (Pierre et al., 2015a); and three deep-learning exemplar-based methods, (c) Deep Exemplar (He et al., 2018), (d) Just Attention (Yin et al., 2021) and (e) XCNET (Blanch et al., 2021).

Table B.3: Quantitative analysis of our model. The metrics SSIM and LPIPS are calculated w.r.t the target groundtruth image, and the HIS metric is calculated w.r.t the ab channel's histogram from the reference image.

	Comparison with groundtruth		Histogram transfer
Model	**SSIM ↑**	**LPIPS ↓**	**ΔHIS ↓**
Ours without reference	0.920	0.164	-
Ours with histogram loss	0.901	0.187	0.252
Ours	**0.925**	**0.160**	**0.234**

where ϵ prevents division by zero and q represents the histogram bins. In our experiments, we set $\epsilon = 1e^{-5}$ and $Q = 441$.

For quantitative evaluation, we consider the addition of histogram loss in our model. The results of this evaluation are shown in Table B.3. As expected, the addition of histogram loss to our framework leads to a much higher raw HIS score, suggesting that more colors from the reference are transferred. However, this comes at the cost of a loss of performance in the two other reconstruction metrics, namely SSIM and LPIPS, in comparison with other variants. Besides, we recall that the goal of this exemplar-based method is not necessarily force the transfer of colors of the reference, but to allow the main model to use these colors as hints to facilitate the final colorization. In this sense, our interpretation is that histogram loss induces a stronger, global transfer of colors between reference and target images, which is not desirable for our application, while without this term it encourages a more specific color transfer.

This interpretation are confirmed by the qualitative results presented in Figure B.4.

Figure B.4: Comparison of results obtained using different variants of our colorization framework.

From these results, we can observe that the histogram loss variant leads to more vivid but unrealistic colors, as well as additional color bleeding.