

CR5 PROJET HARMONISATION DES COULEURS M2

Roland BERTIN-JOHANNET, Benjamin PRE

November 27, 2022

Travail réalisé

Depuis la dernière fois, nous avons :

1. Commencé à implémenter un algorithme de down sampling et d'upsampling avec carte bi-directionnelle des gradients.
2. Implémenté la seconde couche MLE (maximum likelihood estimation) décrite dans l'article *A Color-Pair Based Approach for Accurate Color Harmony Estimation*¹ pour obtenir un score sur une palette à partir d'une méthode de scorage de paires.
3. Comparé nos deux méthodes de scorage de paires de couleurs selon cette extrapolation, en se basant sur la base de données MTurk décrite dans l'article de Yang et al. (les comparaisons sont données figure 1)
4. Comparé l'efficacité des deux méthodes sur la base de données Kuler (du même article), et montré qu'une combinaison linéaire des deux permettait une meilleure estimation du score d'harmonie que chacune individuellement, mais pas meilleure que celle de l'article de Yang et al. (que nous n'avons pas implémenté en entier).
5. Conçu et commencé à implémenter une fonction de coût dérivable (donc rétro-propageable) qui permet de pénaliser la distance de l'histogramme des teintes d'une image générée à une distribution idéale de notre choix.
6. Implémenté une fonction d'évaluation de nos modèles qui regarde si leurs images générées tombent bien dans la bonne classe quand on les classe avec la méthode K-means. (voir figure 2)
7. Entraîné certains modèles pendant plus longtemps (50 000 itérations pour le modèle complémentaire par exemple), et fait des modifications aux méta-paramètres.

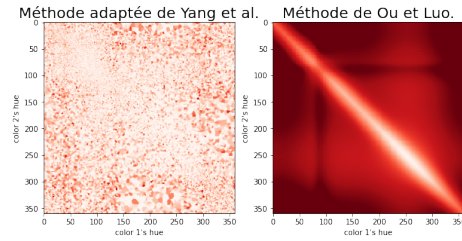
Travail à réaliser

D'ici au prochain compte-rendu, nous espérons :

1. Finir d'implémenter notre fonction rétro-propageable de distance entre histogrammes, et voir si un autoencodeur parvient à la minimiser.
2. Intégrer cette fonction à l'entraînement de notre CycleGAN, et voir si cela améliore ses résultats par rapport à la figure 2
3. Commencer à explorer l'idée de changer nos GANs en Wasserstein GANS, et d'entraîner le critique (le discriminateur) à donner un score d'harmonie (que nous savons calculer) en même temps qu'on l'entraîne en compétition avec le générateur.
4. Finir l'approche d'upsampling par cartes de gradients.

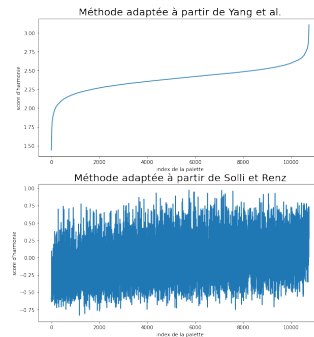
¹https://www.researchgate.net/publication/337247169_A_Color-Pair_Based_Approach_for_Accurate_Color_Harmony_Estimation

Scores d'harmonie des deux méthodes sur des paires de couleurs



(a) heatmap des scores d'harmonie sur des paires de teintes selon les deux méthodes. Nous remarquons dans les deux cas une diagonale descendante qui indique que les deux méthodes jugent plus harmoniques les teintes similaires.

Comparaison des deux scores d'harmonie sur des palettes



(b) Courbe des scores d'harmonies sur des palettes, selon les deux méthodes, avec les palettes triées dans le même ordre. Note : le coefficient de Pearson est 0.3334.

Figure 1: Quelques résultats

5. Explorer les possibilités pour faire tourner notre upsampling (en c++) depuis python (par exemple pyBind, etc.)
6. Si nous arrivons vite à faire cela, entraîner un autoencodeur à corriger, par vignettes, les imprécisions de notre méthode d'upsampling (voir figure 3)
7. Commencer à réfléchir à ce que l'on pourrait mettre dans un sondage en ligne (tester des hypothèses sur les scores d'harmonie, etc.)

Classification des images générées par le modèle rect

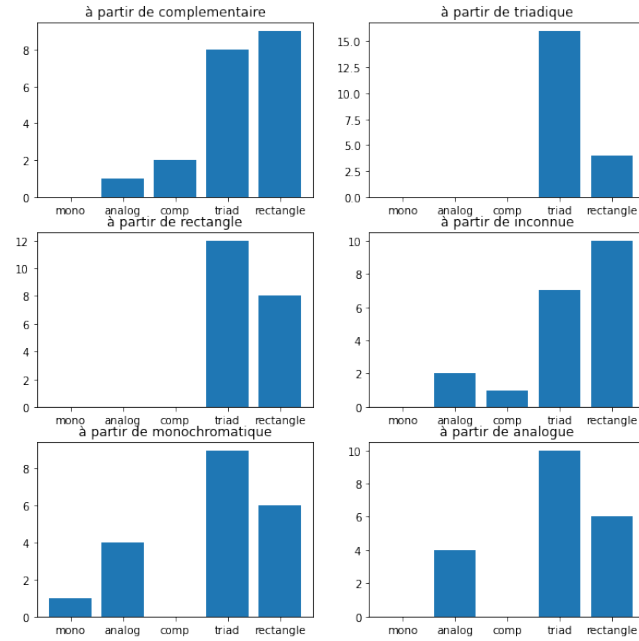


Figure 2: Résultat de notre évaluation du modèle générant des images d'harmonie rectangulaire : nous l'appliquons à des images venant de chaque type et affichons où la méthode kmeans mettrait le résultat. Noter que même sans appliquer le modèle, nous n'obtenons pas des histogrammes parfaits car nous avons rééquilibré les classes lors de la constitution de la base de données en appliquant le kmeans avec d'autres paramètres.

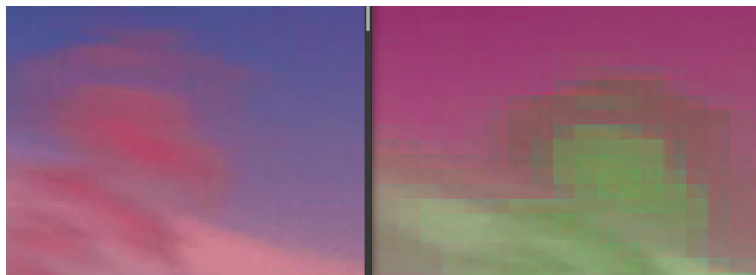


Figure 3: Défauts dans l'upsampling par carte des différences