Rapport
David Alexander (Voronoi diagram)

Description de l'algorithme :

Comme décidé lors de notre discussion, mon algorithme utilisé une partition de Voronoi.
On prend une image en entrée. On a deux possibilités : soit l'on procède a une extraction des 'facial landmarks' de l'image qu'on utilise pour construire un diagramme de voronoi, soit l'on selectionne des positions de manière aléatoire pour construire un diagramme de voronoi.
On utilise la libraire Scipy pour construire le diagramme de Voronoi. La fonction ne permet de pas de créer des diagrammes finis, on la complète donc pour rendre cela possible.
Ensuite pour chaque région du diagramme de Voronoi, on utilise l'algorithme dit de K-means clustering pour extraire la couleur dominante. Remarquons que pour obtenir de meilleurs résultats on utilise cet algorithme sur des couleurs converties au format Lab (qui correspond mieux à la perception humaine des couleurs que RGB).
Pour l'extraction des facial landmarks, on utilise la libraire Dlib, d'autres librairies (Tensorflow,Keras, Pytorch,etc) permettent également de le faire mais il se trouve que j'ai trouvé un article très complet expliquant comment utiliser Dlib pour cette tâche ainsi qu'un modéle 'pretrained' et j'ai donc choisi cette librairie, ne disposant pas de GPU suffisament puissant pour train mon propre modèle.
Le résultat ne donne pas une segmentation rectangulaire/carrée comme chez mondrian. Aprés avoir envisagé différentes normes(Lmax,Manhattan,etc) je suis parvenu à la conclusion qu'aucune d'entre elles ne donne une segmentation 'rectangulaire'. Voir la vidéo suivante par exemple :
https://www.youtube.com/watch?v=sFLuFlhLJFM (voronoi diagrams with Lp norms from 1 to infinity).
J'ai donc choisi d'utiliser la version 'standard' de l'algorithme avec la norme L2.