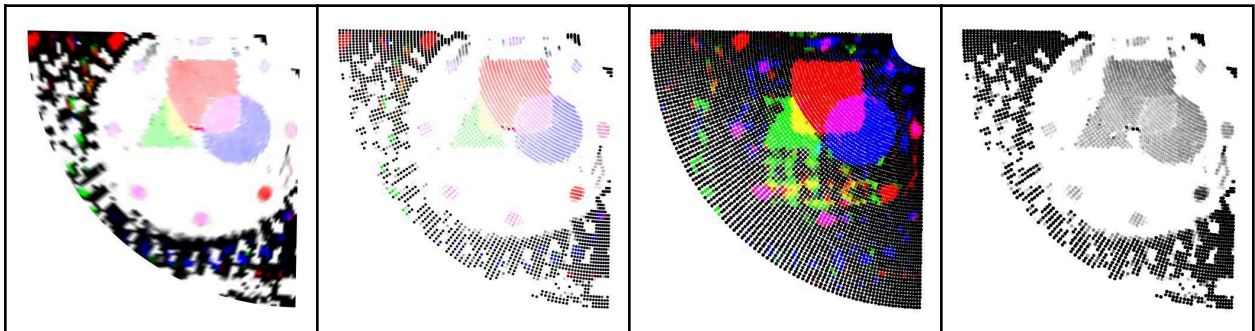
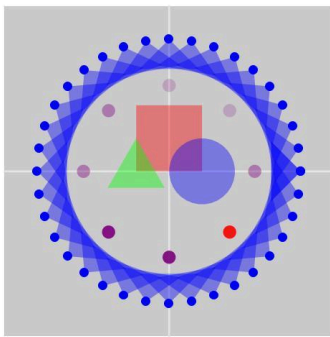


### Question 1

*Un monde avec plus d'objets avec une bonne variation dans les densités. (utilisez des densités  $\leq 1$ )*

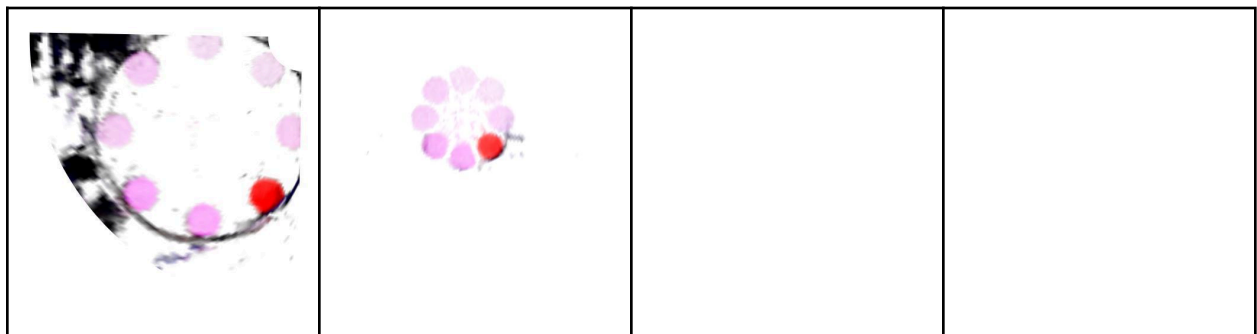
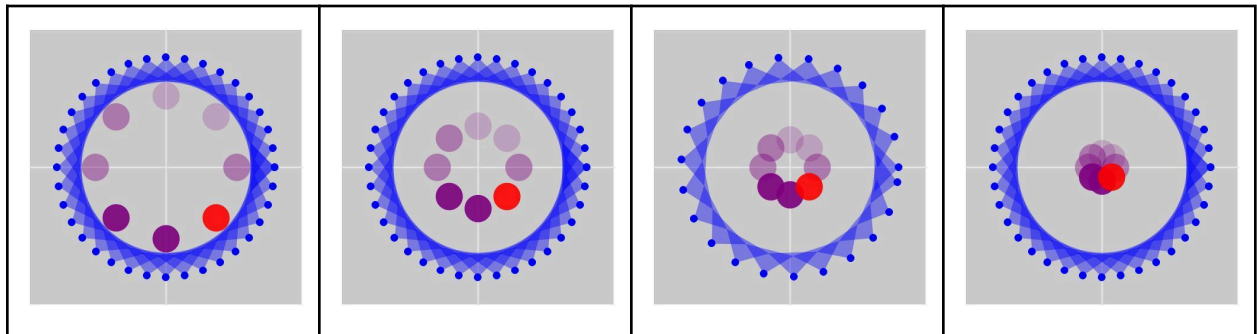
Voici un monde un peu plus complexe avec huit nouveaux cercles de densités différentes. Le modèle a pu apprendre sans difficulté cette nouvelle scène.



## Question 2

*Un monde qui est beaucoup plus difficile à reconstruire (votre à goût) Dans ce cas, on veut que le réseau apprennent bien les images (donc la loss descend très bas) mais ne donne pas une bonne reconstruction.*

En essayant de trouver un cas difficile à reconstruire, je suis tombé sur ce cas simple qui ne fonctionne pas. Le modèle apprend et généralise dans les deux premières images, mais échoue dans la troisième et quatrième images quand les cercles sont plus rapprochés, même si la loss est extrêmement proche de zéro.

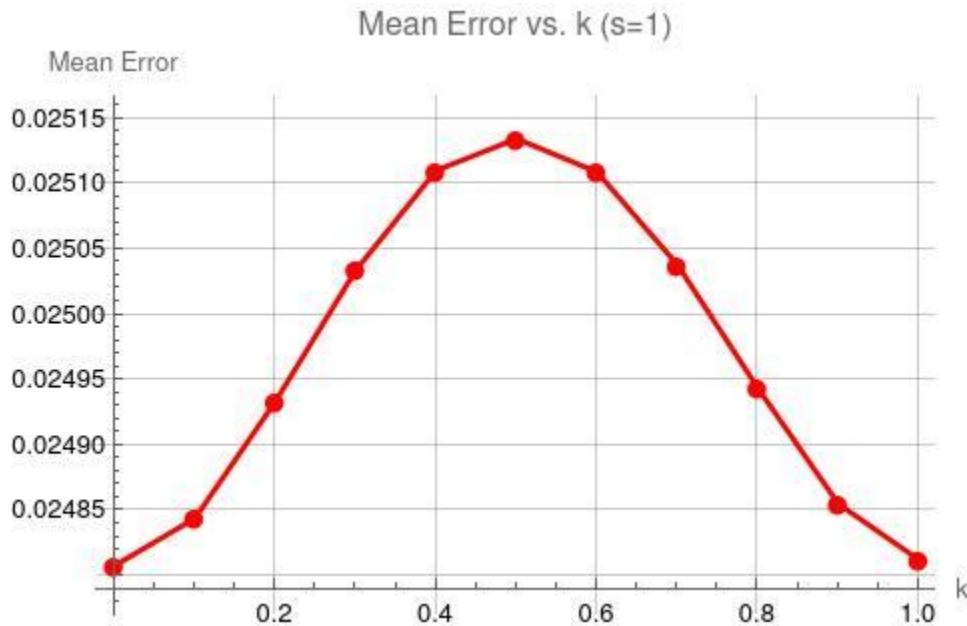


### Question 3

Faites un entraînement pour un  $s=1$ , mais testez avec plusieurs (une dizaine)  $k$ , entre  $k = 0$  et  $k = s$

Affichez l'erreur moyenne obtenue pour différents  $k$ , et expliquez le résultat

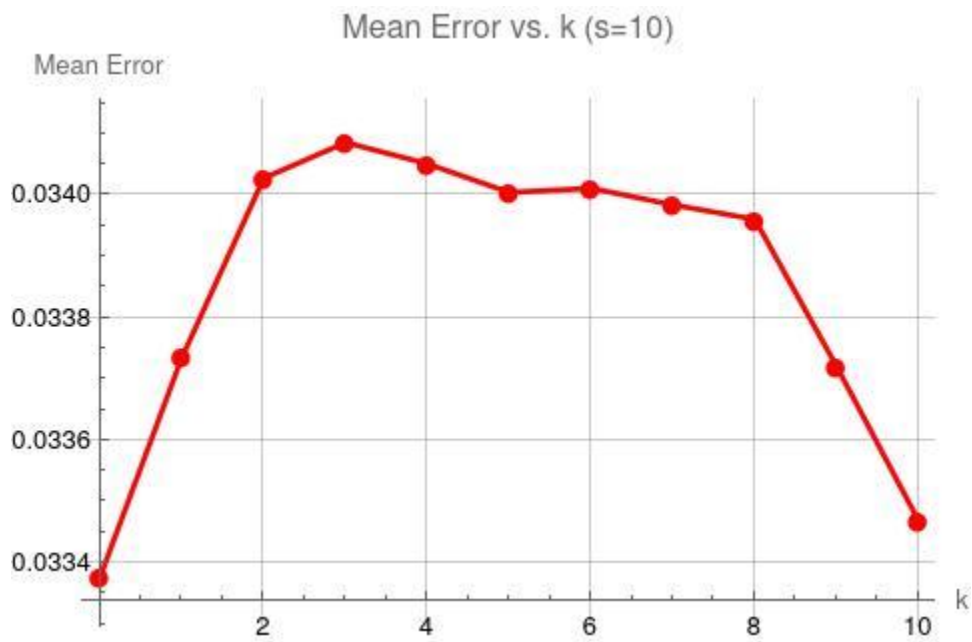
Le graphique est extrêmement proche de ce qu'on pourrait s'imaginer. À 0 et à 1, là où le modèle a été entraîné le modèle performe super bien. La performance diminue en s'éloignant des points d'entraînement.



#### Question 4

Refaites exactement la même chose avec un  $S$  plus grand (par exemple 10). Expliquez le résultat.

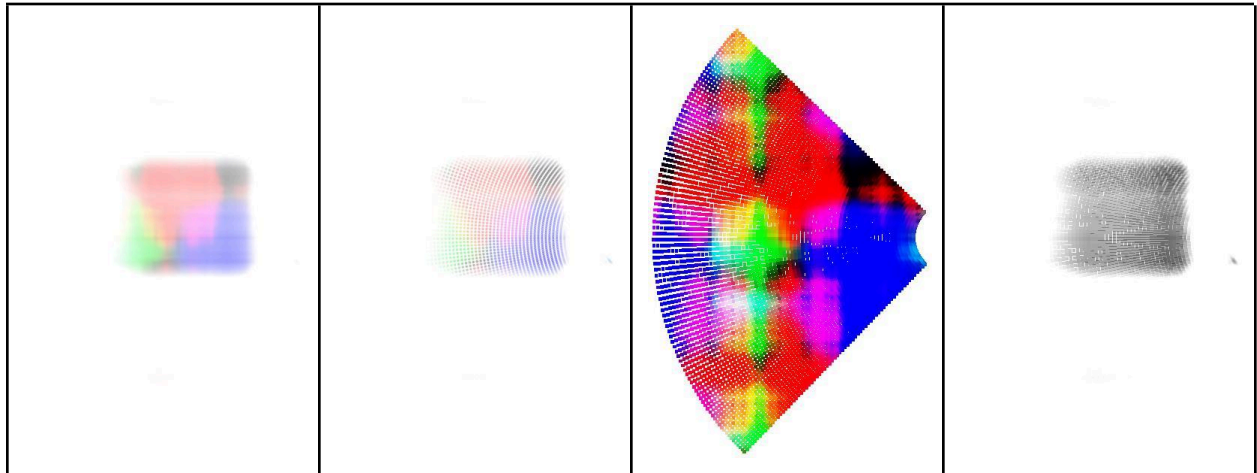
De façon très similaire, le modèle généralise mieux aux positions connues et performe moins bien plus il s'éloigne de celles-ci.



### Question 5

*Changez l'architecture du réseau (les convolutions 128) pour autre chose et comparez les résultats.*

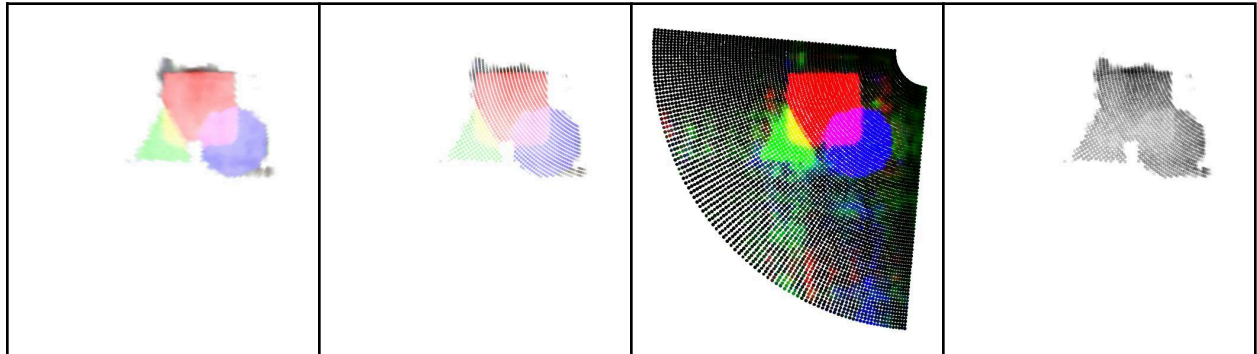
J'ai décidé de changer toutes les convolutions 128 par des couches de Batch Normalization. On voit clairement l'effet que cela a eu. Le modèle confond les positions et est uniquement capable de générer une sorte de moyenne des représentations.



### Question 6

Changez le "positional encoding" pour autre chose (une version plus simple, une plus complexe) et comparez les résultats.

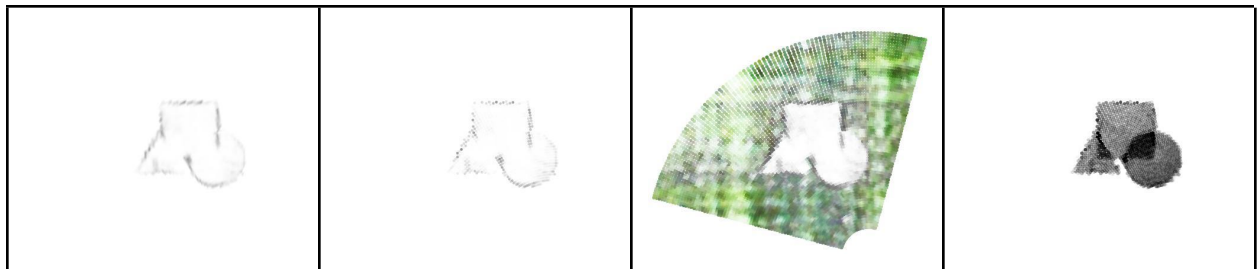
J'ai modifié l'encodeur en ajoutant une couche de sortie exponentielle. Ceci n'a pas eu d'effet notable sur le modèle.



### Question 7

Convertir un monde RGB en ton de gris ("ColorConvert[]") et comparer le résultat de nerf sur les deux mondes.

L'apprentissage s'est bien déroulé, il aurait été très étonnant que le modèle apprenne moins bien qu'en couleur, puisque la couleur n'ajoute pas nécessairement d'informations pertinentes pour le modèle.



### Question 8

*À votre avis, quel est l'impact de cette omission? (Autrement dit, à quoi sert la direction? Dans quel cas est-ce que la direction est vraiment utile pour NERF?)*

L'angle de la caméra comparé à la position est une donnée normalisée et périodique, appliquer un encodage sur l'angle n'aura donc aucun impact sur l'apprentissage du modèle. Les coordonnées spatiales peuvent varier sur de grandes distances de manière linéaire, il est donc nécessaire de les transformer en information plus complexe ayant une fréquence plus proche de la fréquence de sortie désirée.