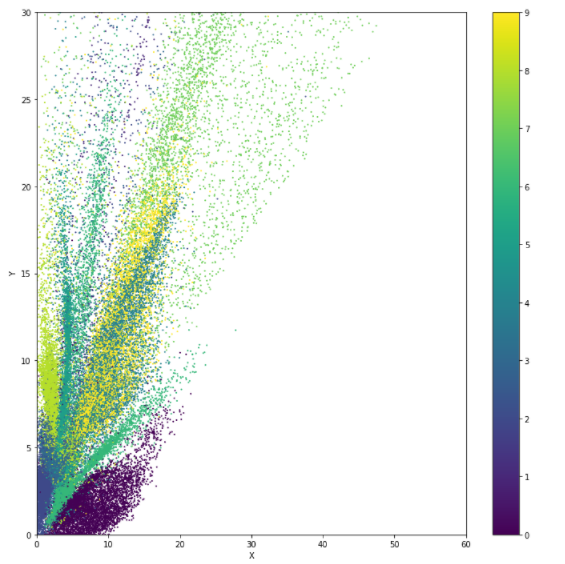
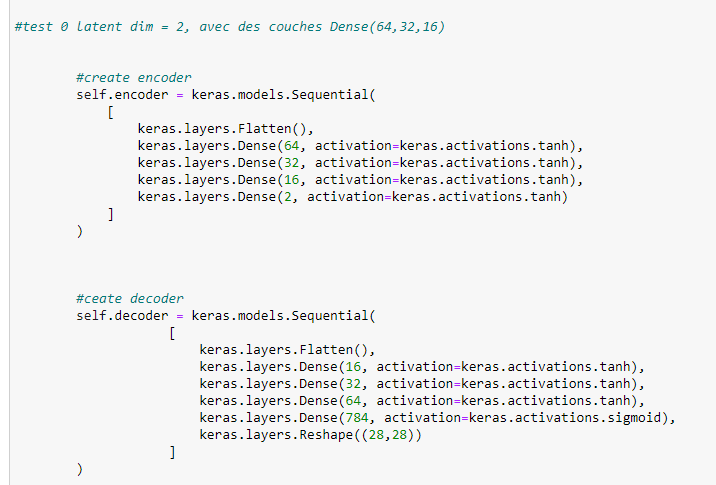
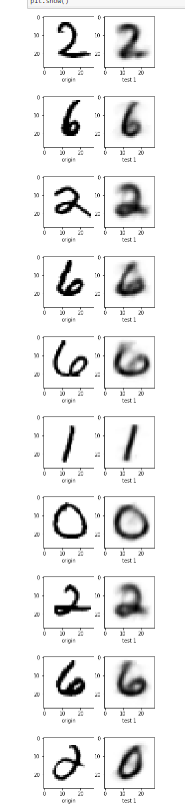
**MNIST DATASET**

Plot dataset encode (latent\_dim = 2)

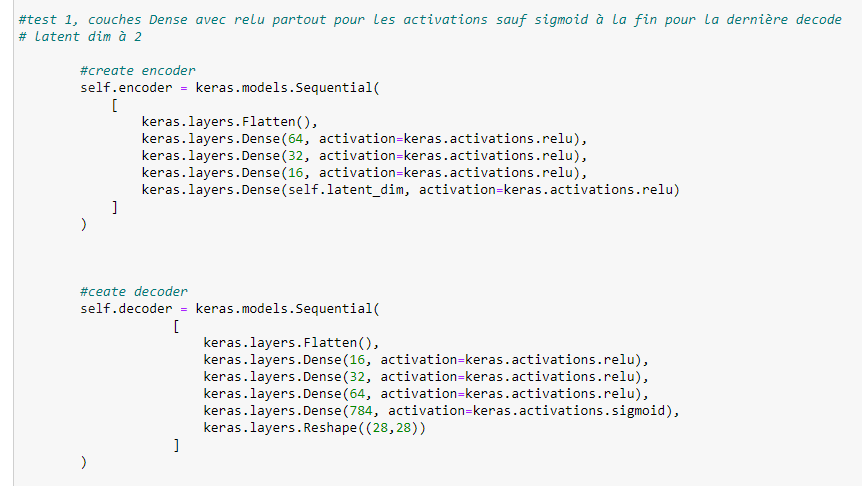


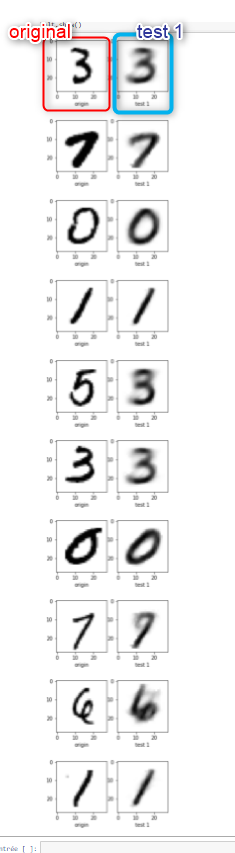
Test 0 (original à gauche, test à droite)





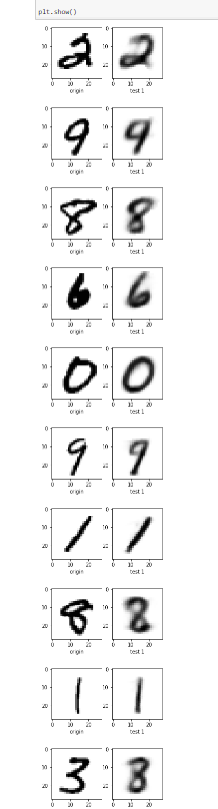
Test 1



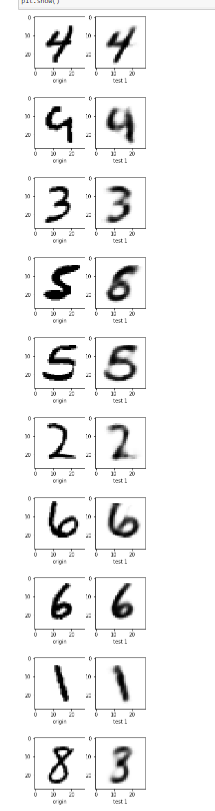


Impression 🡺 Un peu moins de flou sur les images, relu légèrement meilleur que tanh

Test 2 avec 4 en latent\_dim 🡺 quasi identique avec un peu moins de flou, a voir avec 8



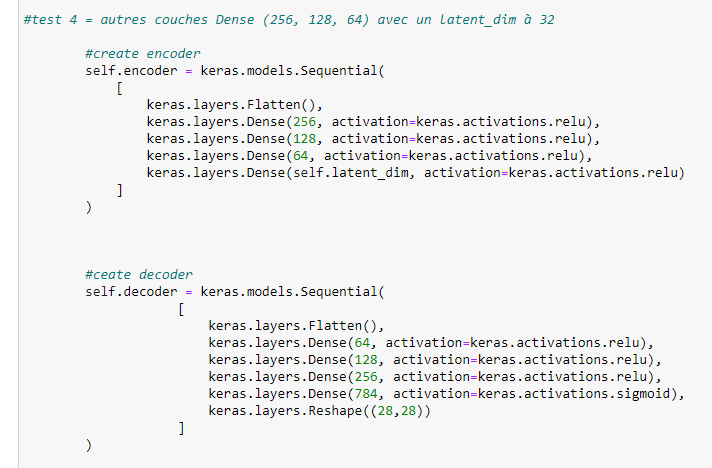
Test 3 avec 8 en latent\_dim

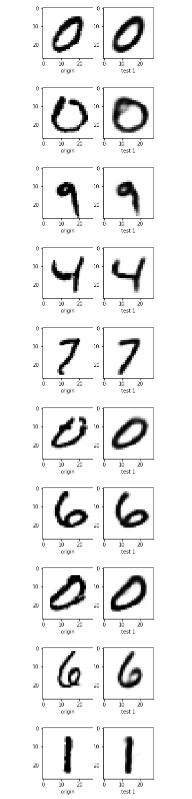


Meilleures images, plus le latent\_dim est grand et plus la qualité de l’image après compression / décompression est bonne

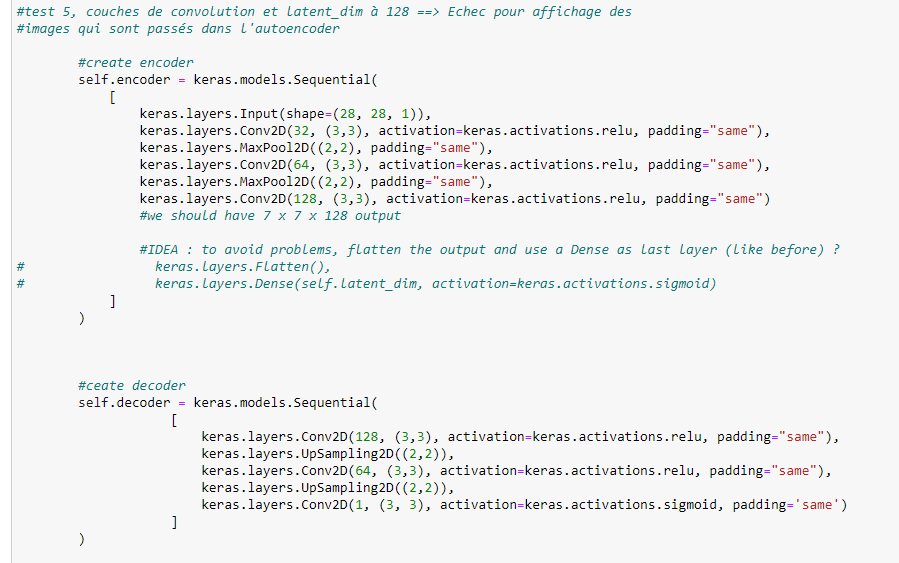
MAIS on a forcément une perte d’efficacité de la compression, la donnée est plus volumineuse.

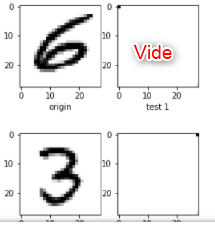
Test 4 : couches 256,128, 64 avec latent\_dim à 32 🡺 excellent résultat (loss qui passe 0.04 à 0.01 et images nettes)

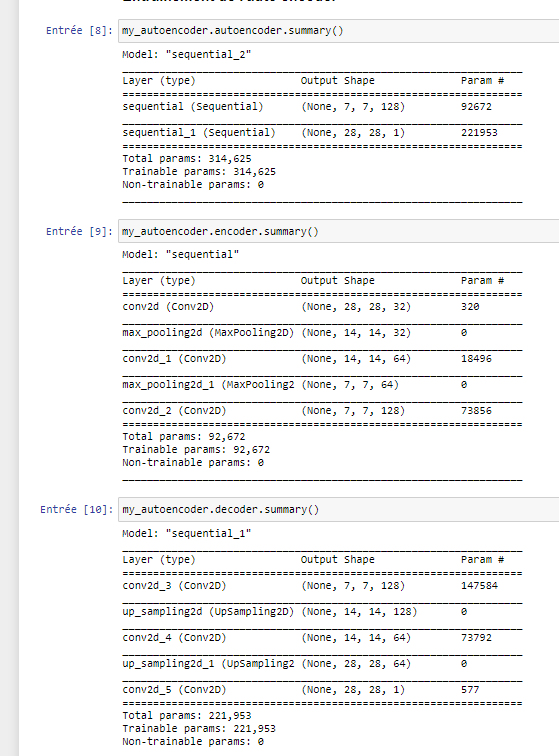




Test 5 : Convnets







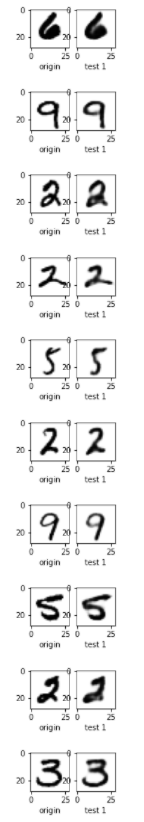
PROBLEME VENAIT DU PADDING == SAME + DES COUCHES DE CONVOLUTION

Test fonctionnel avec des couches de conv 8, 8, 16 (latent dim = 8) en utilisant Adam et Binary\_crossentropy



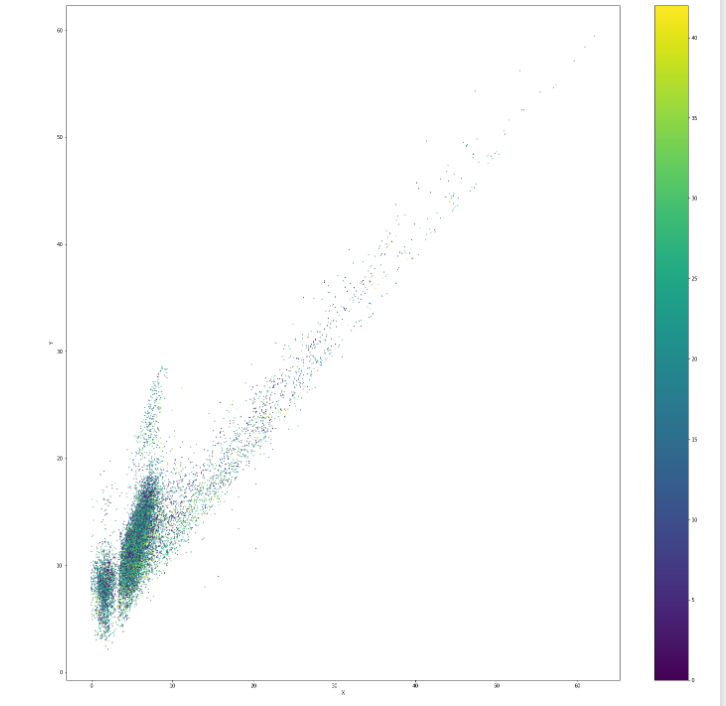


Résultat == très bonnes prédictions niveau qualité

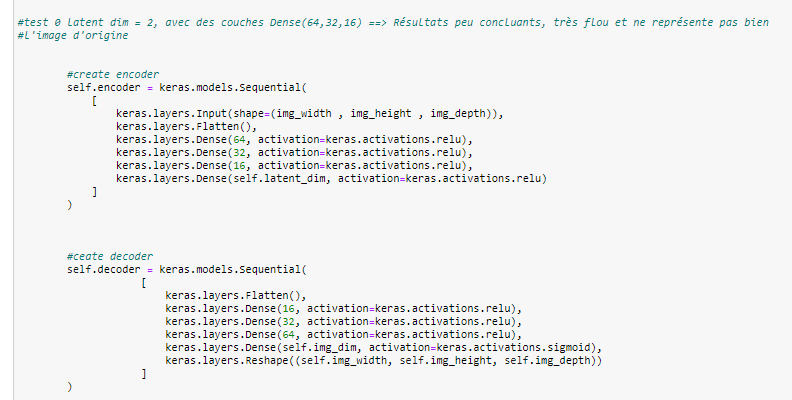


**SIMPSON DATASET**

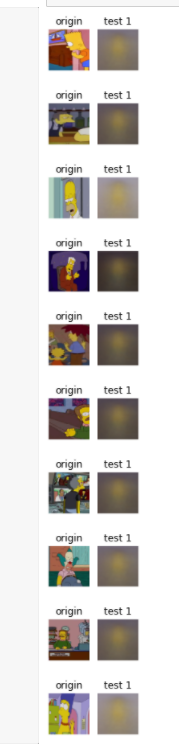
Répartition dataset (pas très lisible)



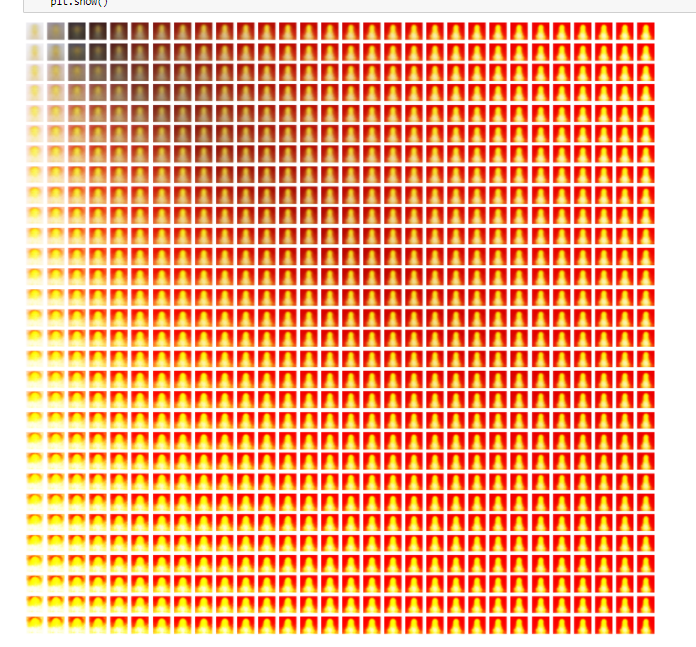
Test 1 (latent dim et couches Dense classique), entrainement avec des images 64x64 et 40 epochs



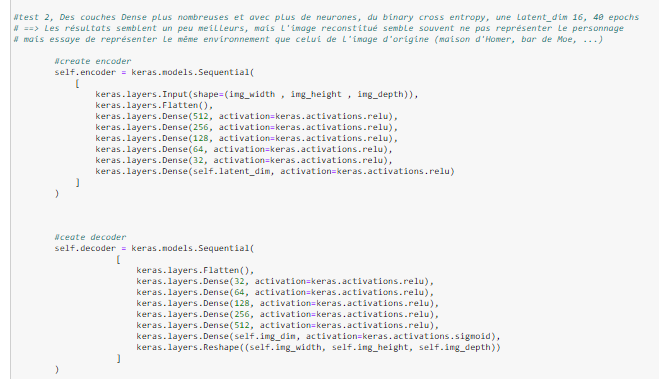
Encode decode pas très concluant, les images sont de très mauvaise qualité et ne représentent pas forcément le personnage



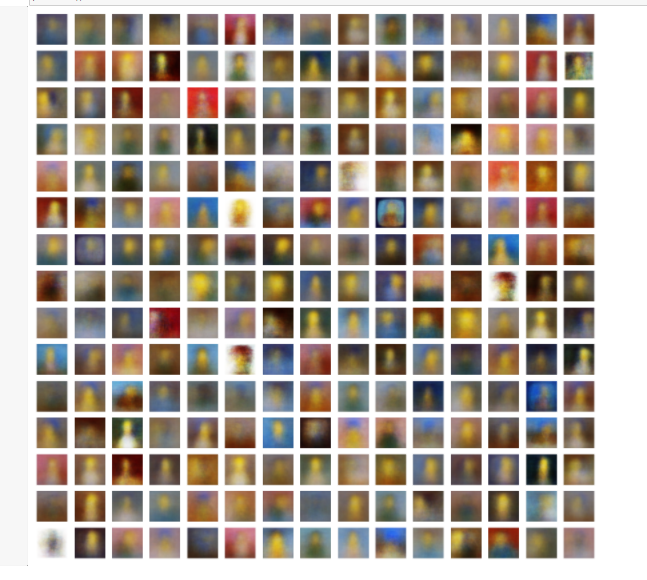
Nouvelles images générées avec une latent dim\_2



Test 2 (Des couches Dense plus nombreuses et avec plus de neurones, du binary cross entropy, une latent\_dim 16, 40 epochs)



Exemples d’images qui sont passées par l’auto encoder (pas trop d’effet de gradient vu qu’il n’est plus possible d’utiliser l’axe X et Y pour affiche 15 exemples ((x1, y1), (x1, y2), (x1, y3), …))

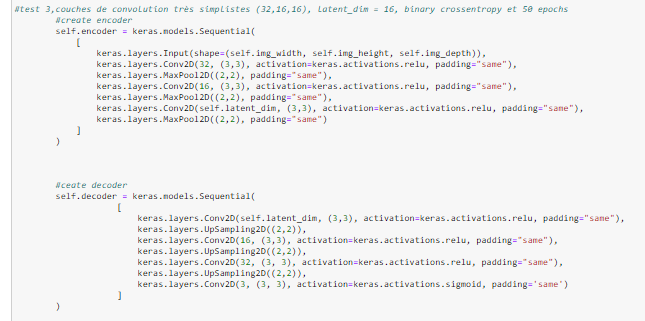


Les résultats du Encode decode semblent un peu meilleurs, mais l'image reconstituée semble parfois ne pas représenter le personnage.

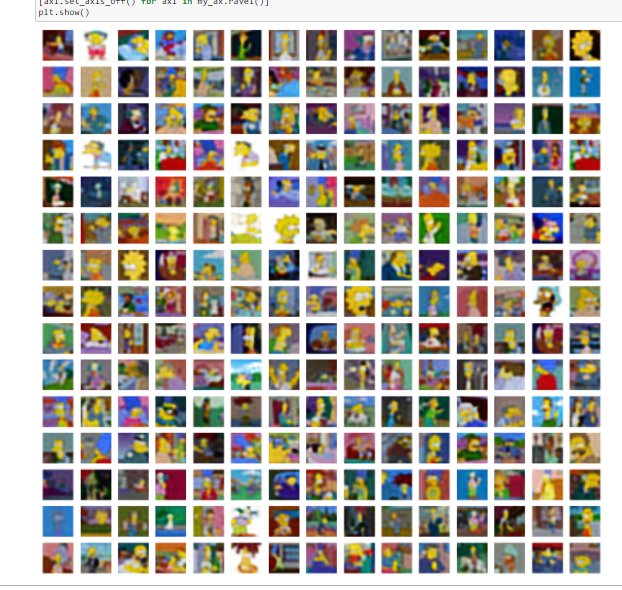
Elle essaye plutôt de représenter le même environnement que celui de l'image d'origine (maison d'Homer, bar de Moe, ...)



Test 3 ,couches de convolution très simplistes (32,16,16), latent\_dim à 16, binary crossentropy et 50 epochs



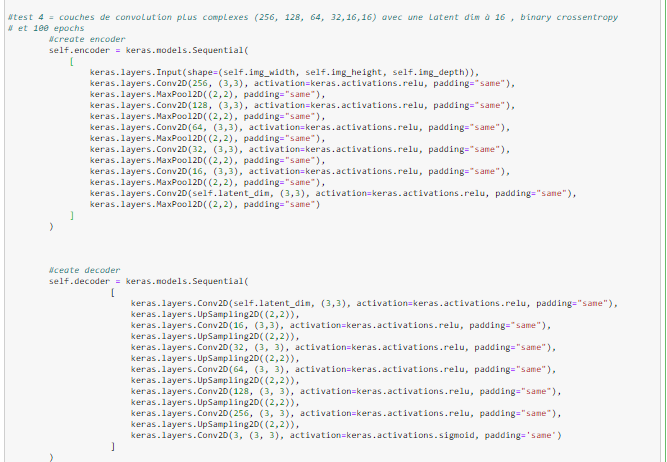
Les images passées dans le autoencoder semblent de meilleure qualité (par rapport aux modèles utilisant des couches Dense) :



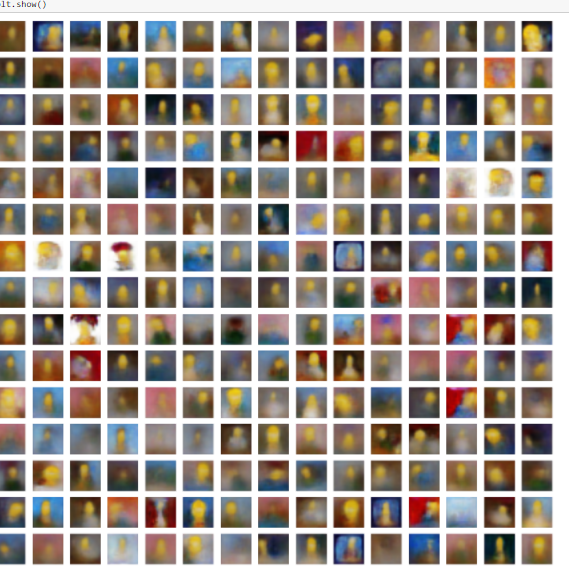
Les résultats du encode decode sont prometteurs ! On reconnait bien l’image originale avec une qualité détériorée (mais probablement due en majorité à la faible taille de l’espace latent : 16) :



Test 4 : couches de convolution plus complexes (256, 128, 64, 32,16,16) avec une latent dim à 16 , binary crossentropy et 100 epochs



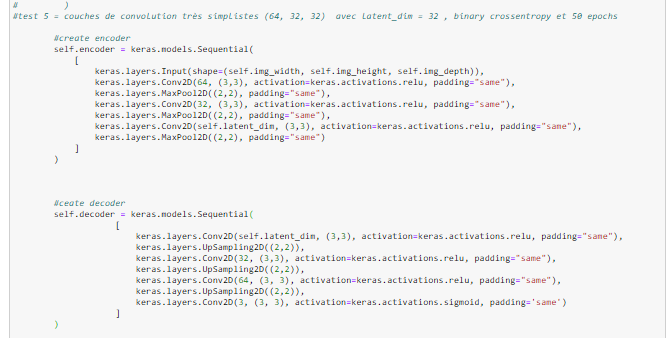
Les images passées par l’autoencoder sont très floues et de mauvaise qualité, faire trop de convolutions semble mauvais



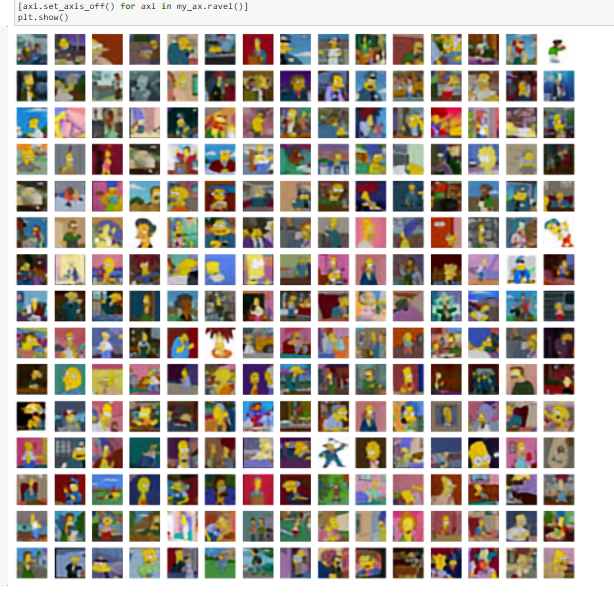
Dans le encode decode, on voit également que les résultats sont moins satisfaisant à cause du flou



Test 5 : couches de convolution très simplistes (64, 32, 32) avec latent\_dim = 32 , binary crossentropy et 50 epochs



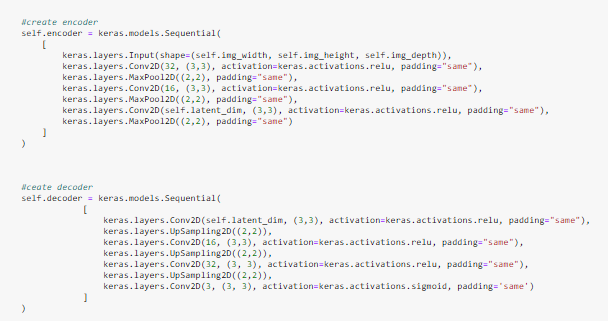
Les images passées par l’autoencoder sont un peu moins floues par rapport au modèle utilisant une latent\_dim de 16, elles sont d’assez bonne qualité



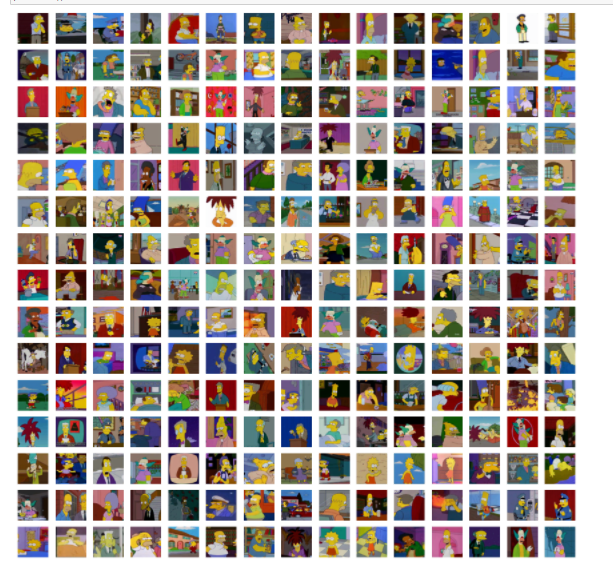
Les résultats du encode / decode sont très bons, on reconnait bien l’image originale avec une meilleure qualité qu’avec l’espace latent 16



Test 6 : Pareil que le test 3 mais avec des images 256,256,3



Les images qui sont passées dans l’autoencoder sont de très bonne qualité !



Les résultats du encode / decode sont très bons ! , on reconnait bien l’image originale malgré la taille de l’espace latent (16)

