REINDERS Erwan CLEMENT Ange

Groupe 9.1

Rapport N°4

Semaine : 14/11/2022

Ce compte rendu est le quatrième dans le cadre du projet image du premier semestre de Master 2 IMAGINE. Il comprend nos avancées dans la recherche et l'implémentation de l'environnement de recherche.

Taches effectuées

Discussion Puech :
Nous avons pu, durant la séance du lundi 14/11, nous entretenir avec monsieur Puech afin de nous éclairer sur l'algorithme de Shamir. Il nous a dans un premier temps expliqué son fonctionnement et ses variantes possibles, puis nous a présenté des exemples concrets de son utilisation, et de la relation qui pourrait exister dans de vrais cas d'utilisation, entre Shamir et des images bruitées.

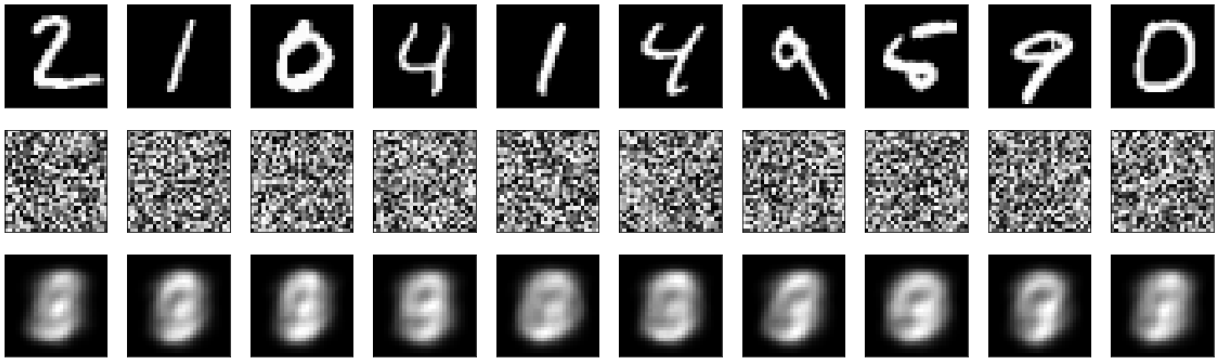
Dans le code :
Nous avons donc pu nous attaquer à la réalisation de l'algorithme de Shamir. Nous avons dans un premier temps mis en place sa version standard, en prenant des valeurs de canaux de pixels modulo 251 (plus proche nombre premier de 255).

Ensuite, nous nous sommes attaqué à la réalisation de sa version avec les corps gallois, en passant par la librairie de chiffrement que nous avons trouvé, PyCryptodome.
Cette librairie passe par GF(2^{128}), soit des blocs de 16 octets.

Résultat des tests CNN :

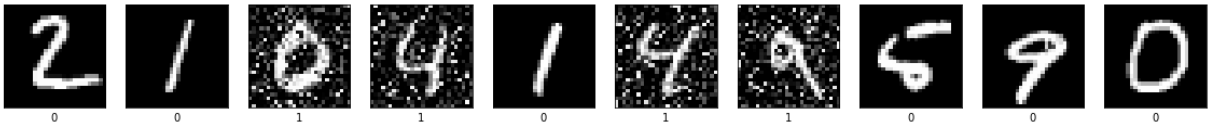
Afin de mieux comprendre les capacités d'un réseau de neurones, nous avons essayé plusieurs problèmes complexes :

- Premièrement, même si on sait que ce n'est pas possible, on voulait essayer d'apprendre au réseau de neurones à déchiffrer une image sans la clé. Voici les résultats : (Première ligne : les images en clair, deuxième ligne : les images chiffré et troisième ligne : le déchiffrage donné par le réseau de neurones)

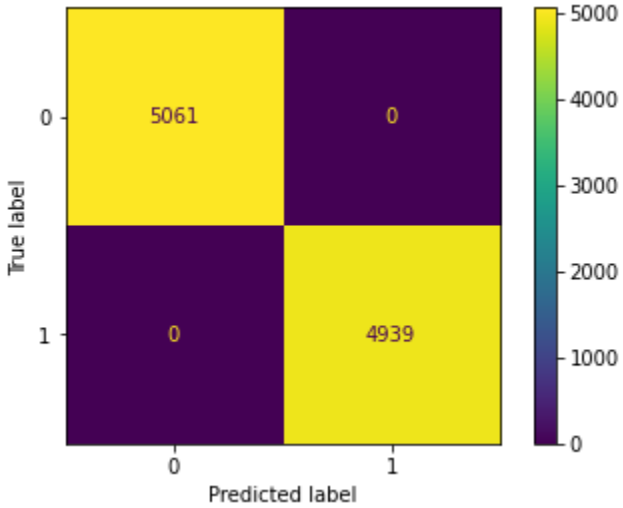


On remarque que le réseau ne peut pas déchiffrer les images et va simplement essayer d'être le plus correct possible. Il va donc faire un mélange statistique.

- Ensuite, on a jugé utile d'entraîner un réseau pour détecter si une image à été bruitée ou non. Voici les images utilisées : (0 si non bruitée et 1 si bruitée)

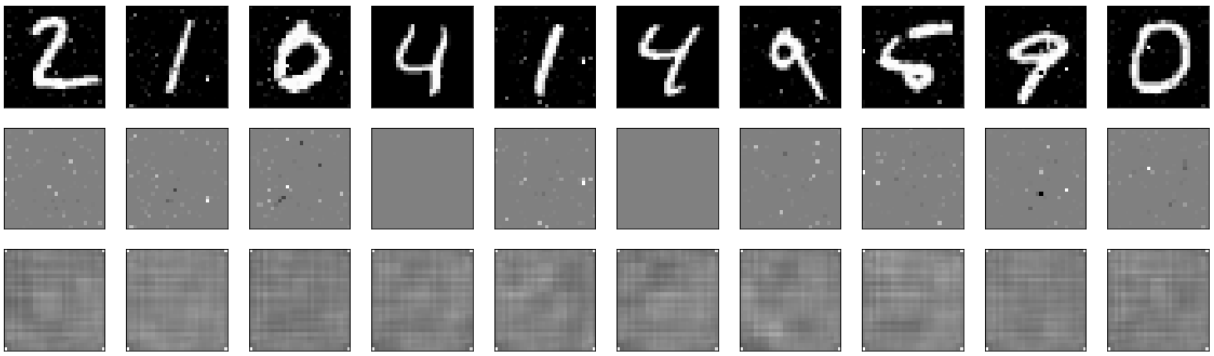


Et voici la matrice de confusion :



On remarque que, pour ces images, l'exercice est trop simple. En effet, pour ce jeu de données, il suffirait de faire la moyenne des valeurs de l'image pour différencier les images bruitées.

- Nous avons également essayé de complexifier la tâche de détection de bruit. On veut entraîner un réseau afin qu'il puisse détecter la position des pixels bruités. Malheureusement, cela n'a pas abouti. Voici les résultats : (Première ligne : les images bruitées, deuxième ligne : le bruit avec des valeurs positives et négatives, troisième ligne : le résultat du réseau)



On remarque que le réseau ne peut pas isoler les valeurs de bruit. De la même manière que pour le déchiffrement, il va faire une estimation statistique sur la position du bruit.

Reste à faire

Au niveau de la recherche :

- Rechercher davantage de techniques de chiffrement et les implémenter.

Dans le code :

- Pipeline complet pour faire un premier traitement sur des images données.
- Mise en place d'une interface pour pouvoir faire tester nos images à des utilisateurs extérieurs.