REINDERS Erwan CLEMENT Ange

Groupe 9.1

Rapport N°2

Semaine : 31/10/2022

Ce compte rendu est le second dans le cadre du projet image du premier semestre de Master 2 IMAGINE. Il comprend les détails de nos premiers travaux sur ce projet, mais également du plan de travail que l'on va chercher à suivre pour structurer notre avancée.

Taches effectuées

Plus de recherche :

- Nous avons approfondi nos recherches pour les besoins de ce projet, en nous documentant plus en détail sur les différents type de réseaux de neurones utilisables pour un tel projet.

  Nous nous sommes ainsi documentés sur les réseaux de neurones auto encodeurs. Pour cela, on a suivi le tutoriel de keras, qui présente les différents types d'auto encodeurs : profonds, avec CNN, pour le débruitage, et variationnels (https://blog.keras.io/building-autoencoders-in-keras.html).
  et nous avons regardé certaines vidéos de formation CNRS (https://www.youtube.com/c/CNRSFormationFIDLE/videos).

  Même si les CNN sont une partie importante de ce projet, nous devons les intégrer intelligemment à la problématique de départ de débruitage d'images déchiffrées. Dépendant du chiffrement employé, il n'est pas forcément nécessaire d'utiliser de telles solutions.

Discussion avec William Puech :

- Lors de la séance du lundi 07/11, nous avons pu échanger avec notre encadrant de projet monsieur Puech. Nous avons pu lui exposer notre travail à ce jour, et lui indiquer nos objectifs futurs.

  Il nous a indiqués que le sujet de projet est un sujet qui peut être vaste, car il existe beaucoup d'algorithmes de chiffrement différents, et qu'ils ne se comportent pas tous de la même manière vis-à-vis du bruit au moment de déchiffrer.

  On peut par exemple prendre le cas d'un simple bruit additif, et d'un chiffrement par XOR. Au moment du déchiffrage, on obtient alors l'image originale bruitée par la même addition. Il peut alors être simple de revenir à l'image initiale. Dépendant de la force d'addition, on peut reconnaître assez facilement le sujet de l'image bruitée après déchiffrage.

  Un autre cas exposé est celui présenté dans ce papier : https://hal.archives-ouvertes.fr/hal-03161507/file/Journal_Noisy_Encrypted_Image_Correction.pdf [de Pauline Puteaux et William Puech].
  Les auteurs vont utiliser des CNN classifieurs sur un type particulier de chiffrement : le chiffrement AES par blocs de 4x4. Dans ce cas, une petite variation dans le bloc chiffré peut engendrer une grosse variation dans le bloc déchiffré. Ce n'est pas

toujours le cas de tous les algorithmes de chiffrement, comme le chiffrement par clés secrètes de Shamir.

Au moment de déchiffrer avec la méthode présentée plus haut, on peut savoir si on a bien décodé l'image ou s'il y a eu des erreurs. Dans ce cas, on peut fixer certaines conditions sur le bruit appliqué, comme le fait d'ajouter un bruit sur un seul bit d'un bloc tous les X blocs. Comme on sait alors qu'un bloc bruité possède une erreur sur un bit, on peut changer des bits de ce bloc chiffré, et tester si on obtient un bon résultat. On peut alors se retrouver à tester toutes les combinaisons de valeurs de bits, d'où l'intérêt de prendre des blocs de taille raisonnable.

Monsieur Puech nous a donc conseillé de se renseigner davantage sur les algorithmes de chiffrement, afin d'en sélectionner certains, et de tester leur comportement face au bruit afin de générer des solutions de débruitage adaptées (faire au k par k sur les propriétés du chiffrement). Il n'est pas impossible de se retrouver avec des algorithmes de chiffrement pour lesquels il sera très compliqué de débruiter.

Ensuite, nous avons abordé le cas des métriques d'évaluation des images débruitées. Outre le fait que l'on peut mesurer la qualité des images par des mesures statistiques (PSNR, entropie par blocs), on peut aussi mettre en place des mesures faisant intervenir des personnes extérieures, évaluant la qualité de notre débruitage.
Monsieur Puech nous a alors conseillé plusieurs scénarios possibles :

- Un premier scénario dans lequel on donnerait plusieurs images à un utilisateur, qu'il pourrait alors décrire selon un certain nombre de métriques (1 : fortement bruitée, 2 : bruitée, 3 : moyennement bruitée, 4 : faiblement bruitée, 5 : claire).
- Un deuxième scénario où on présente l'image originale et l'image débruitée dans la même page, et où on demanderait de noter à l'utilisateur, de la même manière. Il pourrait alors directement comparer les résultats de débruitage.
- Un troisième scénario où l'utilisateur aurait deux images débruitées, et où il devrait indiquer laquelle des deux il trouve la moins bruitée.

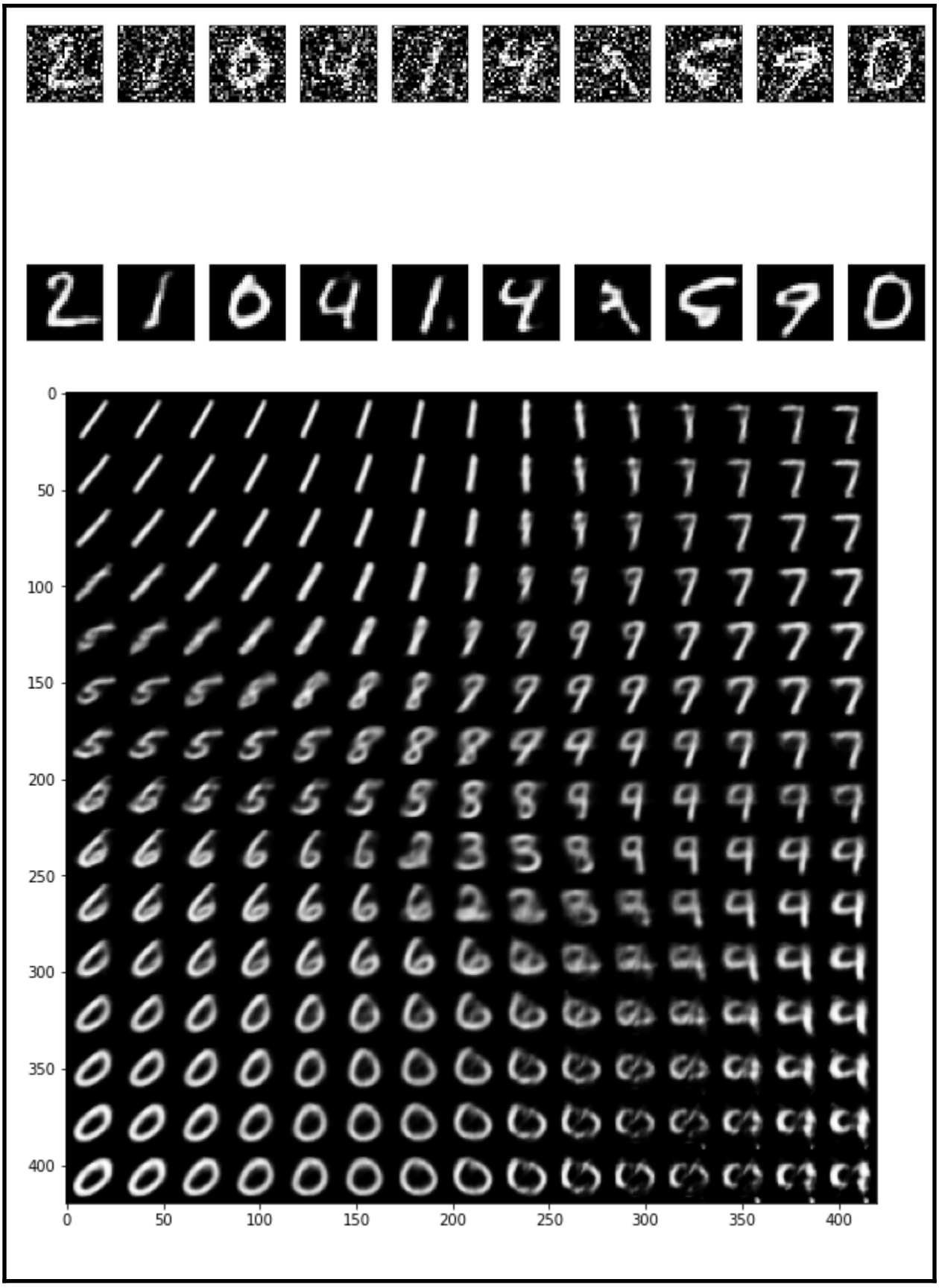
Début de mise en place du code :

- Nous avons donc commencé à produire du code python via des fichiers Google colabs. Nous avons donc suivi des tutoriels keras pour la mise en place de réseaux de neurones dit auto encodeurs. Nous avons également regardé pour la mise en place de réseaux de neurones auto encodeurs variationnels (VAE), toujours via des tutoriels keras.

Ci-dessous des exemples de résultats sur la base de données MNIST :

REINDERS Erwan CLEMENT Ange

Groupe 9.1



Nous avons, parallèlement à ça, commencé à mettre en place des opérateurs de traitement d'images en python.

Mise en place d'un plan global de tâches à effectuer prochainement :

- Dans un premier temps, on va ainsi se focaliser sur des techniques de chiffrement particulières, et sur leur comportement face au bruit (robustesse).
- Dans un second temps, on va sélectionner un certain nombre de techniques de chiffrement, sur lesquelles on va tenter de débruiter l'image déchiffrée.
- Dans un autre temps, on va mettre en place une interface afin de récolter des évaluations qualitatives d'utilisateurs basées sur leur vision.

Reste à faire

- Au niveau de la recherche :

Nous avons par exemple été conseillés par monsieur Puech afin d'étudier les ressources disponibles dans l'état de l'art du papier suivant : https://hal.archives-ouvertes.fr/hal-03161507/file/Journal_Noisy_Encrypted_Image_Correction.pdf [de Pauline Puteaux et William Puech].

Ce papier présente une technique de débruitage d'images déchiffrée bruitée par chiffrement AES en bloc. Dans ce papier, les auteurs utilisent un CNN afin de classifier les blocs de telles images, selon s'ils sont bruités, non bruités, ou bruité car issu du bruitage du bloc le précédent.

Value	Description	Correction
0	"pixel block considered as clear"	complete
1	"pixel block considered as probably incorrectly decrypted pixel block due to noise spreading from the previous pixel block during decryption"	in progress
2	"pixel block considered as probably incorrectly decrypted pixel block due to noise corruption during transmission/storage or noise spreading from the previous pixel block during decryption"	to correct later

TABLE II: Pixel block states $state(i)$ meaning.

Tableau du papier résumant la classification effectuée.

Cette technique se base sur une réponse précise apportée à un algorithme précis de chiffrement, avec une image chiffrée ayant été bruitée suivant un taux d'erreurs binaires de 10^{-3}.

On pourrait alors se demander si une approche similaire pourrait fonctionner pour d'autres types de chiffrement. On pourrait aussi imaginer des réponses précises de débruitage pour des algos particulier de chiffrement.
Ainsi, nous voulons approfondir nos connaissances dans le domaine du chiffrement, et notamment concernant le chiffrement par clés privées de Shannon. Nous pourrons

alors tester la robustesse de ces algos sur différents types de bruit, et ainsi établir des solutions concrètes de débruitage.

- Au niveau du code :

Nous devons encore continuer dans l'implémentation de méthode de traitement d'images :

- Mise en place de bruit par opérateur sur les bits.
- Mise en place de bruit par blocs.
- Mise en place de chiffrement simples (XOR).
- Mise en place de chiffrement par bloc type AES.
- Mise en place de chiffrement par méthode de clé privée de Shamir.
- Comprendre plus en détail l'implémentation des auto encodeurs, surtout variationnels.
- Mise en place d'une interface afin de tester nos résultats sur un panel de personnes.