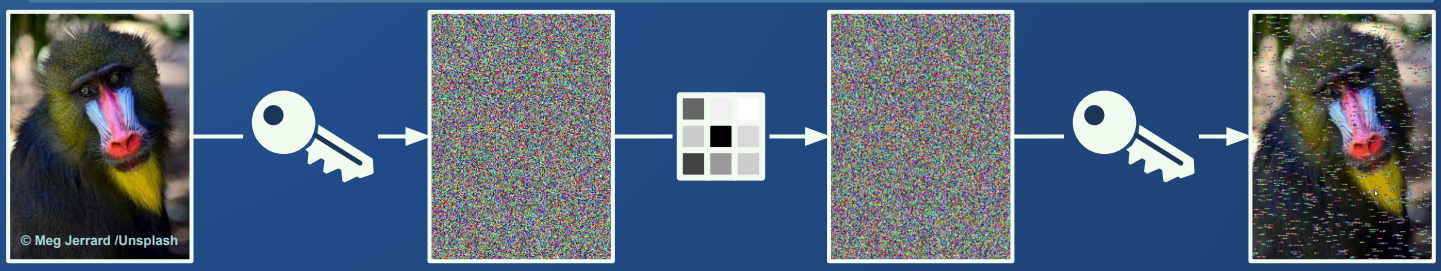
Débruitage par CNN d'images chiffrées ou secrètes bruitées

Par
Ange CLÉMENT - Erwan REINDERS
Sous la direction de
William PUECH - Bianca JANSEN van RENSBURG - Nicolas DIBOT - Pauline PUTEAUX

Problématique

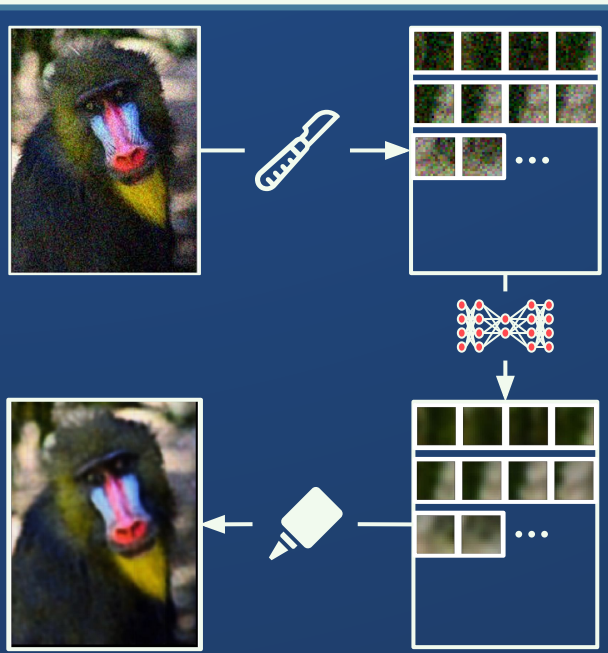
De nos jours, de plus en plus de données sont transmises sur le WEB, ou sauvegardées sur des serveurs comme les CLOUDS par exemple. Il devient alors nécessaire de se poser la question de la **sécurité** de telles communications ou de telles conservations d'informations. Une manière d'y répondre et de mettre en place un **chiffrement** de ces données. Ainsi, même si une personne malveillante arrive à intercepter ces informations, elle ne peut pas reconstruire le message en clair. Cependant, il est possible que pendant la communication (e.g. : bruit de canal) ou lors du stockage de ces données (e.g. : compression avec perte des images chiffrées), ces informations chiffrées soient **altérées**, rendant compliqué le déchiffrement ensuite. C'est sur ce constat que notre projet se base pour tenter d'établir des techniques de **débruitage** sur de telles données altérées.



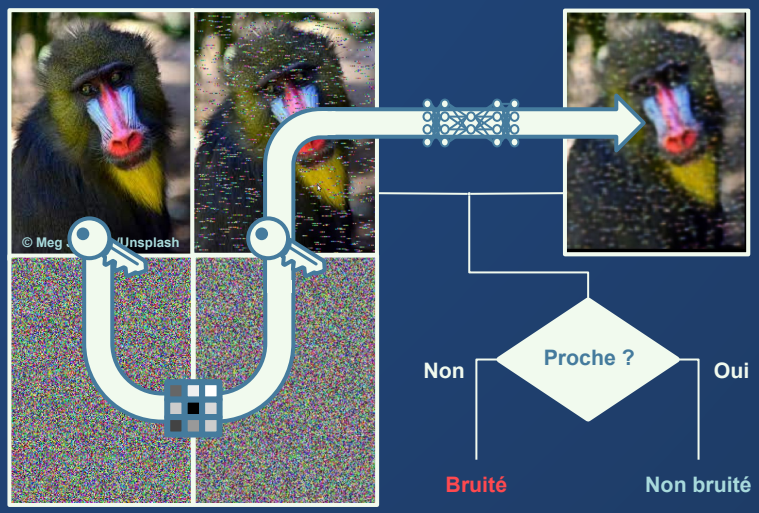
Ainsi, nous pouvons formaliser ce problème via le schéma au-dessus. Il ne s'agit pas simplement de débruiter une image, mais de la débruiter en connaissance des informations sur le chiffrement appliqué (connaissance de l'image chiffrée et de l'image déchiffrée bruitées).

Auto-encodeur approximation

Une des manières de débruiter une image est d'utiliser un réseau de neurones convolutif particulier, un **auto-encodeur**. Il s'agit d'un réseau de neurones par apprentissage non supervisé de type génératif. Ce type de réseau prend dans notre cas une image bruitée en entrée, pour tenter d'en générer sa version la moins bruitée possible en sortie.



Avec cette technique, on a pu constater que l'on arrive à reconstruire l'image d'origine (présence des bonnes **couleurs**, du **sujet**, des **formes**), mais que l'on obtient un résultat flou ; on a du mal, au moment de la reconstruction, à préserver des informations de **texture** notamment.
Pour tenter de conserver cette information, on pourrait se servir des résultats du réseau de neurones afin d'obtenir une image depuis laquelle il nous deviendrait plus facile de détecter des parties bruitées de l'image déchiffrée.



En connaissance de l'algorithme de chiffrement appliqué, ou en faisant certaines **suppositions** sur le bruit appliqué, il nous serait alors possible de **corriger** plus facilement ces zones détectées comme bruitées.
C'est l'avantage de cette stratégie de correction de bruit, contrairement à des techniques de débruitage plus classiques comme des filtrations ou l'utilisation d'auto-encodeurs, qui s'emploient dans des **cadres plus généraux de débruitage d'images**.

Shamir

Jusqu'à présent, il a été question, dans les schémas, de chiffrements **symétriques par clef secrète**. Une autre manière de sécuriser une image est de passer par un **partage de secrets**. On va ainsi partitionner l'image à chiffrer en **n parts** différentes de telle sorte que **k (parmi n) parts** soient suffisantes pour la reconstituer. Cela va se baser sur une utilisation d'un **polynôme** de degrés (k-1) pour chaque pixel, où chaque part va stocker la valeur d'un point sur la courbe de ce polynôme, de telle sorte que la valeur en zéro de ce polynôme soit la valeur du pixel que l'on cherche à cacher. Cela permet une **meilleure responsabilisation** de la part des personnes partageant une part de l'image à sécuriser, mais augmente les sources potentielles de bruit.

Étape de chiffrement/déchiffrement
Bruit sur l'image
Passage à l'auto-encodeur
Segmentation de l'image en blocs
Assemblage des différents blocs