

Rapport intermédiaire IMA - Compression JPG

Maël Le Guillouzic - Ivan Khodakov

Description du projet

Le projet consiste à la détection de compression, mais aussi d'opération de falsification d'images jpg.

Pour cela la première étape est de détecter la présence de « grilles d'origine ». Dans un second temps, grâce à ces informations, il est possible de détecter des opérations de compression ou de falsification.

Qu'est-ce qu'une grille d'origine ? Une image est découpée en bloc de 8x8 pixels avant d'être compressée jpg ou même falsifiée. Il existe donc une multitude façons de découper les pixels de l'image en différents blocs. La grille d'origine est celle utilisée pour la compression.

Comment fonctionne la compression jpg ? L'objectif est de supprimer les hautes fréquences de l'image en luminescence. Pour cela, la dct est appliquée à chaque bloc de 8x8 pixels. En fonction du coefficient de compression voulu, les coefficients de fréquences sont plus ou moins mis à zéro. En appliquant la dct inverse, l'image est ainsi compressée.

Comment détecter cette grille d'origine ? On applique la dct à tous les blocs possibles de l'image. On calcule ensuite le nombre de pixels égaux à 0 par blocs. Les blocs avec le plus grand nombre de pixels à 0 sont ceux le plus susceptibles d'être présents dans la grille d'origine de l'image. Ainsi chacun des blocs "votent", et aident ainsi à déterminer la grille la plus vraisemblable. Nous ne prenons pas en compte pour les votes les blocs où les pixels sont constants verticalement ou horizontalement (ce qui arrive souvent après la compression, avec des zones de floue par exemple), car ces blocs sont de très basses fréquences, le calcul du nombre de 0 n'y est donc pas approprié. Il est nécessaire aussi de pouvoir confirmer qu'il s'agit bien d'une compression, après avoir déterminé la grille d'origine, car même une image non compressée, renverrait une grille en passant dans cet algorithme.

Pour cela, on se sert du NFA (Nombre de Fausse Alarmes), il s'agit de la probabilité d'avoir un nombre de votes supérieur ou égal à celui de la grille trouvée, dans le cas d'une distribution aléatoire (par exemple dans le cas où l'image n'est pas compressée). Si cette probabilité est assez faible, alors on admet qu'il y a eu compression jpg, et que la grille d'origine trouvée est la bonne.

Comment détecter les falsifications de l'image ? Il existe de nombreux cas, mais par exemple dans le cas où l'on trouve deux grilles vraisemblables, cela peut être liée à l'ajout d'une image dans l'autre.

Description des travaux réalisés

Nous avons commencé par lire en profondeur la publication plusieurs fois. Nous avons ensuite pris le temps de lire le code en langage c et de faire le lien avec le document. Nous avons aussi commencé à coder en python, l'algorithme de détection de la grille d'origine et créé un dépôt git pour le projet. Nous sommes en pleine phase de tests de l'algorithme en question, et sommes en train de l'améliorer et le compléter.

Planning du travail à accomplir

semaine du 16/10 : Compléter l'algorithme en python.

semaine du 23/10 : Ajout de fonctionnalités pour la détection de falsifications et tests de l'algorithme final obtenu.

semaine de 30/10 : Mise en place d'un support attrayant pour le test du fonctionnement de l'algorithme. Par exemple, une application.

semaine du 6/11 : Préparation du rapport final

semaine du 13/11 : soutenances de projets.