Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique
Département Informatique

Master 2 SII

VISION PAR ORDINATEUR

Partie 1

-Cacher un texte dans une image-

Réalisé par :

NOUAR	**S**ofia	171731058542	G3
BOUBAKEUR	**I**slam	201500008469	G3

2021 / 2022

Introduction

Dans cette première partie du projet, nous allons nous intéresser à cacher un texte dans une image quelconque et pouvoir aussi le restituer à la fin. En effet, ce type de traitement est appelé la ***stéganographie*** qui veut dire faire passer un message d'une manière inaperçu.

Une image est un ensemble de pixels chacun représenté sur un nombre fixe de bits (8bits, 16bits…). Le bit ***MSB*** est le bit du poids fort Le bit ***LSB*** est le bit du poids faible.

MSB	177						LSB
1	0	1	1	0	0	0	0

Le bit le moins significatif LSB sera utilisé pour **cacher** un texte en code ASCII (8 bits pour chaque caractère).

Par exemple : le message USTHB

lettre	U	S	...
code ascii	55	53	
ascii -> binaire	0101 0101	0101 0011	

Chaque bit du message (de chaque lettre) va être caché dans le bit LSB de chaque pixels.

Exemple : <cacher le premier bit du msg>

pixel 1 : 1 0 1 0 1 0 1 1 --------> pixel 1 : 1 0 1 0 1 0 1 0

On fait de même pour tous les pixels.

Environnement de travail

Pour implémenter ce projet, nous utilisons le langage de programmation *Python* et la bibliothèque ***Opencv*** *(cv2)* et ***pyqt5*** pour implémenter l'interface.

Implementation

En effet, pour pouvoir implémenter la fonction principale qui va nous permettre de cacher un texte dans une image nous avons besoin d'un certain nombre de fonctions supplémentaires.

Pour faciliter le décodage et repérer facilement la fin du message, nous avons inséré à la fin du codage du message le caractère ø qui correspond au codage ascii 248.

```
#Codage : cacher un texte dans une image

def CodeMessage(image)  permet de cacher un message dans une image en suivant les étapes suivantes :

1- Transformer le message en binaire en utilisant la fonction messageToBinary, et le mettre dans une liste binary_list.
2- Ajouter le caractère d'arrêt ø à binary_list.
3- Création d'une image vide qui va contenir le message np.zeros()
4- Convertir les pixels de l'image en binaires grâce à la fonction binaryImage.
5- Cacher le message dans l'image résultante de binaryImage grâce à la fonction codeBinaryImage.
6- Reconvertir l'image binaire obtenue après avoir caché le message en pixels avec la fonction makeFinalImage.
```

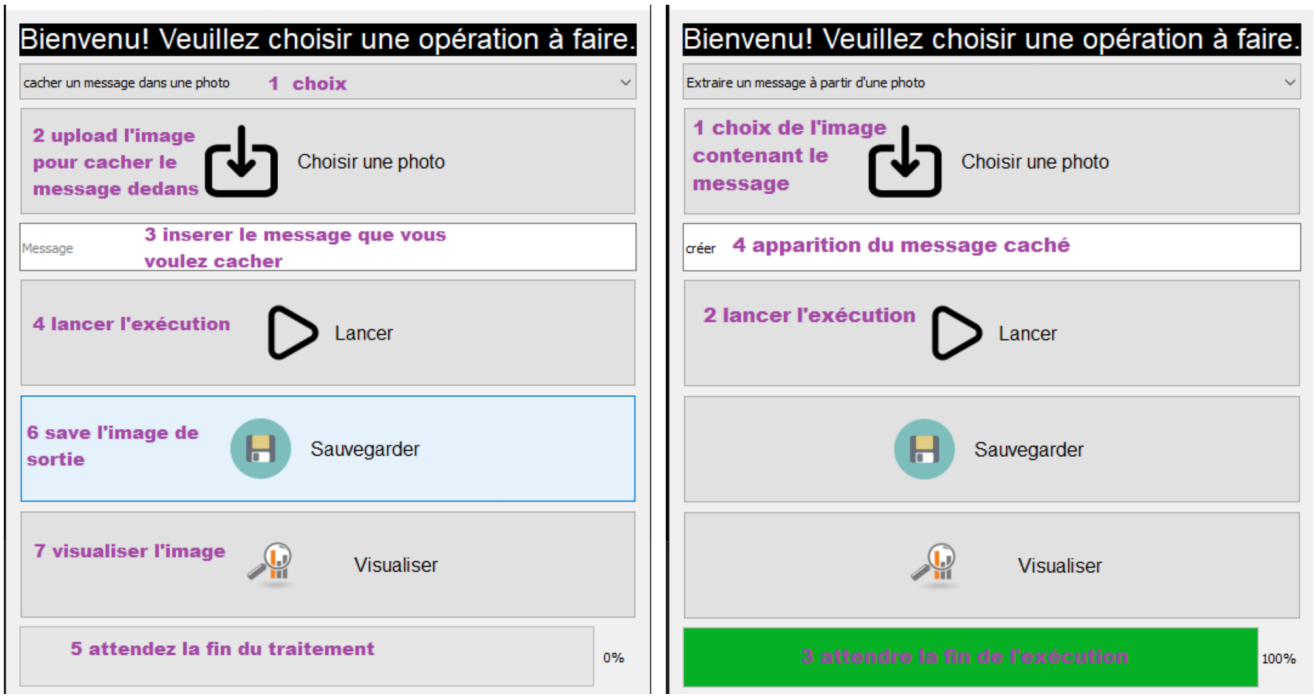
-pseudo-code pour cacher un message dans une image -

```
#Codage : cacher un texte dans une image

def unCodeMessage(image)  permet de récupérer un message cacher dans une image en suivant les étapes suivantes :

1- Convertir les pixels de l'image en binaires grâce à la fonction binaryImage.
2- Restituer le message avec la fonction getMessageFromImage qui va parcourir chaque pixel (converti en binaire) et récupérer le dernier bit, puis les coder 8 à 8 en ascii, et ce jusqu'à retrouver le caractère de fin de message ø.
```

-pseudo-code pour restituer un message cacher dans une image -

Interface

En utilisant la bibliothèque pyqt5 de python, nous avons implémenté une interface permettant d'exécuter les deux actions : cacher un message dans une image (figure de gauche) et récupérer un message caché dans une image (figure de droite).



-Figure 1 : interface permettant de cacher un message-