APPLICATION DE STEGANOGRAPHIE

PAR **ELOFIR ZAKARIA**

CYBERSECURITY

Encadrement:
Dr.**Younes Dhassi**

TABLE DES MATIÈRES

INTRODUCTION

La stéganographie est une technique liée à la dissimulation de messages dans un média particulier.

Les médias couramment utilisés sont l'image, la vidéo, l'audio et le texte.

Le message caché est généralement sous forme de texte, mais cela n'exclut pas la possibilité qu'il s'agisse d'un autre type de média, comme une image ou un fichier audio.

L'image est le média le plus fréquemment utilisé pour dissimuler des messages.

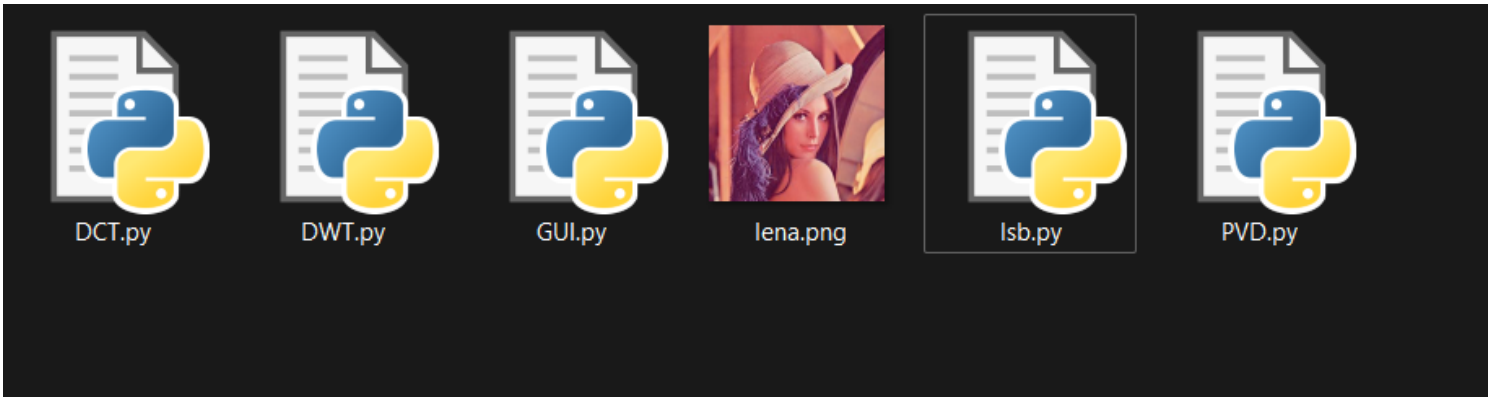
Une image se compose de pixels, chacun contenant des bits de couleur. La technique de stéganographie courante est la méthode du domaine spatial, c'est-à-dire la dissimulation du message dans une image en remplaçant certains des bits de l'image.

Cette méthode est assez populaire car elle modifie légèrement l'image originale et le message caché a une taille assez importante.

NOTRE APPLICATION

Notre application est un outil doté d'une interface graphique (GUI) conçu pour incorporer et extraire de manière sécurisée des informations dans divers fichiers image. L'application comprend une GUI créée à l'aide de la bibliothèque Tkinter en Python et intègre quatre algorithmes distincts :

- LSB (Least Significant Bit)
- DWT (Discrete Wavelet Transform)
- PVD (Pixel Value Differencing)
- DCT (Discrete Cosine Transform).



NOTRE APPLICATION

Caractéristiques :

- **Section Embed (Incorporer) :**
  - Les utilisateurs peuvent sélectionner un fichier image en entrée et choisir d'incorporer du texte brut ou des fichiers.
  - Les algorithmes d'incorporation disponibles comprennent LSB, DWT, PVD et DCT, offrant ainsi une flexibilité et une personnalisation.
  - L'application offre un aperçu en temps réel des images d'entrée et de sortie pour une expérience d'incorporation transparente.
  - Les utilisateurs peuvent spécifier le répertoire de sortie pour stocker l'image incorporée.

- **Section Extract Data (Extraire les données) :**
  - L'extraction prend en charge les quatre algorithmes : LSB, DWT, PVD et DCT, assurant ainsi une compatibilité avec les méthodes d'incorporation.
  - Les utilisateurs peuvent sélectionner le fichier à partir duquel extraire les données et choisir l'algorithme d'extraction souhaité.
  - Les données extraites peuvent être affichées sous forme de texte brut dans l'interface utilisateur ou enregistrées dans un fichier.

NOTRE APPLICATION

Interface graphique :

- L'interface utilisateur est intuitive, avec des boutons interactifs et des menus pour une navigation fluide.

La structure modulaire de l'application, intégrant des fichiers distincts pour les méthodes LSB, DWT, PVD et DCT, favorise la maintenabilité et l'extensibilité.

l'application a pour usage l'incorporation et l'extraction sécurisées de données au sein d'images.

MISE EN RELIEF DES ALGORITHMES UTILISÉS

Stéganographie par bits de poids faible (LSB) :

Le code met en œuvre une classe Python, LSBSteg, pour effectuer la stéganographie par bits de poids faible (LSB) sur des images. La stéganographie LSB consiste à cacher des informations en modifiant les bits de poids faible des valeurs de pixels dans une image.

Encodage :

La classe propose des méthodes pour encoder à la fois du texte et des données binaires dans une image.
L'encodage de texte implique la conversion des caractères en leur représentation binaire, puis leur incorporation dans les bits de poids faible de l'image.
L'encodage binaire permet de cacher n'importe quel type de fichier tout en préservant sa structure.

Décodage :

Le processus de décodage extrait les informations cachées des bits de poids faible des valeurs de pixels dans l'image.

Utilisation :

- La fonction encodeImage prend une image en entrée, des données à cacher, et produit une image avec les données incorporées.
- La fonction decodeImage prend une image en entrée et retourne les données cachées extraites de l'image.

Remarque :

La stéganographie LSB est souvent utilisée pour cacher des informations dans des images de manière imperceptible, mais il est important de noter que cette méthode peut être sensible aux pertes si l'image est compressée avec des formats tels que JPEG.

MISE EN RELIEF DES ALGORITHMES UTILISÉS

PVD (Pixel Value Differencing) :

Le code implémente la bibliothèque PVD (Pixel Value Differencing) pour effectuer la stéganographie en utilisant les différences de valeurs de pixels dans une image. Cette technique permet d'encoder des données dans les bits de poids faible des pixels d'une image.

- **Classe file_bits_reader :**
  - Cette classe permet de lire les bits à partir d'une séquence d'octets.
  - Elle prend en charge la lecture de bits de 1 à 8 à partir de données binaires.
- **Classe file_bits_writer :**
  - Cette classe facilite l'écriture de bits dans un fichier binaire.
  - Elle gère l'écriture de bits de 1 à 8, en prenant en compte la fin de fichier (EOF).
- **Classe pvd_lib :**
  - Cette classe met en œuvre la logique de la bibliothèque PVD.
  - La méthode embed_data prend une image de référence, un message secret et un chemin de sortie, puis effectue l'encodage en modifiant les bits de poids faible des pixels.
  - La méthode extract_data extrait les données encodées d'une image PVD, en utilisant une image de référence pour comparer les différences de valeurs de pixels.
- **Constantes PVD :**
  - Le code utilise des constantes telles que PVD_MAGIC, PVD_VERSION, etc., pour définir les informations de l'en-tête PVD.
- **Remarques :**
  - La bibliothèque PVD utilise la différence de valeurs de pixels pour déterminer combien de bits peuvent être utilisés pour encoder les données dans chaque composante RGB.
  - Les classes file_bits_reader et file_bits_writer fournissent des fonctionnalités génériques de lecture et d'écriture de bits qui peuvent être utilisées pour d'autres besoins.
- **Utilisation :**
  - pvd_lib.embed_data : pour encoder des données dans une image.
  - pvd_lib.extract_data : pour extraire les données encodées d'une image PVD.

MISE EN RELIEF DES ALGORITHMES UTILISÉS

Classe DCT (Transformée en cosinus discrète)

La classe fournie implémente une technique de stéganographie basée sur la transformée en cosinus discrète (DCT). les principales fonctions et caractéristiques de la classe :

- Méthode __init__ :
  - Le constructeur initialise les attributs de la classe, tels que le message, la représentation binaire du message (bitMess), le nombre d'octets d'origine (oriCol et oriRow), et le nombre total de bits du message (numBits).
- Méthode dctenc (Encodage) :
  - Cette méthode prend une image (img) et un message secret à cacher.
  - Le message est d'abord converti en binaire et préfixé par sa longueur, puis divisé en blocs de 8x8 pixels.
  - Les blocs sont soumis à une DCT, puis à une quantification en utilisant une table de quantification prédéfinie.
  - Les bits de poids faible du coefficient DC (composante de basse fréquence) dans chaque bloc sont remplacés par les bits du message.
  - Les blocs quantifiés sont inversés pour obtenir l'image modifiée.
- Méthode dctdec (Décodage) :
  - Cette méthode prend une image modifiée et récupère le message caché.
  - Les blocs de l'image sont soumis à une DCT inverse et à une déquantification.
  - Les bits de poids faible du coefficient DC de chaque bloc sont extraits pour reconstruire le message original.
- Méthode chunks :
  - Cette méthode est un utilitaire pour diviser une liste en morceaux de taille fixe.
- Méthode addPadd :
  - Cette méthode ajoute un rembourrage à une image pour la rendre divisible par 8x8.
- Méthode toBits :
  - Cette méthode convertit le message en une liste de chaînes binaires représentant les octets ASCII.

Remarques :

- La classe effectue l'encodage en modifiant les bits de la composante DC des blocs 8x8 de l'image, ce qui est une approche courante pour la stéganographie basée sur la DCT.
- Le décodage extrait les bits de poids faible du DC de chaque bloc pour reconstruire le message.
- Les blocs sont soumis à une quantification basée sur une table de quantification prédéfinie, une pratique courante pour réduire la capacité de détection de la stéganographie.
- Le message est préfixé par sa longueur pour faciliter l'extraction.

MISE EN RELIEF DES ALGORITHMES UTILISÉS

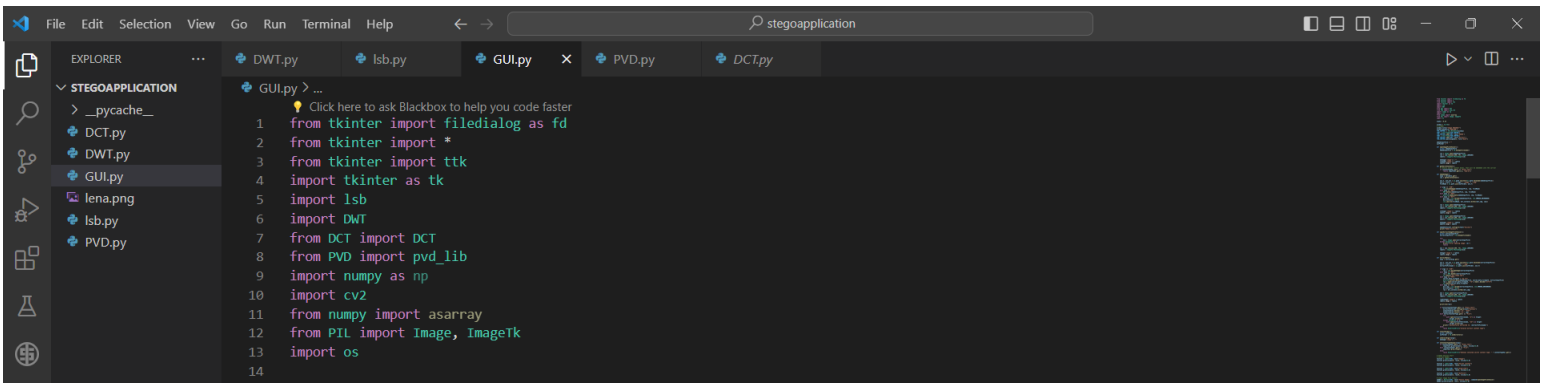
Steganographie par l'Ondelette Discrète (DWT) :

- **Fonction dwtenc :**
  - Applique la transformation en ondelettes discrètes (DWT) à une image pour encoder un message.
  - Les coefficients DWT sont modifiés en fonction des bits du message.
  - Les coefficients modifiés sont inversés pour obtenir l'image encodée.
  - Les métadonnées et le schéma de codage sont sauvegardés pour le décodage ultérieur.

- **Fonction dwtdec :**
  - Applique la transformation en ondelettes discrètes (DWT) inverse pour extraire le message de l'image encodée.

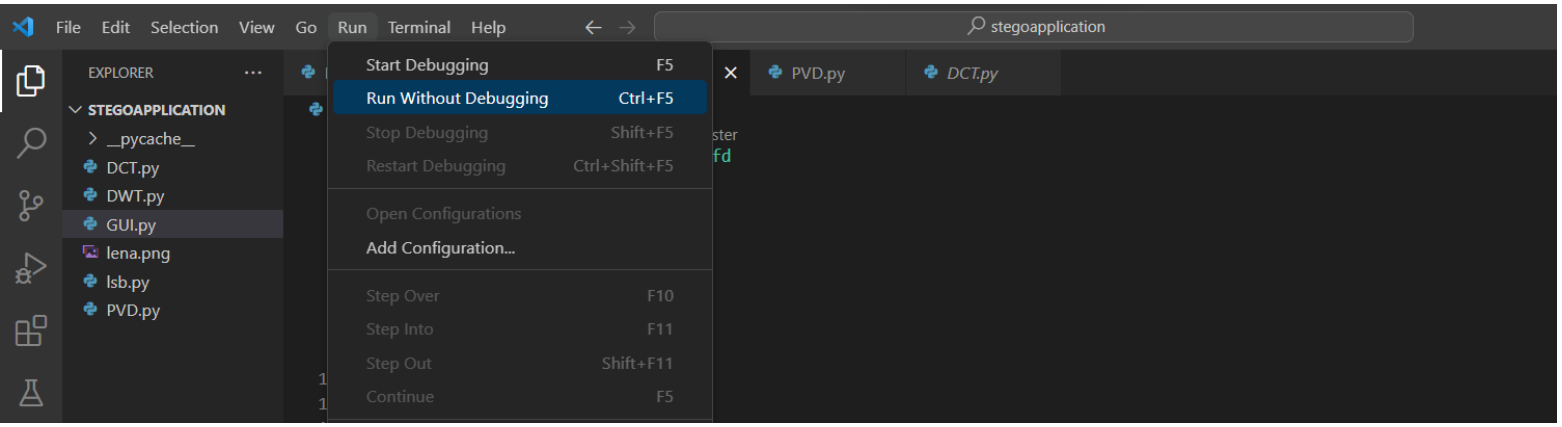
Remarques :

- Ces fonctions implémentent une approche en utilisant cette fois-ci la DWT pour la transformation fréquentielle.

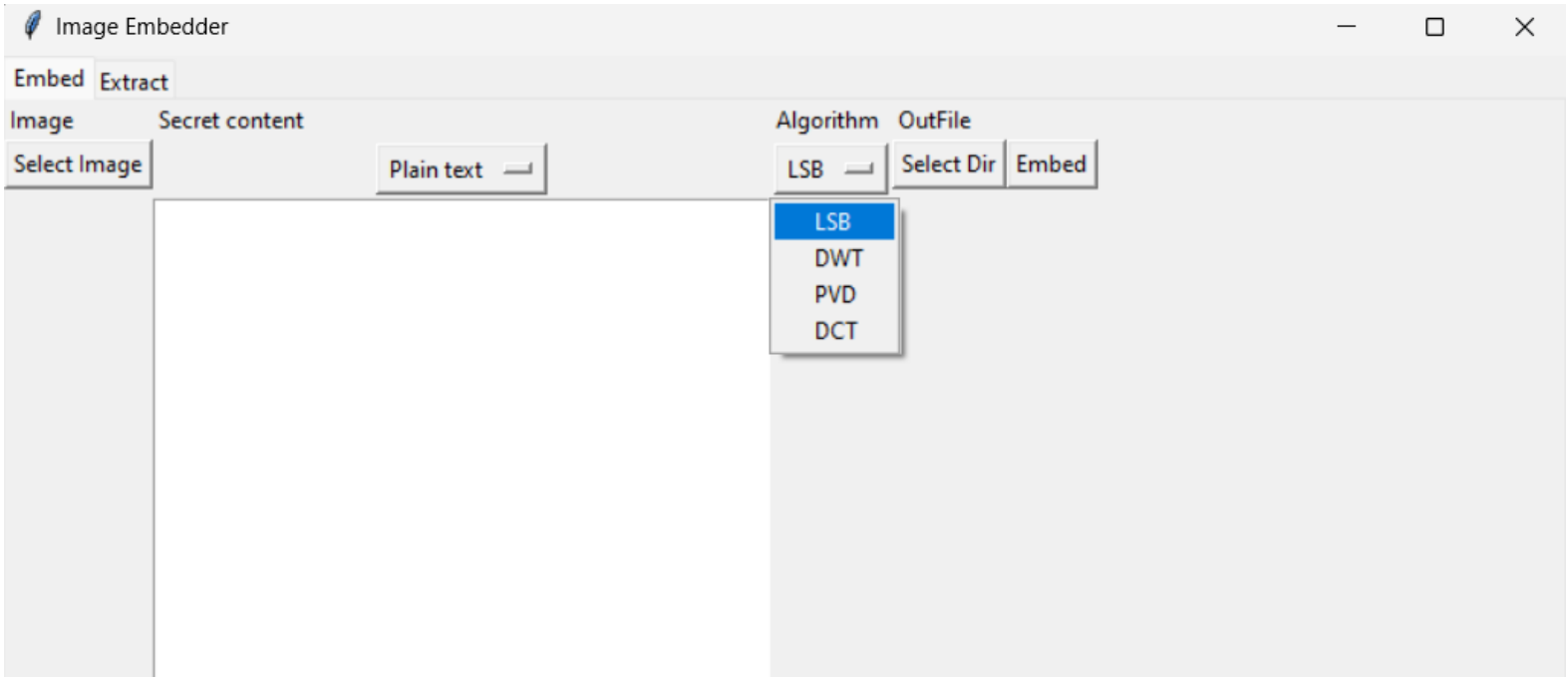


FONCTIONNEMENT

D'ABORD ON EXECUTE NOTRE APPLICATION:

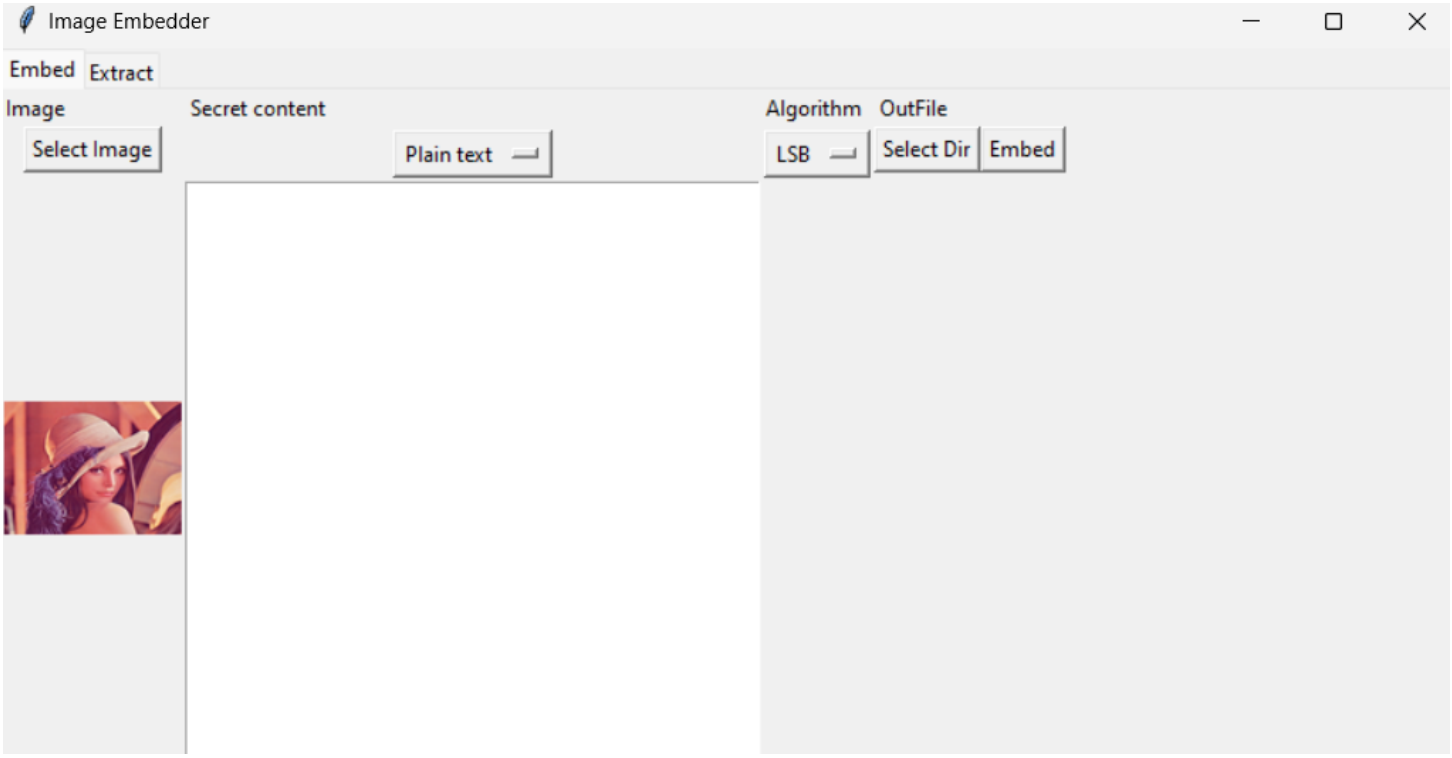


VOILA NOTRE INTERFACE GRAPHIQUE SIMPLE A UTILISER QUI PERMET LES DEUX OPERATION "EMBEDDING' ET "EXTRACTING" ON UTLISANT LES 4 METHODES QU'ON A VU DANS LE COURS NOTAMMENT LSB,PVD,DCT ET DWT

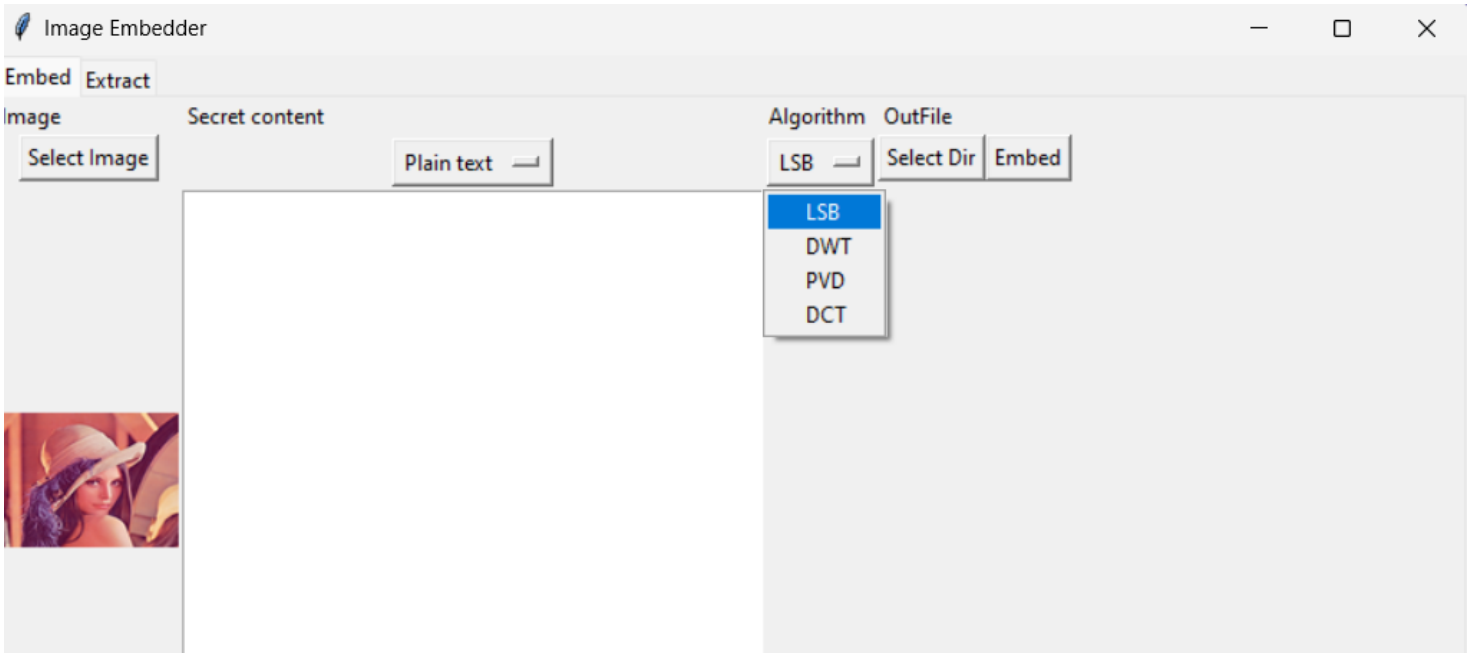


FONCTIONNEMENT

ON SELECTIONNE L'IMAGE SUPPORT OU ON VA DISSIMULER LE MESSAGE ICI ON A CHOISI 'LENA'

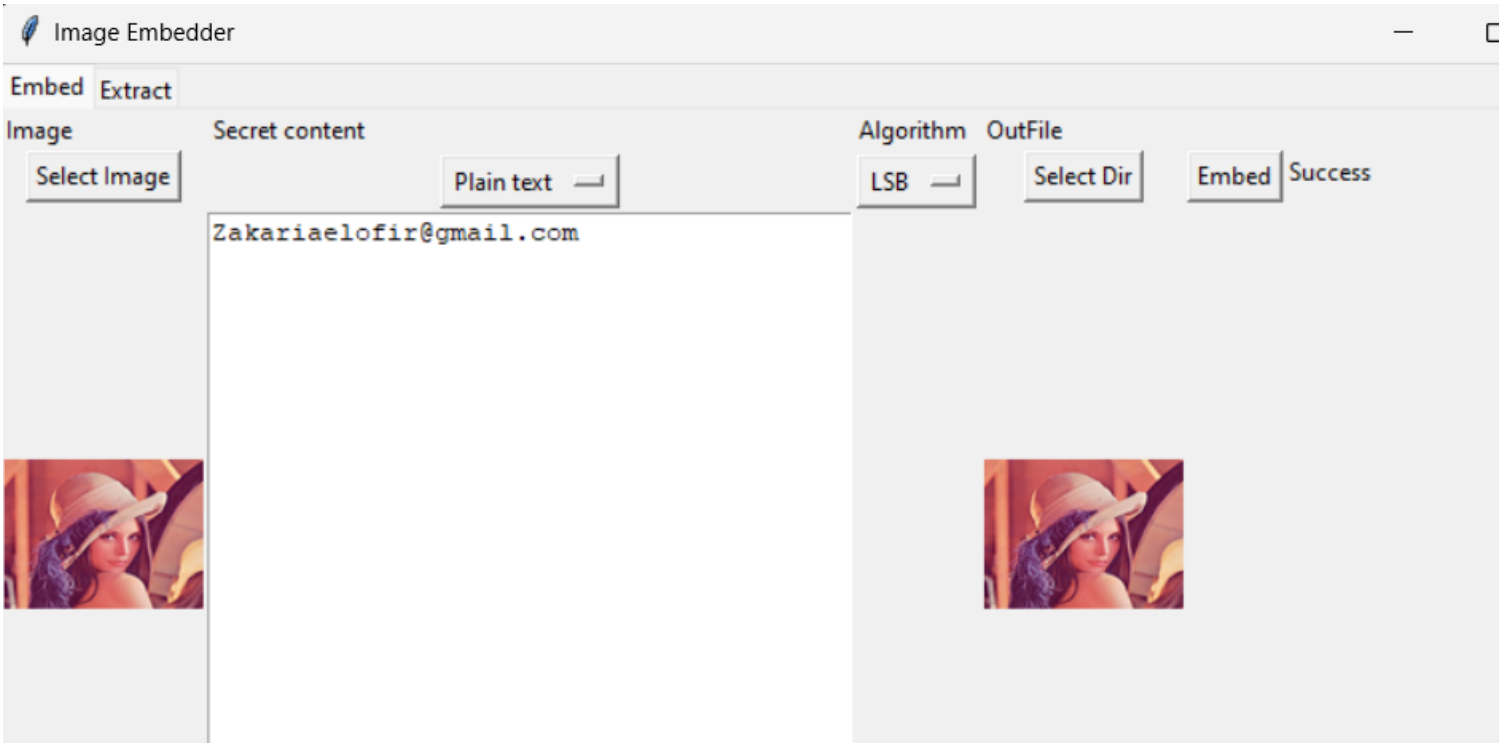


ON CHOISI LA METHODE VOULUE POUR LE PROCESSUS DE STEGANOGRAPHIE ICI ON A OPTE POUR LSB

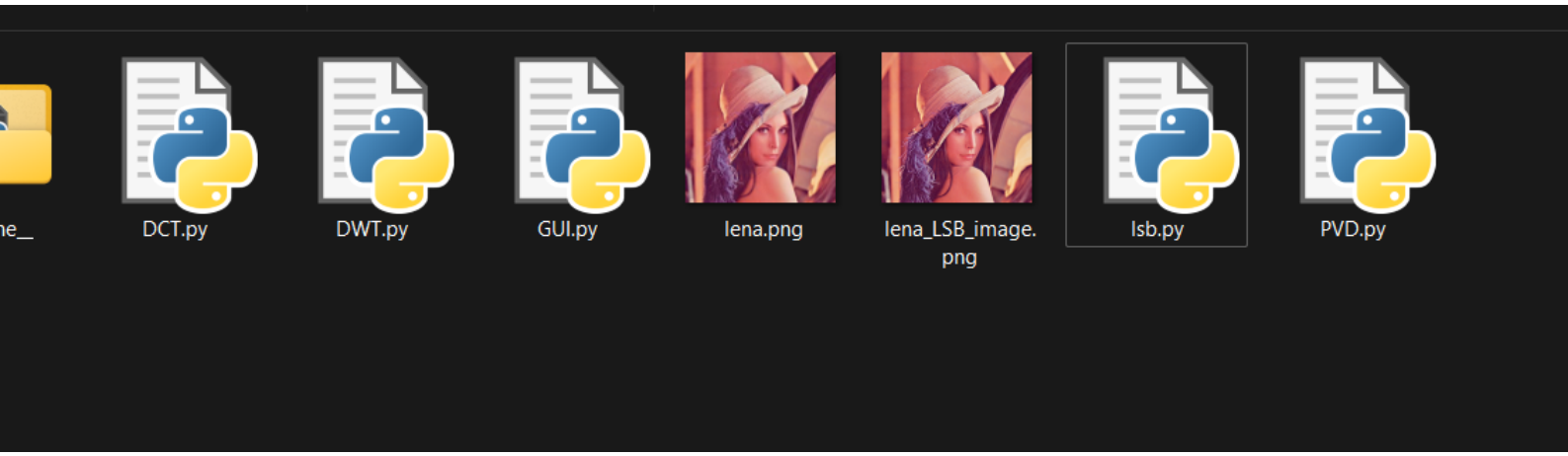


FONCTIONNEMENT

On insere le message souhaite .ici c'est Zakariaelofir@gmail.com



ON RECUPERE L'IMAGE STEGO LENA_LSB_IMAGE.PNG DANS NOTRE DOSSIER



FONCTIONNEMENT

Maintenant pour extraction on selcetionne l'image stego pour recuperer ce qu'on a dissimule avant.

Ici on constate qu'on a bien recuperer le message dissimule dans la rubrique extracted data

L'interface a aussi afficher le flag 'Success'

CONCLUSION

Le projet offre une introduction intéressante à la stéganographie en utilisant des techniques bien connues basées à la fois sur les domaines spatial et fréquentiel.

L'application peut être développée davantage en diversifiant le format des entrées et en ajoutant d'autres techniques de stéganographie. Un volet spécial dédié à la stéganalyse et ses méthodes pourrait également être envisagé. En résumé, il s'agit d'un bon prototype pouvant être amélioré en synchronisation avec l'accumulation des connaissances dans ce vaste et divers domaine qu'est la stéganographie et ses techniques.

Nos sincères remerciements au cher professeur qui a su nous passionner pour ce domaine délicat. Sa guidance éclairée a joué un rôle essentiel dans l'exploration de ces concepts complexes et dans le développement de ce projet.