Projets MIAGE-MBDS 2023-2025

SteganographIA

UNIVERSITÉ CÔTE D'AZUR

1. Introduction

1.1 Contexte

Le projet *SteganographIA* est développé dans le cadre du programme Master MIAGE-MBDS 2023-2025, avec pour objectif la création d'une plateforme web innovante dédiée à la gestion, la protection et l'analyse des images. Il s'inscrit dans un projet d'innovation technologique intégrant des outils avancés de vision par ordinateur et des mécanismes de compression.

1.2 Objectifs

- Identifier les images contenant des signatures numériques.
- Ajouter des signatures de protection aux images.
- Détecter les images générées par des modèles d'IA générative.
- Optimiser les processus via des mécanismes de compression et des outils de computer vision.

---

2. Description du projet

2.1 Fonctionnalités principales

i. Identification des images signées

- Scanner les images téléchargées pour détecter la présence d'une signature numérique (watermark ou hash).
- Récupérer et afficher les métadonnées associées à la signature.

ii. Ajout de signatures numériques

- Ajouter des signatures cryptographiques aux images afin d'en garantir l'authenticité et la propriété.
- Permettre un choix entre différents types de signatures (watermark visible/invisible, hash, etc.).

iii. Détection d'images générées par IA génératives

- Identifier les images créées par des outils tels que DALL-E, MidJourney, ou Stable Diffusion.

- Utiliser des algorithmes spécifiques pour détecter les artefacts ou caractéristiques propres aux images générées par IA.

iv. Mécanismes de compression d'images

- Intégrer des algorithmes de compression avancés pour réduire la taille des images tout en préservant leur qualité.
- Proposer différents formats de sortie (JPEG, PNG, WebP, etc.).

2.2 Technologies utilisées

- **Frontend** : Angular ou React pour une interface utilisateur réactive et moderne.
- **Backend** : Node.js (Express) ou Django pour la gestion des API.
- **Base de données** : MongoDB ou PostgreSQL pour le stockage des données.
- **Outils de vision par ordinateur** : OpenCV, TensorFlow ou PyTorch pour l'analyse des images.
- **Hébergement** : Cloud AWS, Azure ou Firebase.

---

3. Architecture technique

3.1 Architecture de la plateforme
// À Discuter

- **Modèle MVC** (Modèle-Vue-Contrôleur) pour une séparation claire des responsabilités.
- Utilisation d'API RESTful pour la communication entre le frontend et le backend.

3.2 Flux de données

1. Upload de l'image par l'utilisateur.
2. Analyse des métadonnées et détection automatique.
3. Stockage sécurisé dans une base de données ou un stockage cloud.
4. Application des traitements (signature, détection IA, compression).

---

4. Exigences fonctionnelles

4.1 Exigences utilisateur

- Interface utilisateur intuitive et responsive.
- Chargement rapide des images et des résultats.
- Tableau de bord pour visualiser les statistiques (ex : nombre d'images signées ou détectées).

4.2 Exigences techniques

- Traitement en temps réel ou quasi-temps réel pour les fonctionnalités de détection.
- Sécurisation des données utilisateurs (authentification et stockage sécurisé).
- Compatibilité avec les principaux navigateurs et appareils mobiles.

---

5. Livrables

5.1. Prototype fonctionnel :

- Interface avec les fonctionnalités principales.

5.2. Documentation technique :

- Guide d'installation, d'utilisation et de maintenance.

5.3. Rapport final :

- Synthèse des résultats obtenus et des perspectives d'amélioration.

---

6. Contraintes

1. Respect des délais du projet (année académique 2023-2025).
2. Gestion des performances pour traiter des images lourdes.
3. Protection des données sensibles (conformité RGPD).

---

7. Planification

1. **Phase 1** : Analyse des besoins (1 semaine).
2. **Phase 2** : Développement des fonctionnalités principales (3 semaines).
3. **Phase 3** : Tests et validations (2 semaines).
4. **Phase 4** : Déploiement et démonstration finale (1 semaine).