



Faculté des sciences de Monastir

Cahier des charges pour le développement du logiciel de gestion des sauvegardes mémoire pour caméra de surveillance avec catégorisation des éléments vidéo

Aymen Meguebli
Raslen Sebai
Mohamed Aziz Guesmi

1 mars 2024

Contents

1	Introduction	2
1.1	Objectif du projet	2
1.2	Parties prenantes	2
2	Description du projet	2
2.1	Optimisation du stockage pour caméras de surveillance	2
2.2	Interface utilisateur conviviale	2
3	Portée du projet	2
3.1	Développement du logiciel	2
3.2	Catégorisation des séquences vidéo	2
4	Exigences fonctionnelles	3
4.1	Détection d'objets avec YOLOv8	3
4.2	Enregistrement sélectif des séquences vidéo	3
4.3	Catégorisation automatique des vidéos	3
4.4	Interface utilisateur intuitive	3
5	Exigences techniques	3
5.1	Utilisation de YOLOv8 pour la détection d'objets	3
5.2	Développement de l'interface utilisateur avec PyQt6	3
5.3	Utilisation de Python comme langage principal	3
5.4	Gestion de version avec Git	3
6	Contraintes de délai et de budget	3
6.1	Délai de livraison	3
6.2	Budget alloué	3
7	Gestion des risques	4
7.1	Risques liés à l'utilisation du logiciel	4
7.2	Risques de configurations variables des caméras	4
8	Conditions de livraison	4
8.1	Livrables	4
8.2	Critères de réussite	4
9	Modalités de validation	4
9.1	Tests d'utilisation par les utilisateurs finaux	4

1 Introduction

1.1 Objectif du projet

Développer un logiciel intermédiaire entre les caméras de surveillance CCTV et le stockage, permettant d'optimiser la mémoire en enregistrant sélectivement des séquences vidéo basées sur la détection d'éléments spécifiques (humains, animaux, véhicules, autres).

1.2 Parties prenantes

Utilisateurs finaux qui installent des caméras de surveillance dans leur maison, leur entreprise ou leur local.

2 Description du projet

2.1 Optimisation du stockage pour caméras de surveillance

Le logiciel doit permettre aux utilisateurs finaux d'optimiser le stockage de leurs caméras de surveillance en enregistrant sélectivement des séquences vidéo basées sur la détection d'objets spécifiques.

2.2 Interface utilisateur conviviale

L'interface utilisateur doit être conviviale et accessible pour les propriétaires de caméras de surveillance.

3 Portée du projet

3.1 Développement du logiciel

Inclut le développement du logiciel avec une interface utilisateur conviviale pour les propriétaires de caméras de surveillance.

3.2 Catégorisation des séquences vidéo

Mise en œuvre de la catégorisation des séquences vidéo dans des dossiers spécifiques.

4 Exigences fonctionnelles

4.1 Détection d'objets avec YOLOv8

Le logiciel doit permettre aux utilisateurs de définir les objets à détecter (humains, animaux, véhicules, autres).

4.2 Enregistrement sélectif des séquences vidéo

Les utilisateurs doivent pouvoir configurer les paramètres d'enregistrement en fonction de la détection d'objets spécifiques.

4.3 Catégorisation automatique des vidéos

Les séquences vidéo doivent être automatiquement classées dans des dossiers spécifiques en fonction de la catégorie détectée.

4.4 Interface utilisateur intuitive

Une interface conviviale doit être fournie pour permettre aux propriétaires de caméras de configurer facilement le logiciel.

5 Exigences techniques

5.1 Utilisation de YOLOv8 pour la détection d'objets

5.2 Développement de l'interface utilisateur avec PyQt6

5.3 Utilisation de Python comme langage principal

5.4 Gestion de version avec Git

6 Contraintes de délai et de budget

6.1 Délai de livraison

01/05/2024

6.2 Budget alloué

—

7 Gestion des risques

7.1 Risques liés à l'utilisation du logiciel

Risque de compréhension et d'utilisation du logiciel par les utilisateurs finaux.

7.2 Risques de configurations variables des caméras

Risques liés à la variabilité des configurations des caméras de surveillance.

8 Conditions de livraison

8.1 Livrables

Logiciel installable, documentation utilisateur, manuel d'installation.

8.2 Critères de réussite

Le logiciel doit être fonctionnel et compréhensible pour les utilisateurs finaux.

9 Modalités de validation

9.1 Tests d'utilisation par les utilisateurs finaux

Le logiciel sera soumis à des tests d'utilisation par des utilisateurs finaux pour évaluer sa convivialité et son efficacité.