

La sécurité est un enjeu majeur aujourd'hui et une **application de la reconnaissance visuelle** à la détection d'arme à feu peut aider les forces de l'ordre ou les particuliers au renforcement de leur sécurité.

Les utilisateurs seront donc divers et sans connaissance particulière de l'informatique, il est ainsi nécessaire de livrer une **interface simple, claire et intuitive** à l'utilisation.

Pour l'utilisation via des caméras de surveillance, une **rapidité** du produit est exigée afin d'avoir un temps de réaction de détection de menace acceptable. Une **alerte** devra être envoyée sur l'écran mais aussi à distance via une notification

Il est nécessaire d'avoir un **seuil minimal de détection** pour éviter les faux-positifs trop fréquents qui perturberaient l'utilisation du produit.

Pour la conception du produit il faut :

Sprint 1 : Récupération de l'image

- Fonctionnalité 0 : Importer une vidéo (en live ou différée) [Mathis]
- Fonctionnalité 1 : transformer la vidéo en plusieurs images à analyser [Luc]

Sprint 2 : Préparation

- Fonctionnalité 2 : Créer une base de données d'images (positives/négatives ou classes) pour l'entraînement d'un Haar Cascade et les tests [Augustin, Elia, Dac-An]
- Fonctionnalité 3 : Entraînement Haar cascade (non concluant) [Elia, Dac-An]

Sprint 3 : Traitement de l'image

- Fonctionnalité 4 : Images processing (grey, resize, Haar cascade) [Semaine 1]
- Fonctionnalité 5 : Classification de l'image via Resnet50 [Augustin]
- Fonctionnalité 6 : Transformation de la classification en détection (fenêtre glissante) [Elia]

Sprint 4 : Interface et finalisation

- Fonctionnalité 7 : Interface graphique [Mathis]
- Fonctionnalité 8: Alerte (notification pop up et mail) [Mathis et Eliott]