

# Cahier des charges

## 1. Objectif

Notre programme se divise en deux grandes parties :

- Une partie statistique : Détermine le nombre de personnes dans un local et détecte leur postures (assise, debout, allongé...) en se basant sur les images captés par une caméra installé dans le robot, et en faisant une tour pour visualiser tout l'endroit.
- Une partie comportementale : Détecte l'attention des personnes envers le robot. Dans cette partie, l'algorithme va se baser sur les images capté par la caméra (qui va être fixé à une position bien déterminé) pour détecter des gestes prédéfinie faite par la personne sur lequel la caméra est fixée.

## 2. Langage de programmation

L'implémentation de notre programme sera sur le langage C.

## 3. Analyse fonctionnelle

FP1 (fonction principale 1)	Création d'un dataset.
FP2 (fonction principale 2)	Evaluation du nombre de personnes dans un bureau.
FP3 (fonction principale 3)	Détection de l'attention des personnes envers le robot.
FS1 (fonction secondaire 1)	Détection des postures (assise, debout, allongée...).

Cahier de charges			
Repère	Fonction	Critère	Niveau
FP1 (fonction principale 1)	Création d'un dataset, c'est un ensemble de séquence d'images pour tester les résultats du logiciel.	- Scenarios différents	- 3 scénarios différents pour tester la fonction qui compte le nombre de personnes dans un local.
		- Scenarios proche de la vie réelle.	- 3 scénarios de la vie courante pour tester la fonction qui détection de l'attention des personnes envers le robot.
FP2 (fonction principale2)	Evaluation du nombre de personnes dans un local.	- Précision et robustesse.	- Dans 90% des cas, l'algorithme est capable de donner le nombre exact des personnes dans le bureau.
		- Rapidité .	- Temps réel (<x s).
FP3 (fonction principale 3)	Détection de l'attention des personnes envers le robot.	- Nombre de gestions.	- Trois gestions.
		- Rapidité.	- Temps réel (<y s).
FC1 (fonction secondaire 2)	Détection des postures (assise, debout, allongée...)	-Nombre de postures.	- Assise, debout.
		- Rapidité.	- Temps réel (<z s).

## 4. Faisabilité technique

Utilisation de la bibliothèque « OpenCV »du langage C pour les traitements d'image.

Utilisation des algorithmes préexistants qui travaille sur la détection de visages.