

**République Algérienne Démocratique et Populaire Ministère de  
L'Enseignement Supérieur et de la Recherche Scientifique  
UNIVERSITE SAAD DAHLEB DE BLIDA**

**Faculté des Sciences  
Département d'Informatique**



**MEMOIRE DE FIN D'ETUDES  
Pour l'obtention  
Du Diplôme de Master en Informatique  
Option : Systèmes Informatiques et Réseaux  
THEME**

---

**Solution IoT pour la détection des fuites de gaz à domicile**

---

**Organisme d'accueil : NOVISOFT**

**Réalisé par : M. TCHERAK Amine et M. OUZNADJI Younes**

**Encadreur : Mme. FEDOUL Saloua (NOVISOFT)**

**Promoteur : M. KAMECHE ABDALLAH HICHAM (Université de BLIDA)**

**Président : Mme. YKHLEF Hadjer (Université de BLIDA)**

**Examineur : Mme. GUESSOUM Dalila (Université de BLIDA)**

***Promotion 2019/2020***

## ملخص

يهدف مشروعنا إلى حماية الأرواح البشرية و إنقاذها من الحوادث المنزلية مثل تسرب الغازات الضارة و وقوع الأشخاص أرضا والتي قد تحدث بشكل يومي، من خلال إنشاء نظاما قادرا على مراقبة مستويات الغازات السامة الموجودة في الهواء، وهذا في الوقت الفعلي، والتدخل آليا على مستوى المنزل عندما يتجاوز أحد الغازات الحد الأدنى المسموح به.

بالإضافة إلى ذلك، يجب أن يضمن النظام أيضا المراقبة الدائمة لسلوك الأشخاص في المنزل، وتنبيه أفراد أسرهم الموجودين خارج المنزل و كذا مصالح الحماية المدنية في حالة حدوث طارئ.

لاختبار قدرة نظامنا على التعرف وبشكل موثوق على مختلف وضعيات الجسم وكذا الوقوع أرضا، قمنا بتسجيل مجموعة من مقاطع الفيديو حيث يمارس الشخص حركات ووضعيات مختلفة.

### الكلمات المفتاحية :

الحوادث المنزلية، تسرب الغازات الضارة، كشف وقوع الأشخاص أرضا، إنترنت الأشياء، التشغيل الآلي، تطبيق ويب.

## Résumé

Notre projet vise à préserver et à sauver des vies humaines contre les accidents domestiques tels que les fuites de gaz nocifs et les chutes qui peuvent se produire quotidiennement, en mettant en place un système qui peut contrôler, en temps réel, les taux de gaz toxiques présents dans l'air et d'intervenir d'une manière automatique au niveau du domicile lorsqu'un des gaz dépasse le seuil de tolérance qui lui est associé.

Par ailleurs, le système doit aussi assurer la surveillance en permanence des comportements de personnes à domicile, en avisant les membres de leur famille se trouvant en dehors de la maison ainsi que les services de secours en cas d'une situation de détresse.

Pour tester la capacité de notre système à reconnaître d'une manière fiable et robuste les différentes postures et par la même les chutes, nous avons enregistré un ensemble de vidéos où la personne exerce divers postures et mouvements du quotidien.

**Mots clés** : accidents domestiques, fuites des gaz nocifs, détection des chutes, Internet des Objets, domotique, application Web.

## **Abstract**

Our project aims to preserve and save human lives from domestic accidents such as harmful gas leaks and falls which can occur on a daily basis, by setting up a system that can monitor in real time the levels of toxic gases present in the air and intervene automatically at home when one of the gases exceeds the tolerance threshold associated with it. In addition, the system must also ensure the permanent monitoring of the behavior of people at home, notifying their family members outside the home as well as the emergency services in the event of a distress situation.

To test the ability of our system to reliably and robustly recognize different postures and thereby falls, we recorded a set of videos where the person does various postures and movements in everyday life.

### **Keywords:**

Domestic accidents, harmful gas leaks, fall detection, Internet of Things, automation, Web application.

## **Remerciements**

*Ce projet n'aurait pu voir le jour sans la bénédiction du Bon Dieu, le Tout Puissant et le Tout Miséricordieux qui a exaucé nos prières en nous prodiguant courage et volonté afin de le réaliser.*

*Tout d'abord, nous souhaitons exprimer notre profonde gratitude et reconnaissance à :*

- *Notre Promoteur M. KAMECHE Abdallah Hicham, d'avoir eu confiance en nous, en acceptant de diriger ce travail ainsi que pour avoir été disponible durant la période de son élaboration. Il a su nous renseigner et nous orienter sur toutes les questions ayant trait à notre projet,*
- *Notre Encadreur Mme. FEDOUL Saloua, pour son accueil chaleureux au sein de l'entreprise NOVISOFT et sa gentillesse. Elle a su nous mettre à l'aise et nous a énormément aidé en nous faisant profiter de son expérience.*

*Enfin nous les remercions vivement, tous les deux, pour leur contribution, par les précieux conseils qu'ils nous ont donné, ainsi que pour le temps qu'ils nous ont consacré pour la réussite de notre travail IN CHAA ALLAH.*

*Nous tenons à remercier les membres du jury pour avoir accepté de juger notre travail tout en les assurant de notre sincère gratitude et profond respect. Nous souhaitons toutefois leur patience et indulgence quant à notre modeste travail.*

*Nos remerciements vont également à l'ensemble de nos enseignants qui nous ont transmis leur savoir et leurs expériences durant tout notre cursus universitaire et qui nous ont permis d'atteindre le niveau d'expertise nécessaire pour la réalisation de ce projet de fin d'études et l'accomplissement de soi. Nous adressons nos remerciements à tous ceux qui de près ou de loin ont contribué à l'élaboration de ce mémoire sans oublier nos parents, tous nos proches et amis, qui nous ont toujours soutenus et encouragé durant toutes ces années.*

*Merci à tous et à toutes.*

## ***Dédicaces***

*Je dédie ce précieux projet à la mémoire de mon cher oncle LAMINE SALIM qui vient brutalement de nous quitter le 12-08-2020 au moment où je voulais lui offrir ce travail. C'est lui qui, en qualité de chargé de la communication à la protection civile m'a procuré les statistiques des dernières années, relatives à mon projet. C'était une personne admirable :il était très généreux, humble, simple et plein de bonté. Repose en paix khalou SALIM.*

*À tous ceux qui me sont chers :*

*À mes chers parents « Samia » et « M'hamed » pour leur amour indéfectible, leur soutien, leurs encouragements, leurs énormes sacrifices, ils m'ont toujours aidé et soutenus durant toutes ces années.*

*À mon frère « Anis » et ma sœur « Asma » pour leurs encouragements, leur soutien, leur aide et leur disponibilité.*

*À toute ma famille : Grands parents, tantes et oncles, cousines et cousins et à tous mes amis.*

***TCHERAK Amine***

## إهداء

إلهي لا يطيب الليل إلا بشكرك و لا يطيب النهار إلا بطاعتك، لك الحمد حتى ترضى و لك الحمد إذا

رضيت و لك الحمد بعد الرضى

إلى من تجرّع الكأس فارغا ليسقيني قطرة حب، إلى من كلت أنامله ليقدّم لنا لحظة سعادة، إلى من  
حصد الأشواك عن دربي ليمهد لي طريق العلم، إلى حكمتي و علمي، إلى أدبي و حلمي، إلى طريقي

المستقيم .. **أبي الغالى**

إلى من أرضعتني الحب و الحنان، إلى رمز الحب و بلبس الشفاء، إلى القلب الناصع بالبياض، إلى من

كافحت و جاهدت لسعادتي و هنائي، إلى من كان دعاؤها سر نجاحي .. **أمي الغالية** حماك الله

أنتما قدوتي في الحياة، فلا شيء أعظم من امرأة عظيمة قامت بتربيتي و أب أفخر دائما عندما يختتم

إسمي بإسمه، أرجوا من الله أن يمد عمركما لتريا ثمارا قد حان قطافها بعد طول انتظار

إلى أمي الثانية، إلى التي لم تبخل علي بشيء من أجل دفعي الى طريق النجاح، إلى التي علمتني كيف

أرتقي سلم الحياة، إلى قدوتي و سندي .. **إلى جدتي خليفى فاطمة**

إلى القلوب الطاهرة الرقيقة و النفوس البريئة، إلى رياحين حياتي..

إخوتي لينة سامي راشة و محمد

إلى من أظهروا لي أجمل ما في الحياة، إلى من كانوا ملاذي و ملجئي في أصعب اللحظات، إلى من

عشت معهم أجمل الذكريات، إلى من ساقهم الله لي، إلى أخوتي الذين لم تلههم أمي..

إلى اصدقائي **عبد الرشيد و فاروق و محسن**

إلى من جاهد معنا لإتمام مذكرتنا في أحسن حال، إلى من تكرم بوقته الثمين و صبر علينا، إلى من نورنا

بعلمه و توجيهاته.. إلى مؤطري الدكتور **كاشم عبد الله هشام و سلوى فذول**

إلى من علمونا حروفا من ذهب و كلمات من درر و عبارات من أسمى العبارات في العلم، إلى من

صاغوا لنا علمهم حروفا و فكرهم منارة تنير لنا مسيرة العلم و النجاح ... إلى أساتذتنا الكرام

**أوزناجي يونس**

# Table des matières

INTRODUCTION GENERALE.....	1
1. Contexte générale.....	2
2. Problématique.....	2
3. Objectifs.....	3
4. Organisation du mémoire.....	4
Chapitre 1 : Détection des fuites de gaz et IoT.....	5
1. Introduction.....	6
2. Introduction à l'IoT.....	6
3. Concepts de base.....	6
4. Travaux liés à la détection des fuites de gaz.....	7
4.1. Détecteurs des fuites de gaz.....	7
4.2. Système de détection de fuites de gaz basé sur GSM.....	9
4.3. Systèmes de détection de fuites de gaz basés sur l'IoT.....	10
5. Conclusion.....	13
Chapitre 2 : Détection de personnes en situation de détresse.....	14
1. Introduction.....	15
2. Détection des chutes.....	15
2.1. Introduction.....	15
2.2. Basés sur les capteurs.....	15
2.3. Basés sur la vidéo.....	21
3. Autres situations de détresse.....	23
3.1. Situation de panique.....	23
3.2. Détresse respiratoire.....	24
3.3. Détresse cardiaque.....	25
4. Conclusion.....	25
Chapitre 3 : Machine Learning.....	27
1. Introduction.....	28
2. Définition.....	28
3. Types de Machine Learning.....	28
3.1. Apprentissage supervisé.....	28
3.2. Apprentissage non supervisé.....	29
3.3. Apprentissage par renforcement.....	29
4. Réseaux de neurones.....	30
4.1. Le neurone biologique.....	30
4.2. La fonction d'activation.....	31



4.3.	Le neurone artificiel .....	32
4.4.	Un réseau de neurones artificiels.....	33
4.5.	La rétropropagation (Backpropagation) .....	34
5.	CNN (Convolutional Neural Network) .....	35
5.1.	Définition .....	35
5.2.	L'opération de convolution .....	36
5.3.	Couche d'entrée (Input layer).....	36
5.4.	Les étapes du traitement effectué dans la couche cachée (Hidden layer).....	37
5.5.	Architectures CNN pour la vision par ordinateur.....	40
5.6.	Le Framework Mask R-CNN .....	44
5.6.1.	Backbone.....	45
5.6.2.	Feature Pyramid Network.....	45
5.6.3.	Region Proposal Network (RPN) .....	45
5.6.4.	RoI Classifier et Bounding Box Regressor.....	47
5.6.5.	RoI pooling.....	47
5.6.6.	Les masques de segmentation .....	47
6.	Conclusion.....	48
Chapitre 4 : Étude de la partie matérielle du projet.....		49
1.	Introduction .....	50
2.	Les choix de la solution électronique .....	50
2.1.	Carte de développement Arduino.....	50
2.2.	Carte de développement Raspberry Pi .....	50
2.3.	Carte de développement NodeMCU .....	51
2.4.	Les Critères de choix de la solution .....	51
3.	Étude comparative.....	52
3.1.	Table comparative .....	52
3.2.	Discussion du tableau comparative .....	52
4.	La carte ESP32 .....	53
4.1.	Présentation de la carte.....	53
4.2.	Caractéristiques principales.....	53
4.3.	Programmation de la carte ESP32.....	53
5.	Capteur utilisé.....	54
5.1.	Capteur de gaz MQ-2 .....	54
6.	Actionneurs utilisés .....	54
6.1.	MG996R Servo Motor.....	55
6.2.	Relais one channel.....	55
6.3.	Piezo buzzer .....	55

6.4.	Module GSM SIM800L .....	56
6.5.	Moteur driver l298.....	56
6.6.	Ventilateur .....	57
7.	Etude socioéconomique.....	58
8.	Conclusion.....	58
Chapitre 5 : Analyse et Conception du système .....		59
1.	Introduction .....	60
2.	Conception et Modélisation du système de détection des fuites de gaz .....	60
2.1.	Le principe de fonctionnement du système de détection des fuites de gaz .....	60
2.2.	Spécification des besoins.....	61
2.2.1.	Besoins fonctionnels.....	61
2.2.2.	Besoins techniques .....	62
2.3.	Conception .....	63
2.3.1.	La conception software.....	63
2.3.2.	Le déploiement du système .....	70
3.	Conception du système de détection des chutes.....	75
3.1.	Le principe de fonctionnement du système de détection des chutes .....	75
3.2.	Extraction de la silhouette du corps humain.....	78
3.2.1.	Obtention de la silhouette en utilisant le Framework Mask R-CNN.....	78
3.3.	Le réseau de neurones convolutif.....	80
3.3.1.	La fonction de perte (loss function).....	80
3.3.2.	L'algorithme d'optimisation.....	81
3.3.3.	Les métriques .....	82
3.3.4.	Image data augmentation.....	84
3.4.	Détermination de la zone représentant le sol.....	85
4.	Conclusion.....	87
Chapitre 6 : Implémentation et Tests.....		88
1.	Introduction .....	89
2.	Implémentation et Tests du système de détection des fuites de gaz.....	89
2.1.	Présentation de la plateforme .....	89
2.1.1.	Module Tableau de bord.....	90
2.1.2.	Module Historique.....	90
2.1.3.	Module Commande .....	91
2.1.4.	Module Gestion .....	92
2.1.5.	Module Alerte.....	92
2.1.6.	Module Paramètre .....	92
2.2.	Tests et Evaluation du système.....	93

2.2.1.	Tests des fonctionnalités du système.....	93
2.3.	Evaluation des performances du système.....	95
3.	Implémentation et tests du système de détection des chutes .....	96
3.1.	La phase d'entraînement (training) du réseau de neurones convolutif.....	96
3.1.1.	Le jeu de données (dataset) .....	96
3.1.2.	Évaluation des performances de notre classificateur.....	98
3.1.3.	Graphes représentant l'évolution des métriques pendant 15 epochs .....	99
3.2.	Tests et évaluation de notre système de détection des chutes .....	100
3.2.1.	Tests de reconnaissance de postures.....	100
3.2.2.	Tests de détection des chutes.....	102
3.2.3.	Tests de notre système de surveillance.....	103
3.3.	Comparaison et validation de notre travail.....	104
3.3.1.	La méthode d'extraction de la silhouette du corps humain .....	105
3.3.2.	Le jeu de données .....	106
3.3.3.	Tests de reconnaissance de postures en utilisant le modèle 4 .....	111
4.	Environnements et Outils de développement .....	111
	ARDUINO IDE .....	112
	XAMPP.....	112
	Visual Studio Code .....	112
	MySQL .....	112
	PHP .....	113
	JavaScript.....	113
	Bootstrap 4.....	113
	FRIZING.....	113
	Google colab.....	114
	Python .....	114
5.	Conclusion.....	114
	Conclusion générale .....	115
	Bibliographie.....	118
	Annexe : Fonctionnement d'un réseau de neurones.....	121

## Liste des figures

Figure 1 Schéma montre le fonctionnement et l'interaction d'un système IoT [1].....	7
Figure 2 Détecteur de gaz de ville et gaz en bouteille [2] .....	7
Figure 3 Détecteur de monoxyde de carbone CO [3].....	8
Figure 4 Détecteur de monoxyde de carbone CO [4].....	9
Figure 5 Architecture du système de capteurs de détection de chute [10] .....	16
Figure 6 Photos de l'expérience [10].....	17
Figure 7 Chute, taux d'échantillonnage 100Hz [10].....	17
Figure 8 Chute, taux d'échantillonnage 33Hz [10].....	17
Figure 9 Chute, taux d'échantillonnage 25Hz [10].....	18
Figure 10 S'asseoir, taux d'échantillonnage 100Hz [10] .....	18
Figure 11 S'accroupir, taux d'échantillonnage 100Hz [10] .....	18
Figure 12 Marcher, taux d'échantillonnage 100Hz [10].....	19
Figure 13 Détecteur de chute basé sur les vibrations du sol [11].....	19
Figure 14 Schéma du principe de fonctionnement du détecteur de chute basé sur les vibrations du sol [11] .....	20
Figure 15 illustration des signes de panique physique [16].....	24
Figure 16 Le sang coule du cœur vers la tête via les artères carotides de chaque côté de la tête [17] .....	24
Figure 17 Illustration d'une difficulté respiratoire [18] .....	25
Figure 18 Exemple d'apprentissage supervisé [19] .....	29
Figure 19 Exemple d'apprentissage non supervisé [19] .....	29
Figure 20 Exemple d'apprentissage par renforcement [19].....	30
Figure 21 L'unité de base d'un cerveau biologique : le neurone [20] .....	30
Figure 22 L'entrée n'a pas encore atteint le seuil [20] .....	31
Figure 23 L'entrée a atteint le seuil [20].....	31
Figure 24 La fonction échelon unité [20].....	31
Figure 25 La fonction sigmoïde [20].....	32
Figure 26 La fonction rampe [21] .....	32
Figure 27 Le fonctionnement d'un neurone artificiel.....	33
Figure 28 Un réseau de neurones biologiques [20] .....	33
Figure 29 Un réseau de neurones artificiels .....	34
Figure 30 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds (avec les erreurs de sorties calculées et propagées vers le réseau).....	35
Figure 31 Schéma général d'un CNN.....	36
Figure 32 Schéma détaillé d'un CNN [22].....	36
Figure 33 Les trois couches d'un CNN [23] .....	37
Figure 34 Transformation d'une image en une matrice [24].....	37
Figure 35 opération de convolution [24].....	38
Figure 36 Application de la fonction d'activation ReLU aux feature maps en sorties d'une couche de convolution [24] .....	38
Figure 37 Appliquer l'opération de pooling aux Feature Maps [24].....	38
Figure 38 Aplatis la matrice Pooled Feature Map en un vecteur à une seule dimension [24].....	39
Figure 39 Représentation de la couche Fully connected [25] .....	39
Figure 40 Une vue globale sur les différentes étapes effectuées par un réseau de neurones convolutif pour classer une image en entrée du réseau [25] .....	40
Figure 41 Principe de fonctionnement de l'algorithme R-CNN - 1 [29] .....	41
Figure 42 Principe de fonctionnement de l'algorithme R-CNN - 2 [29] .....	41
Figure 43 Principe de fonctionnement de l'algorithme Fast R-CNN [29].....	42
Figure 44 Comparaison des algorithmes de détection d'objets [29].....	42

Figure 45 Principe de fonctionnement de l'algorithme Faster R-CNN [29].....	43
Figure 46 Comparaison du temps de test effectué par les algorithmes de détection d'objets [29] .....	44
Figure 47 Le framework Mask R-CNN pour la segmentation d'instance [33].....	44
Figure 48 Illustration simplifiée du réseau backbone [33].....	45
Figure 49 Feature Pyramid Network [33] .....	45
Figure 50 Illustration simplifiée montrant 49 boîtes d'ancrage [33] .....	46
Figure 51 3 boîtes d'ancrage en pointillé et le décalage/redimensionnement qui leur est appliqué pour s'adapter précisément à l'objet (boîtes d'ancrage en solide). Plusieurs ancres peuvent correspondre au même objet [33].....	46
Figure 52 Le feature map ici est issue d'une couche de bas niveau, à titre d'illustration, pour faciliter la compréhension de cette phase [33] .....	47
Figure 53 L'adaptation du masque de segmentation 28x28 pixels représentant l'objet au cadre qui le délimite [33] .....	48
Figure 54 Une carte Arduino [34] .....	50
Figure 55 Une carte Raspberry Pi [35].....	51
Figure 56 Des cartes NodeMCU [36].....	51
Figure 57 Capteur de gaz MQ-2 [37] .....	54
Figure 58 Le servomoteur MG996R [38].....	55
Figure 59 Relais one channel [39].....	55
Figure 60 Piezo buzzer [40] .....	56
Figure 61 Module GSM SIM800L [41] .....	56
Figure 62 Moteur driver l298 [42] .....	57
Figure 63 Ventilateur 12 volts [43] .....	57
Figure 64 Diagramme expliquant d'une manière générale le principe de fonctionnement du système de détection de fuites de gaz .....	61
Figure 65 Diagramme de cas d'utilisation global.....	64
Figure 66 Diagramme de séquence du cas Vérifier la connectivité des cartes ESP32.....	65
Figure 67 Diagramme de séquence du cas Modifier le mode de fonctionnement du système de détection des fuites de gaz.....	67
Figure 68 Diagramme de classes.....	68
Figure 69 La classe Utilisateur.....	69
Figure 70 La classe Setting .....	69
Figure 71 La classe Alert .....	69
Figure 72 Diagramme de composants.....	70
Figure 73 Architecture du système.....	71
Figure 74 Schéma du système de détection des fuites de gaz toxiques.....	72
Figure 75 Schéma du système de contrôle de la vanne principale de gaz.....	72
Figure 76 Schéma du système de contrôle de l'alimentation électrique .....	73
Figure 77 Schéma du système d'envoi des alertes via SMS/Appel .....	73
Figure 78 Schéma du système de contrôle de l'alarme sonore.....	73
Figure 79 Schéma du système de contrôle des fenêtres et du ventilateur .....	74
Figure 80 Le branchement du système global avec ses différents composants.....	74
Figure 81 Diagramme expliquant d'une manière générale le principe de fonctionnement du mode surveillance .....	76
Figure 82 Diagramme expliquant d'une manière générale le principe de fonctionnement du mode détection des chutes .....	77
Figure 83 Illustration du principe de fonctionnement du Framework Mask R-CNN.....	79
Figure 84 Insertion du masque de segmentation 28x28 pixels représentant la silhouette d'une personne dans une image sans arrière-plan 30x30 pixels .....	80
Figure 85 La structure du réseau neuronal convolutif développé.....	80

Figure 86 La perte est élevée dans le modèle de gauche par contre elle est faible dans le modèle de droite.	81
Figure 87 Application du retournement horizontal sur les images du training set	85
Figure 88 Application du retournement vertical sur les images du training set	85
Figure 89 Application de la rotation sur les images du training set	85
Figure 90 La zone représentant le sol (polygone) définie par 04 points P1, P2, P3 et P4	86
Figure 91 Le point Pc représentant le centre du cadre délimitant la personne	86
Figure 92 La personne est allongée à l'intérieur de la zone représentant le sol	87
Figure 93 La personne n'est pas allongée à l'intérieur de la zone représentant le sol	87
Figure 94 Vue globale sur les modules de la plateforme développée	89
Figure 95 Module Tableau de bord qui montre l'évolution des paramètres du système	90
Figure 96 Module historique (la page Charts)	91
Figure 97 Module Commande	91
Figure 98 Module Gestion	92
Figure 99 Module Alerte	92
Figure 100 Module Paramètre	93
Figure 101 Fonction de contrôle du système	93
Figure 102 Fonction d'acquisition des taux actuels des gaz toxiques présents dans l'air	94
Figure 103 Alerte via une alarme sonore et visuelle (la LED rouge est allumée lors d'une situation dangereuse)	94
Figure 104 Alerte via une notification envoyée à l'application Web/Mobile	94
Figure 105 Alerte via un appel téléphonique	95
Figure 106 Graphe représentant l'évolution des taux des gaz nocifs dans le temps	95
Figure 107 Illustration des étapes suivies dans la création de notre jeu de données	97
Figure 108 Graphes représentant l'évolution des métriques pendant 15 epochs	99
Figure 109 Les tests de reconnaissance de postures effectués	101
Figure 110 La personne été allongé dans la zone représentant le sol pendant une durée de 30 secondes	102
Figure 111 Une alerte via un appel et une notification envoyée à l'application Web/Mobile indiquant qu'une personne a chuté	102
Figure 112 Historique des chutes	103
Figure 113 Tests de notre système de surveillance	103
Figure 114 Une alerte via un appel et une notification envoyée à l'application Web/Mobile indiquant qu'une personne maleveillante c'est introduit au domicile	104
Figure 115 Historique des intrusions	104
Figure 116 La nouvelle structure du jeu de données des auteurs de l'article [45] que nous avons modifié	108
Figure 117 Graphes représentant l'évolution des métriques du modèle 4 pendant 25 epochs	110
Figure 118 Tests de reconnaissance de postures en utilisant le modèle 4	111
Figure 119 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds	122
Figure 120 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds (avec les entrées 1.0 et 0.5)	122
Figure 121 Le fonctionnement d'un neurone artificiel (avec les poids des liens reliés à ce neurone)	123
Figure 122 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds (avec les sorties calculées)	124

## Liste des tableaux

Tableau 1 Points forts et faibles du détecteur [2] .....	8
Tableau 2 Points forts et faibles des détecteurs [3] [4] .....	9
Tableau 3 Points forts et faibles des systèmes [5] [6] [7] [8] [9] .....	12
Tableau 4 Points forts et faibles des systèmes [10] [11] .....	21
Tableau 5 Points forts et faibles des systèmes [12] [13] [14].....	23
Tableau 6 Étude comparative de quelques cartes disponibles sur le marché .....	52
Tableau 7 Estimation du prix de revient de notre projet .....	58
Tableau 8 Les pins utilisés dans la carte ESP32.....	75
Tableau 9 Représentation de la matrice de confusion .....	83
Tableau 10 Représentation de la structure de notre jeu de données.....	96
Tableau 11 Le partitionnement des vidéos enregistrées portant création de notre jeu de données .....	97
Tableau 12 Quelques exemples de notre jeu de données .....	98
Tableau 13 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle pendant 11 epochs .....	98
Tableau 14 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle pendant 15 epochs .....	98
Tableau 15 La matrice de confusion décrivant les performances de notre classificateur entraîné et évalué pendant 15 epochs.....	99
Tableau 16 Avantages et inconvénients de la méthode d'extraction de la silhouette du corps humain conçu par les auteurs de l'article [45] .....	106
Tableau 17 Quelques images de postures contenues dans le jeu de données [45] .....	107
Tableau 18 Avantages et inconvénients du jeu de données conçu par les auteurs de l'article [45].....	107
Tableau 19 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 0).....	108
Tableau 20 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 1).....	109
Tableau 21 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 2).....	109
Tableau 22 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 3).....	109
Tableau 23 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 4).....	109
Tableau 24 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 5).....	109
Tableau 25 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 6).....	109
Tableau 26 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 7).....	109
Tableau 27 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 8).....	109
Tableau 28 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 9).....	110

## Liste des abréviations

**SMS** : Short Message Service

**LED** : light-emitting diode

**LCD** : Liquid Crystal Display

**PPM** : Partie Par Million

**IoT** : Internet of Things

**TV** : Télévision

**LIE** : Limite Inférieure d'Explosivité

**GSM** : Global System for Mobile Communications

**VoIP** : Voice Over Internet Protocol

**Email** : Electronic mail

**ML** : Machine Learning

**3D** : Trois Dimensions

**2D** : Deux Dimensions

**IA** : Intelligence artificielle

**TCP/IP** : Transmission Control Protocol/Internet Protocol

**WPA** : Wi-Fi Protected Access

**GPS** : Global Positioning System

**Wi-Fi** : Wireless Fidelity

**CNN** : Convolutional Neural Network



# **INTRODUCTION GENERALE**

## 1. Contexte générale

Chaque année de nombreuses personnes dans le monde décèdent ou sont blessées suite à un accident lié à l'utilisation domestique du gaz, surtout en période hivernale où le recours aux différents appareils chauffants utilisant le gaz (chauffage, chauffe bain, etc...) est éminent.

En Algérie, ce phénomène a atteint des niveaux alarmants, des centaines de citoyens trouvent la mort suite à une intoxication au monoxyde de carbone (*CO*) ou à une explosion/asphyxie due à la fuite de gaz. Selon les statistiques de la protection civile, 100 personnes ont péri en 2018 et pas moins de 145 morts enregistrés durant l'année 2019 et 32 depuis le début du mois de janvier 2020, il ne se passe pas une année sans que ce fléau n'emporte avec lui des familles entières à travers le territoire national.

Pour y faire face, différentes entreprises de l'Etat tel que les médias (chaines télévisées, chaines de la radio, etc...) ainsi que les services de la protection civile ont lancé des campagnes de sensibilisation ayant pour objectif de donner des conseils aux citoyens pour mieux se protéger contre les accidents domestiques du gaz. Ajoutant à cela, l'existence sur le marché de détecteurs de fuite de gaz capables de diminuer le nombre de morts ou blessés. Malheureusement tout cela reste insuffisant face à l'ignorance et l'inconscience de certains citoyens sur l'ampleur de ce danger, et le manque d'efficacité de ces détecteurs.

Par ailleurs, le gaz n'est pas le seul facteur de blessés ou de morts à domicile, d'autres accidents domestiques sont susceptibles de survenir, comme les chutes et les problèmes respiratoires, qui peuvent être aussi le résultat des fuites de gaz nocifs. Ces accidents peuvent se produire à tous les moments de la vie. Les enfants, les adultes et les seniors sont tous concernés, mais passé un certain âge, elles peuvent revêtir une toute autre importance car la majorité des chutes/difficultés respiratoires de personnes âgées se produisent à leur domicile, se retrouvant seules, elles ne peuvent pas appeler à l'aide étant inconscientes ou immobilisées, ce qui va aggraver la situation.

Dans ce contexte, notre contribution consiste à mettre en œuvre un système informatique permettant de contrôler et surveiller en temps réel l'état des différents gaz nocifs présents dans l'air ambiant et d'intervenir manuellement/automatiquement (fermeture de la vanne de gaz, ouverture des fenêtres, actionnement de la ventilation, etc...) lors de perception d'une anomalie. Le système doit permettre aussi la surveillance en temps réel des comportements des personnes à domicile de telle sorte à ce qu'il avertit les services de secours ainsi que les membres de leur famille se trouvant en dehors de la maison en cas d'une situation de détresse, et donc ils peuvent intervenir au moment opportun et sauver une personne d'une mort certaine.

## 2. Problématique

Malgré l'existence des différents détecteurs de gaz toxiques pouvant avertir les locataires d'une maison (via une alarme sonore et visuelle) et les services de la protection civile (via un appel téléphonique) de

l'émanation de gaz, ceux-là restent inefficaces face aux différentes contraintes liées à la vie quotidienne. Nous citerons les exemples suivants :

- Une personne âgée, malentendante, malvoyante ou même une personne en plein sommeil qui n'arrivera ni à entendre le son de l'alarme, ni à voir son indication,
- Un retard causé par des perturbations du trafic routier privant ainsi les services de la protection civile d'intervenir et les personnes en danger d'être secourues en temps opportun.

De plus, les accidents de gaz ne sont pas les seules responsables des blessures ou décès à domicile, les chutes et les malaises respiratoires peuvent s'avérer aussi dévastateurs, car ils peuvent toucher n'importe quelle tranche d'âge et êtres plus dangereux lorsqu'il s'agit d'une personne âgée surtout si elle se retrouve seule à domicile. Des détecteurs de chutes/difficultés respiratoires existent, comme des accéléromètres ou des boutons d'aide, cependant, le problème majeur avec ces types de technologies est que les personnes âgées omettent souvent de les porter, et dans le cas d'un bouton d'aide, cela ne sert à rien si la personne est inconsciente après la chute.

Ce sont là quelques raisons pour lesquelles nous sommes persuadés que les vies humaines peuvent être sauvées grâce à l'utilisation des solutions dites « Smart » intervenant de manière automatique lors des situations dangereuses.

### 3. Objectifs

L'objectif de notre projet de fin d'étude est de réaliser un système informatique qui doit assurer de manière automatique ou manuelle à travers Internet ou localement les fonctionnalités suivantes :

- ❖ Contrôler en temps réel l'état des différents gaz toxiques présents dans l'air et émanant des différents appareils du quotidien (cuisinière, chauffage, chauffe bain, etc.),
- ❖ Configurer les consignes qui régulent le seuil de gaz toxiques à ne pas dépasser,
- ❖ Configurer les actions à entreprendre (fermeture de la vanne principale de gaz, ouverture des fenêtres, actionnement de la ventilation, etc.) lors de perception de cas anormaux (le taux d'un gaz toxique dans l'environnement dépassant le seuil toléré),
- ❖ Alerter via *SMS*/Appel les occupants du domicile lors d'un cas anormal,
- ❖ Alerter les services de la protection civile lors d'une anomalie en fournissant la géolocalisation du domicile,
- ❖ Possibilité d'intervenir manuellement sur les différents actionneurs (via une application *Web* responsive qui s'adapte aux différentes tailles d'écran),
- ❖ Surveiller la qualité de l'air,
- ❖ Désactiver l'alimentation électrique du domicile pour éviter le contact électricité-gaz qui provoque une explosion dévastatrice,

- ❖ Détecter une situation de détresse des personnes en utilisant des capteurs vidéo et des méthodes de *Machine Learning*,
- ❖ Exploiter l'historique des différents états des gaz toxiques enregistrés dans une base de données en visualisant ces données sous forme de graphes.

#### **4. Organisation du mémoire**

Le présent rapport illustre les démarches entreprises afin d'achever notre projet de fin d'études, il constitue donc une sorte de documentation qui peut éventuellement servir comme outil à des individus voulant s'inspirer des systèmes que nous avons développés afin de concevoir d'autres plus améliorés. Il se compose de six chapitres structurés comme suit :

- Chapitre 1 : Détection des fuites de gaz et IoT
- Chapitre 2 : Détection de personnes en situation de détresse
- Chapitre 3 : Machine Learning
- Chapitre 4 : Étude de la partie matérielle du projet
- Chapitre 5 : Analyse et Conception du système
- Chapitre 6 : Implémentation et Tests

# **Chapitre 1 : Détection des fuites de gaz et IoT**

## 1. Introduction

Dans ce premier chapitre, nous allons d'abord introduire la notion de l'Internet des objets (*Internet of Things (IoT)*) et présenter ces concepts de base, ensuite nous étudierons les différents détecteurs traditionnels présents actuellement sur le marché ainsi que des systèmes intelligents basés sur la technologie *IoT* permettant de détecter les fuites de gaz et potentiellement d'exécuter des mesures correctives en cas de danger. Cette étude va nous permettre de ressortir les fonctionnalités manquantes/présentes dans ces systèmes qui doivent être impérativement incluses dans notre système de détection des fuites de gaz. C'est donc une excellente approche pour se lancer plus tard dans la modélisation puis la conception de notre système.

## 2. Introduction à l'IoT

Qui ne connaît pas internet aujourd'hui dans notre société ? ce monstre d'information qui a envahi nos ordinateurs, smartphones et TV, et qui a pu rapprocher le monde et le virtualiser en un petit village. On peut chercher l'information et on a également la possibilité de la partager. L'utilité d'Internet ne s'arrête pas au fait qu'elle soit une source d'information, son utilité principale est la communication directe (vidéo-conférence, *VoIP*, etc.) ou indirect (courriel, etc.). Afin d'assurer cette communication, Il y a une architecture sous-jacente masquée pour l'utilisateur, composée de serveurs, de câbles et même de satellites. Grâce à l'internet, notre monde a beaucoup évolué car elle joue un rôle important rendant l'information disponible à tout moment et à n'importe quel endroit.

La technologie qui offre la possibilité d'accéder à internet a beaucoup évolué, il suffit d'avoir un smartphone et d'être connecté à un réseau internet pour commencer à consommer ou ajouter de l'information.

Depuis les années 90, l'Internet ne cesse de croître à une vitesse exponentielle, et poursuivre son évolution jusqu'à pouvoir connecter les objets qui nous entourent (lampe, climatiseur, chauffage ...) pour améliorer notre vie et la rendre plus confortable. Les chercheurs ont créé une nouvelle façon d'utiliser internet et d'interagir avec les objets, c'est le concept de « l'internet des objets » ou bien plus simplement en utilisant l'abréviation *IoT* pour « Internet Of Things » en anglais.

## 3. Concepts de base

Un écosystème *IoT* se compose de dispositifs *Web* compatibles intelligents qui utilisent des processeurs embarqués, des capteurs et du matériel de communication pour recueillir, envoyer et agir sur les données qu'ils acquièrent de leur environnement. Les dispositifs de l'*IoT* partagent les données des capteurs qu'ils collectent en se connectant à une passerelle ou un autre dispositif *IoT* de bord où les données sont soit transmises vers le *cloud* ou analysées localement. Parfois, ces appareils communiquent avec d'autres appareils connexes et agissent sur les informations qu'ils obtiennent les uns des autres. Les dispositifs font la plupart des travaux sans intervention humaine, bien que les gens puissent interagir avec les

dispositifs, par exemple, pour les mettre en place, leur donner des instructions ou avoir accès aux données. La connectivité, les réseaux et les protocoles de communication utilisés avec ces appareils *Web* compatibles dépendent en grande partie des applications spécifiques *IoT* déployées.

## Exemple d'un système IoT



Figure 1 Schéma montre le fonctionnement et l'interaction d'un système IoT [1]

## 4. Travaux liés à la détection des fuites de gaz

### 4.1. Détecteurs des fuites de gaz



Figure 2 Détecteur de gaz de ville et gaz en bouteille [2]

Il mesure la concentration de gaz naturel ou de ville (méthane), gaz en bouteille (propane), dans l'environnement. Si le niveau de gaz atteint le seuil nocif, une alarme visuelle et sonore se déclenche immédiatement.

Points forts	Points faibles
<ul style="list-style-type: none"> <li>• Capable de détecter les fuites de gaz dangereux et inflammables, comme le gaz de ville (méthane) ou le gaz en bouteille (propane).</li> <li>• Utilisation facile : il suffit simplement de l'installer auprès des appareils qui dégagent les gaz nocifs (méthane, propane) sans l'intervention d'une personne ayant des connaissances dans le domaine informatique.</li> <li>• Un prix abordable.</li> </ul>	<ul style="list-style-type: none"> <li>• Ne permet pas la détection de tous les gaz toxiques, tel que le monoxyde de carbone (<math>CO</math>), la fumée, le butane, le <i>GPL</i> (Gaz de pétrole liquéfié), etc.</li> <li>• N'est pas équipé d'une batterie, il doit être branché en permanence à une source d'énergie, par conséquent, lors d'une coupure d'électricité il ne fonctionnera plus.</li> <li>• Doté seulement d'une alarme sonore et visuelle, ceci peut être contraignant pour une personne âgée, malentendante, malvoyante ou même une personne en plein sommeil qui n'arrivera ni à entendre le son de l'alarme, ni à voir son indication.</li> <li>• N'affiche pas la concentration actuelle des différents gaz toxiques dans l'air, donc ne permet pas à une personne d'entreprendre les actions nécessaires (ouverture des fenêtres) avant que ces valeurs n'atteignent le seuil nocif.</li> <li>• L'alarme ne se déclenche que lorsqu'un des gaz atteint le seuil nocif.</li> </ul>

Tableau 1 Points forts et faibles du détecteur [2]



Figure 3 Détecteur de monoxyde de carbone CO [3]

Simple à installer et sans branchement électrique (alimentation par batterie), déclenche immédiatement une alerte sonore puissante avant même que la concentration de  $CO$  n'atteigne le seuil nocif.





Figure 4 Détecteur de monoxyde de carbone CO [4]

En plus des fonctionnalités contenues dans le détecteur précédent, celui-ci est doté d'un écran *LCD* qui affiche en temps réel en *PPM* (Partie Par Million) le taux de monoxyde de carbone (*CO*) capté dans l'air ambiant.

Points forts	Points faibles
<ul style="list-style-type: none"> <li>• Détecte un des gaz les plus mortels qui est le monoxyde de carbone (<i>CO</i>).</li> <li>• Alimentation par batterie, donc en cas de coupure d'électricité il fonctionnera toujours.</li> <li>• L'alarme se déclenche avant que le taux du <i>CO</i> dans l'air n'atteigne le seuil nocif, ce qui permet à une personne d'entreprendre les actions nécessaires.</li> <li>• Doté d'un écran <i>LCD</i> affichant en temps réel le taux du <i>CO</i> présent dans l'air, ceci permet à une personne d'avoir un pas d'avance avant qu'une situation dangereuse ne se produise.</li> <li>• Facile à utiliser.</li> <li>• Un prix abordable.</li> </ul>	<ul style="list-style-type: none"> <li>• Ne détecte pas tous les gaz nocifs, comme le gaz de ville (méthane) et le gaz en bouteille (propane).</li> <li>• Équipé seulement d'une alarme sonore, même si elle est puissante ça reste toujours une contrainte pour une personne malentendante ou si celle-ci se trouve en plein sommeil.</li> <li>• Pas de possibilité de recharger la batterie vu qu'il n'est pas branché à une source d'énergie, par conséquent, si la batterie est HS et sans que personne ne s'en rende compte, le détecteur n'accomplit pas sa tâche.</li> </ul>

Tableau 2 Points forts et faibles des détecteurs [3] [4]

## 4.2. Système de détection de fuites de gaz basé sur GSM

Les auteurs de [5] présentent un système permettant de détecter les fuites de gaz et de les arrêter en coupant l'alimentation principale de gaz, il désactive aussi l'alimentation électrique et avertit les occupants du domicile via une alarme sonore ainsi que les différents services de secours en leurs envoyant un *SMS* ou un message vocal.

### 4.3. Systèmes de détection de fuites de gaz basés sur l'IoT

Les auteurs de [6] proposent un système de détection de fuites de gaz basé sur la technologie *IoT*, ayant des techniques d'alerte intelligente à savoir l'appel et l'envoi de *SMS* en utilisant un module *GSM* et *Email* aux services de secours ainsi qu'aux occupants du domicile. Il a aussi la capacité de prévenir une situation dangereuse afin que les personnes puissent être informées à l'avance, en effectuant le *Data Analytics* sur les lectures des capteurs.

Le système utilise un capteur dédié au contrôle de l'état des gaz toxiques qui peuvent être présents dans l'air ambiant, comme : l'hydrogène (*H<sub>2</sub>*), Gaz de Pétrole Liquéfié (*GPL*), le méthane (*CH<sub>4</sub>*), le monoxyde de carbone (*CO*), l'alcool, la fumée et le propane.

Lorsque le système constate la présence de quantités excessives de gaz nocifs il :

- Alerte les locataires présents au domicile via une alarme sonore ainsi que celles absentes et les services de secours via un Appel/*SMS* sur la situation avant qu'un accident ne se produise, un *Email* sur les détails de la zone est aussi envoyé,
- Coupe l'alimentation électrique principale de la maison lorsque la concentration de gaz inflammables est sur le point d'atteindre sa limite inférieure d'explosivité (*LIE*).

Les auteurs de [7] ont conçu un système permettant de détecter les fuites de bouteilles de gaz. Lors de détection d'une fuite il :

- Avertit les locataires du domicile ainsi que le voisinage via une alarme sonore,
- Ferme mécaniquement le robinet du régulateur pour saisir toute fuite de gaz,
- Coupe l'alimentation électrique principale de la maison et active un ventilateur d'extraction qui expulsera tout le gaz contenu dans l'environnement,
- Envoi un *SMS* et *Email* aux personnes concernées.

Les auteurs de [8] proposent un système permettant de surveiller en permanence une éventuelle fuite de gaz inflammables ou toxiques. En cas de détection de fuite il :

- Alerte les habitants du domicile via une alarme sonore et une notification qu'ils vont recevoir dans l'application mobile,
- Ferme la vanne de gaz et active le ventilateur d'extraction.

Le système alertera aussi sur les conditions environnementales comme le niveau de gaz et la température de cet emplacement d'installation en utilisant les réseaux sociaux comme *Twitter* ou via une notification par *Email*.

L'utilisateur peut aussi intervenir manuellement sur les appareils connectés au système comme un ventilateur d'extraction ou un moteur pour fermer la vanne de gaz à l'aide d'une application mobile.

Les auteurs de [9] proposent un système permettant non seulement de détecter les fuites de gaz mais aussi de faire face au problème du gaspillage de gaz en utilisant un capteur infrarouge. Une alarme se déclenche à chaque fois que le capteur ne détecte aucun récipient sur le brûleur au-delà d'une certaine période de temps.

Lorsque le système détecte une fuite de gaz il avertit la personne concernée via une notification qu'elle va recevoir dans l'application mobile, et en utilisant cette dernière elle peut fermer la vanne de gaz indépendamment de l'endroit où la personne se trouve.

Points forts	Points faibles
<ul style="list-style-type: none"> <li>• Coupent l'alimentation électrique principale, ce qui permet d'éviter une explosion dévastatrice due au contact électricité-gaz. [5] [6] [7]</li> <li>• Ferment la vanne principale de gaz/le robinet du régulateur d'une bouteille de gaz, ce qui permet d'arrêter la propagation massive du gaz dans l'air, et ainsi augmenter la chance de survie des personnes présentes au domicile à une asphyxie qui peut être mortelle. [5] [7] [8]</li> <li>• Avertissent les services de secours et les occupants absents du domicile de la situation, ceci est très utile dans le cas où les personnes présentes au domicile n'entendent pas l'alarme sonore ou elles sont en plein sommeil ou encore elles sont inconscientes suite à l'émanation du gaz. [5] [6] [7]</li> <li>• Enregistre les lectures (concentrations de gaz) des capteurs dans une base de données. Ces données pourraient être utilisées afin d'effectuer des analyses permettant de comprendre les conditions habituelles de la zone, dans quelles conditions les lectures des capteurs augmentent et quand commencent-ils à devenir dangereux</li> </ul>	<ul style="list-style-type: none"> <li>• Ne prennent pas en charge d'autres actions automatiques telles que l'activation de la ventilation ou l'ouverture des fenêtres, qui permettent d'évacuer les gaz nocifs, ce sont là des actions qui peuvent s'avérer très utiles si les services de secours ainsi que les occupants absents du domicile tardent à arriver sur le lieu. [5] [6]</li> <li>• Il n'est pas connecté à Internet, donc ne permet pas aux habitants du domicile de suivre en temps réel l'état des différents gaz toxiques pouvant être présents dans l'environnement, et ainsi de prendre les actions nécessaires lorsqu'une anomalie est aperçue. [5]</li> <li>• En cas d'une fuite de gaz, il ne ferme pas la vanne principale de gaz/le robinet du régulateur d'une bouteille de gaz, donc le taux du gaz nocif augmentera en permanence, mettant les personnes présentes au domicile face au danger d'une asphyxie mortelle. [6]</li> <li>• Contrairement aux détecteurs traditionnels, l'utilisation de ces systèmes n'est pas facile car leurs installations nécessitent l'intervention d'une personne ayant des</li> </ul>

<p>nécessitant ainsi plus d'attention. Ceci augmentera la précision du système, réduisant les fausses alarmes et devenant ainsi un système très fiable. [6]</p> <ul style="list-style-type: none"> <li>• Si l'utilisateur oublie accidentellement d'éteindre le brûleur à gaz, ces systèmes l'informent en activant une alarme, donc le problème de gaspillage d'énergie est résolu. [7] [9]</li> <li>• Il surveille en permanence le niveau de gaz présent dans la bouteille en utilisant un capteur dédié, si le niveau de gaz atteint le seuil minimum, il envoie automatiquement une notification à l'agence de gaz afin que l'utilisateur puisse réserver une nouvelle bouteille de gaz. [7]</li> <li>• Ils activent un ventilateur d'extraction qui expulsera tout le gaz toxique contenu dans l'air ambiant. [7] [8]</li> <li>• Permettent le suivi en temps réel des conditions environnementales comme le niveau de gaz, ainsi, lorsqu'une anomalie est constatée par la personne, elle peut intervenir manuellement sur les appareils connectés au système comme un ventilateur d'extraction ou un moteur pour fermer la vanne de gaz. [8] [9]</li> <li>• Par rapport aux détecteurs traditionnels, ils sont équipés d'un capteur capable de détecter plusieurs gaz comme l'hydrogène (<math>H_2</math>), Gaz de Pétrole Liquéfié (<math>GPL</math>), le méthane (<math>CH_4</math>), le monoxyde de carbone (<math>CO</math>), l'alcool, la fumée et le propane. [5] [6] [7] [8] [9]</li> </ul>	<p>connaissances dans le domaine informatique. [5] [6] [7] [8] [9]</p> <ul style="list-style-type: none"> <li>• Le prix de ces systèmes reste inaccessible pour certaines catégories de personnes. [5] [6] [7] [8] [9]</li> </ul>
--	---

Tableau 3 Points forts et faibles des systèmes [5] [6] [7] [8] [9]

## 5. Conclusion

D'après l'étude que nous avons effectuée, les détecteurs des fuites de gaz traditionnels existant sur le marché restent inefficaces, car ils n'ont pas la capacité de détecter tous les gaz toxiques qui peuvent être présents dans l'environnement. Mais aussi ils se contentent simplement d'avertir les personnes présentes à domicile d'une anomalie via une alarme sonore et/ou visuelle qui est insuffisant lorsque la personne est malentendante, malvoyante, elle est en plein sommeil ou encore elle est inconsciente suite à l'émanation du gaz. Donc, pour combler cette insuffisance de ces détecteurs, l'élaboration des systèmes intelligents basés sur l'*IoT* étaient une nécessité, car ils permettraient non seulement de détecter les fuites de gaz mais aussi d'intervenir de manière automatique (fermeture de la vanne principale de gaz/le robinet du régulateur d'une bouteille de gaz, couper l'alimentation électrique principale, activation d'une ventilation, etc.) pour éviter qu'une situation dangereuse ne se produise. Par ailleurs, ces systèmes ne sont pas capables de détecter des situations de détresse telles que les chutes et les problèmes respiratoires qui peuvent aussi causer la mort des personnes à domicile si elles ne sont pas secourues au bon moment.

Donc, dans le but de concevoir un système complémentaire, nous étudierons dans le prochain chapitre les différents capteurs et systèmes intelligents permettant de détecter et d'avertir les personnes concernées lors d'une situation de détresse.

## **Chapitre 2 : Détection de personnes en situation de détresse**

## 1. Introduction

La détection de personnes en situation de détresse en utilisant le concept de *Machine Learning* et les réseaux de neurones pour traiter les données collectées par des capteurs, représente un vrai défi, car dans un premier temps on doit déterminer les situations de détresse que nous pouvons détecter et comment, et ce, pour les traiter avec les algorithmes de *ML (Machine Learning)*. Le choix du matériel le mieux adapté à notre projet pour capturer les différents signes de détresse est une autre partie du challenge que l'on doit relever, tout ça doit interagir avec un système connecté en temps réel pour sauver des vies humaines en temps opportun.

La section suivante est consacrée à la présentation des types de détresses et comment on peut les détecter afin de compléter notre étude théorique. Nous présentons en premier lieu la détection des chutes et les systèmes existant dans ce cadre, nous clôturons ce chapitre par étudiant d'autres situations de détresse qui peuvent se produire suite à une fuite de gaz.

## 2. Détection des chutes

### 2.1. Introduction

Face à l'augmentation de la population de seniors dans le monde, de nouveaux systèmes de santé doivent être mis en place afin d'assurer la sécurité des personnes âgées surtout celles qui vivent seules. Les chutes sont l'un des principaux risques pour ces personnes, provoquant souvent des blessures graves. La gravité de la situation peut augmenter si la personne ne peut pas appeler à l'aide, étant inconsciente ou immobilisée. Pour surmonter ces problèmes, plusieurs chercheurs ont élaboré des systèmes de détection de chutes, utilisant divers approches et technologies.

### 2.2. Basés sur les capteurs

Les auteurs de [10] proposent un système de capteurs de détection de chutes basé sur un accéléromètre. Ils utilisent un nœud *TelosW* (carte électronique) avec un accéléromètre comme détecteur, le système peut différencier les chutes des activités normales en se basant sur un seuil qui n'est pas défini d'une manière empirique, minimisant ainsi les fausses alarmes. Si une chute est détectée, un message d'alerte est généré puis envoyé aux contacts prédéterminés. La carte est attachée à la taille de la personne pour capturer les données de mouvement. Comment le système fonctionne-t-il ?

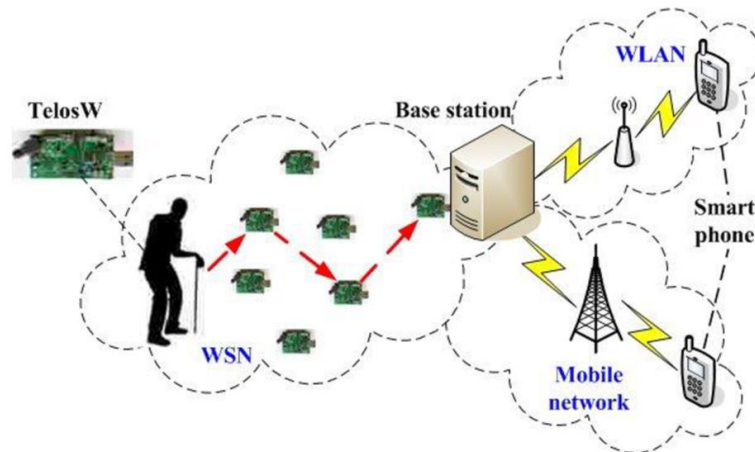


Figure 5 Architecture du système de capteurs de détection de chute [10]

D'abord, les données d'accélération sont échantillonnées pour être traitées localement au nœud du capteur ou transmises à la station de base, où un ordinateur puissant peut effectuer une analyse plus sophistiquée comme la reconnaissance des formes, etc. Cependant, en utilisant cette dernière méthode, les données échantillonnées doivent être livrées en continu à la station de base, ce qui impose une grande charge sur la communication et n'est pas économe en énergie. Donc, il est plus économique de traiter les données localement au nœud du capteur. Dès qu'une chute est détectée, seul le message d'alerte est transmis à la station de base.

L'accéléromètre fournit des lectures numériques qui sont relatives à l'accélération dans les directions des axes  $x$ ,  $y$  et  $z$ . Elles sont représentées respectivement par les vecteurs  $A_x$ ,  $A_y$  et  $A_z$  : soit  $A_c$  la composition des accélérations dans les trois directions, dont l'amplitude peut être calculée par l'équation :

$$|A_c| = \sqrt{|A_x|^2 + |A_y|^2 + |A_z|^2}.$$

L'algorithme de détection de chutes se base principalement sur cette valeur calculée  $|A_c|$ . Si la valeur  $|A_c|$  dépasse un certain seuil configuré, donc une chute est détectée.

Pour distinguer les chutes des autres activités normales, les auteurs ont identifié la caractéristique de chaque activité (le seuil maximal d'accélération de chaque activité). Plusieurs expériences de test ont été menées en utilisant *TelosW* pour la collecte de données d'accélération. Pour simplifier la communication et se concentrer sur la méthode de détection, les auteurs de [10] ont uniquement construit un réseau à un saut, ce qui signifie que le nœud d'échantillonnage transmet les données d'accélération directement à la station de base.





Figure 6 Photos de l'expérience [10]

Avec quelques accélérations typiques observées par différentes activités (marcher, s'asseoir, s'accroupir, tomber) et différents taux d'échantillonnage, ils ont obtenu les résultats suivants :

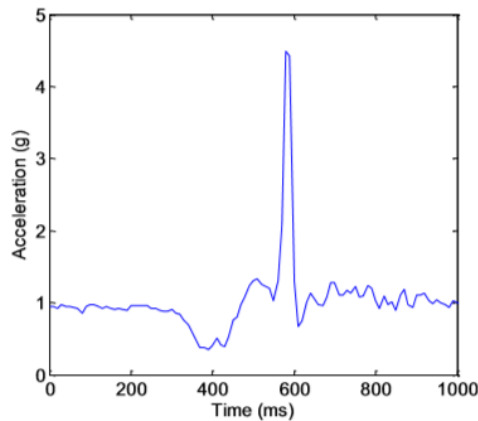


Figure 7 Chute, taux d'échantillonnage 100Hz [10]

À  $t = 580$  ms sur la figure 7, on constate un grand pic, indiquant l'impact. Après le pic de l'impact, la valeur retombe à environ 1g car celui-ci a été absorbé par le sol et le corps de la personne a atterri sur le sol.

Le taux d'échantillonnage constitue un facteur critique pour capturer la valeur de crête car, comme le montrent les figures 8 et 9, les valeurs de crête ne sont pas si évidentes lorsque les taux d'échantillonnage diminuent.

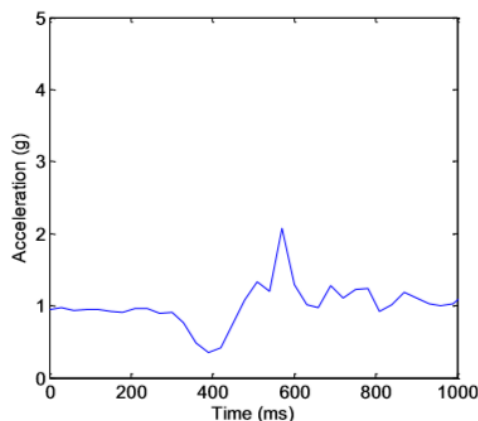


Figure 8 Chute, taux d'échantillonnage 33Hz [10]

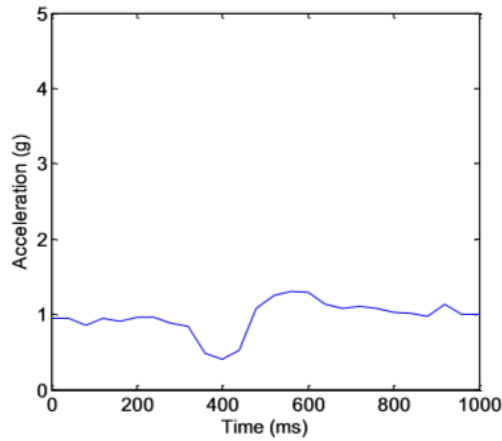


Figure 9 Chute, taux d'échantillonnage 25Hz [10]

Les profils d'accélération lorsqu'une personne s'assoit et s'accroupie sont similaires à ceux de la chute, cependant, leurs valeurs maximales sont beaucoup plus petites, ce qui peut être considéré comme la différence la plus notable. Le profil d'accélération de la marche est très différent de celui des autres activités, comme le montre la figure 12. Les valeurs d'accélération n'ont que de petites fluctuations autour de 1g.

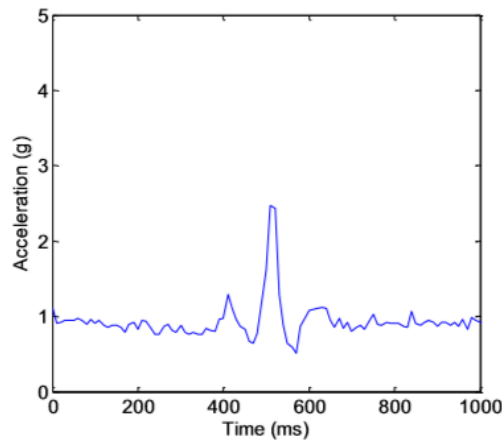


Figure 10 S'asseoir, taux d'échantillonnage 100Hz [10]

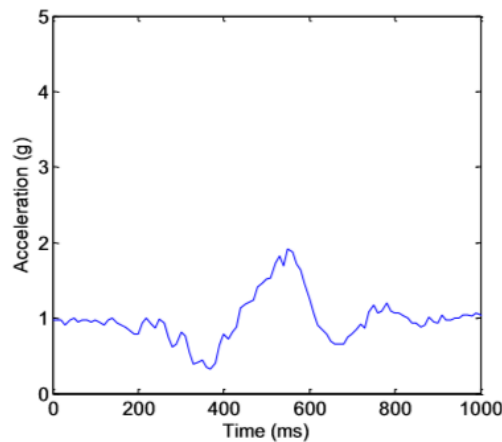


Figure 11 S'accroupir, taux d'échantillonnage 100Hz [10]

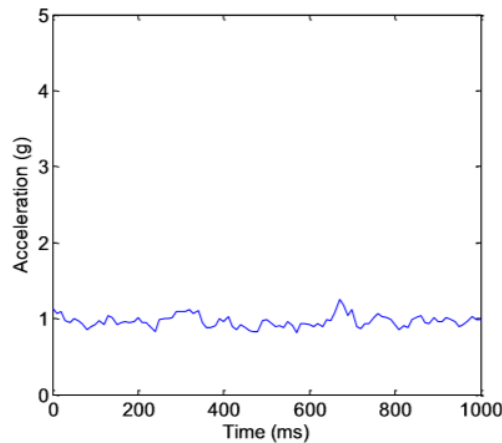


Figure 12 Marcher, taux d'échantillonnage 100Hz [10]

Pour obtenir le meilleur seuil d'accélération permettant de détecter les chutes, mais aussi de minimiser les fausses alarmes et maximiser le taux de détection du système, les auteurs ont utilisés le modèle *Neyman-Pearson*.

Les auteurs de [11] considèrent que les détecteurs de chutes basés sur des accéléromètres sont intrusifs et nécessitent que l'utilisateur les portes en permanence et les active, donc ils peuvent échouer en cas de non-conformité de l'utilisateur. Pour cela ils ont conçu un détecteur de chute basé sur les vibrations du sol qui est complètement passif et discret pour le résident. Comment fonctionne le système ?

Les auteurs ont observé que les activités humaines, comme la marche et la course, peuvent provoquer des vibrations mesurables au sol. Il est tout à fait clair que les chutes humaines provoqueront également des vibrations sur le sol. Lorsqu'un humain tombe, les impacts d'une partie du corps avec le sol génèrent des vibrations qui sont transmises à travers le sol. Le principe de fonctionnement du détecteur de chute basé sur les vibrations du sol est fondé sur l'hypothèse qu'il est possible de détecter les chutes humaines en surveillant les modèles de vibration dans le sol. Cette hypothèse implique que :

- Les vibrations du sol générées par la chute d'un être humain est significativement différente de celles générées par les activités quotidiennes normales comme la marche, les tapotements, etc.
- Les vibrations du sol générées par une chute humaine est significativement différente de celle générée par les objets tombant sur le sol.



Piezo Transducer

Figure 13 Détecteur de chute basé sur les vibrations du sol [11]

Le détecteur de chute basé sur les vibrations du sol, illustré dans la figure 13, utilise un capteur piézoélectrique (*piezoelectric sensor*) également connu sous le nom de transducteur piézoélectrique (*piezoelectric transducer*) qui est un appareil utilisant l'effet piézoélectrique pour mesurer les changements de pression, d'accélération, de température, de déformation ou de force en les convertissant en une charge électrique. Ce détecteur est couplé à la surface du sol au moyen d'un agencement de masse et de ressort, combiné à une électronique de prétraitement alimentée par batterie pour évaluer les modèles de vibration du sol et générer un signal de chute binaire, et un émetteur sans fil qui relaie l'alarme de chute à une passerelle de communication.

Sur la base des expériences menées, ils ont observé qu'il y avait des différences significatives dans les modèles de vibrations induites au sol par différentes activités. Il y avait également une différence significative dans le modèle de vibration généré par la chute d'un objet par rapport à celui généré par une chute d'un humain. Cette différence de réponse du sol à différentes activités a été efficacement exploitée pour détecter les chutes avec un degré de sensibilité et de spécificité plus élevé (moins de fausses alarmes). L'appareil ne détecte une chute que lorsque le modèle de vibration obtenu, à partir du sol sur une courte période, correspond au modèle induit lorsqu'une personne tombe sur le sol. Le détecteur peut ensuite signaler l'alerte de chute au répondeur via un portail de communication approprié, telle que l'utilisation du téléphone pour envoyer un message à un téléavertisseur radio ou à un téléphone portable.

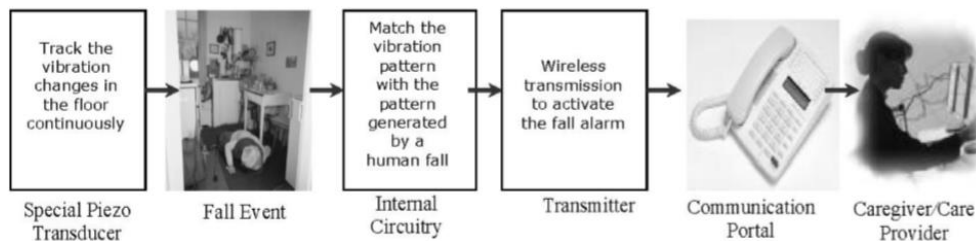


Figure 14 Schéma du principe de fonctionnement du détecteur de chute basé sur les vibrations du sol [11]

Points forts	Points faibles
<ul style="list-style-type: none"> <li>Assurent la confidentialité des résidents. [10] [11]</li> <li>Capables de distinguer entre les activités normales (s'asseoir, s'accroupir, etc.) et les chutes réelles. [10] [11]</li> <li>Lors de la chute d'une personne, ils alertent les services de secours ainsi que les proches de la situation. [10] [11]</li> <li>Mise en place facile. [11]</li> </ul>	<ul style="list-style-type: none"> <li>Les détecteurs de chutes portables comme les accéléromètres ou les boutons d'aide doivent être portés en permanence par la personne. Si celle-ci oublie de les porter, et par la suite elle tombe, donc la chute n'est pas détectée ce qui ne déclenchera pas l'alerte pour secourir la personne. De plus, dans le cas d'un bouton d'aide, cela ne sert</li> </ul>

	<p>à rien si la personne est inconsciente après la chute. [10]</p> <ul style="list-style-type: none"> <li>• Fonctionnent uniquement sur batterie, ils doivent être remplacés ou rechargés régulièrement pour un bon fonctionnement. Supposons que la batterie est HS et que la personne ne s'est pas rendue compte, alors si elle tombe à cet instant, la chute n'est pas détectée pour déclencher une alerte. [10] [11]</li> <li>• Il dépend de la dynamique du sol parquet, béton, carrelage, etc. [11]</li> <li>• Cette approche ne peut pas fournir une précision parfaite car un bruit excessif est présent dans un environnement de vie normal ce qui augmente la possibilité de fausses alarmes (une chute dans un appartement voisin). [11]</li> </ul>
--	--

Tableau 4 Points forts et faibles des systèmes [10] [11]

### 2.3. Basés sur la vidéo

Dans l'article [12], les auteurs proposent une nouvelle méthode pour détecter les chutes en analysant le changement du mouvement de la personne au cours d'une séquence vidéo. Ils concluent qu'une chute se caractérise par un mouvement important suivi de quelques changements dans la forme humaine. Plus précisément, au cours des activités habituelles, la forme humaine changera progressivement et lentement, tandis que lors d'une chute, la forme humaine changera radicalement et rapidement. Ainsi, ils ont choisi de détecter les chutes au cours d'une séquence vidéo en quantifiant la déformation de la forme humaine. Les principales étapes du système sont les suivantes :

- Obtenir seulement la silhouette de la personne par une méthode de segmentation de premier plan (*foreground segmentation method*),
- Extraire certains points de contour de la silhouette,
- Pour savoir si un changement de forme corporelle a eu lieu, il faut faire une correspondance des points de deux silhouettes consécutives. On obtient alors, une valeur de mesure de déformation 'le coût de correspondance moyen (*mean matching cost*)',
- Une autre technique est aussi exécutée, l'analyse procustéenne (*Procrustes analysis*) qui permet de comparer des formes, suite à cette technique on obtient une autre mesure de déformation 'la

distance complète de procrustes (*Full Procrustes Distance*), qui va permettre de quantifier la déformation entre les deux formes. Ces valeurs devraient être élevées pendant une chute et faibles juste après la chute,

- Enfin, le modèle de mélange gaussien (*Gaussian Mixture Model (GMM)*) est utilisé pour classer les différentes activités comme étant une chute ou non, basé sur une déformation de forme pendant la chute suivie d'un manque de mouvement significatif après la chute.

Les auteurs de [13] présentent une autre approche permettant de reconnaître les chutes en utilisant un cadre de délimitation de la silhouette (*bounding box*). Les principales étapes de leur approche sont les suivantes :

- La segmentation de l'individu de son arrière-plan qui aboutit à une silhouette indiquant uniquement la position du corps de l'individu dans une image, cette silhouette va être suivie en permanence durant le temps,
- Une fois la silhouette acquise, le rapport largeur sur hauteur du cadre de délimitation (*bounding box*) de la silhouette est ainsi calculé. Si la silhouette est debout, le rapport largeur sur hauteur est faible, par contre, lorsque le sujet est au sol, le rapport largeur sur hauteur devient beaucoup plus grand.

Les auteurs de [14] proposent une nouvelle méthode pour détecter les chutes en se basant sur la trajectoire *3D* de la tête, ceci leur a permis de distinguer les chutes des activités normales en utilisant la vitesse *3D*. Par rapport aux auteurs de [15], leur système est basé sur une estimation de vitesse verticale *2D* qui peut être insuffisante pour discriminer une chute réelle d'une personne qui s'assoie brutalement.

Les principales étapes de cette méthode sont :

- Suivi de la tête de la personne, car, elle est généralement visible dans la scène (pas de problème d'occlusion), et a un grand mouvement lors d'une chute,
- Extraction des informations *3D* concernant la trajectoire de la tête, ensuite, il calcule les caractéristiques de vitesse (vitesse verticale  $V_v$  et vitesse horizontale  $V_h$ ), qui sont utiles pour la détection des chutes,
- Il peut différencier une chute des activités normales en utilisant deux seuils appropriés pour la vitesse verticale  $V_v$  et vitesse horizontale  $V_h$ ,
- Il reconnaît une chute si un pic se produit pratiquement en même temps pour  $V_v$  et  $V_h$ , et si le pic négatif  $V_v$  est inférieur à  $-1.5$  m/s et le pic positif  $V_h$  est supérieur à  $2$  m/s.

Points forts	Points faibles
<ul style="list-style-type: none"> <li>• La personne n'a pas besoin de porter d'appareil sur son corps. [12] [13] [14]</li> </ul>	<ul style="list-style-type: none"> <li>• N'est efficace que si la caméra est placée sur le côté et peut échouer en raison de l'occlusion d'objets. [13]</li> </ul>

<ul style="list-style-type: none"> <li>• Assurent la confidentialité des résidents par l'extraction de la silhouette de l'arrière-plan. [12] [13]</li> <li>• Capables de distinguer les activités normales (marcher, s'asseoir brutalement ou s'agenouiller) des chutes réelles. [12] [13] [14]</li> <li>• Si une situation dangereuse est détectée, ils envoient un signal d'alarme vers une entité extérieure (via un appel ou Internet). [12] [14]</li> <li>• Robuste à la plupart des difficultés de traitement d'image, car il prend en compte un ensemble de données vidéo réalistes contenant des occlusions, le transport d'objets, le changement de vêtements et différents points de vue. [12]</li> <li>• Il détecte la chute pendant qu'elle se produit plutôt qu'après. [14]</li> <li>• Ces systèmes sont alimentés par des prises murales électriques avec éventuellement une alimentation de secours (batterie). [12] [13] [14]</li> </ul>	<ul style="list-style-type: none"> <li>• La détection de la tête d'une personne ne se fait pas d'une manière automatique. [14]</li> <li>• La détection des chutes en se basant sur la vidéo a des limites lorsqu'il y a des angles morts et des obstructions. [12] [13] [14]</li> </ul>
--	---

Tableau 5 Points forts et faibles des systèmes [12] [13] [14]

### 3. Autres situations de détresse

#### 3.1. Situation de panique

Une attaque de panique est caractérisée par une forte crise d'angoisse associée à une peur extrême de la mort qui conduit certaines personnes à perdre le contrôle d'eux-mêmes et à commettre des actes anormaux, par exemple : se jeter par la fenêtre, se rouler par terre en hurlant, provoquer un accident, etc. On peut distinguer 4 types de symptômes : psychiques, psychomoteurs, physiques et neurovégétatifs. Ils sont souvent intenses mais pas toujours objectifs. Le type de symptôme qui nous intéresse est le physique, qui se manifeste comme suit : palpitations cardiaques, tremblements, difficultés respiratoires, sensation d'étouffement, sensation de vertiges, sensation de malaise, douleurs thoraciques, diarrhée.

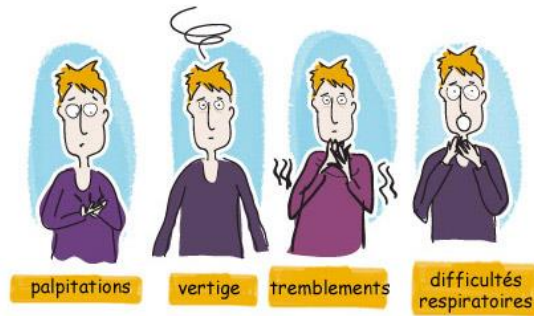


Figure 15 illustration des signes de panique physique [16]

La panique peut s'avérer invraisemblablement utile du fait que les signes de panique permettront de détecter et d'anticiper le sauvetage d'une personne en danger augmentant ainsi ses chances de survie.

### 3.2. Détresse respiratoire

La fréquence cardiaque est un signe vital essentiel pour le diagnostic médical. Il existe un intérêt croissant pour son extraction sans contact, en particulier pour les populations telles que les nouveaux nés prématurés et les personnes âgées pour qui la peau est fragile et endommageable par les capteurs traditionnels. En outre, à mesure que la population vieillit, une surveillance continue ou au moins fréquente en dehors des environnements cliniques peut fournir aux médecins non seulement des échantillons opportuns, mais aussi des tendances à long terme et des analyses statistiques.

Dans l'article [17], les auteurs exploitent de subtiles oscillations de la tête qui accompagnent le cycle cardiaque pour extraire des informations sur l'activité cardiaque à partir de vidéos. En plus de fournir un moyen discret pour mesurer la fréquence cardiaque, la méthode peut être utilisée pour extraire d'autres informations cliniquement utiles sur l'activité cardiaque, tels que les changements subtils de la longueur des battements cardiaques associés à la santé du système nerveux autonome.

Le mouvement cyclique du sang du cœur vers la tête via l'aorte abdominale et les artères carotides (Figure 16) provoque un mouvement périodique de la tête. Leur algorithme détecte l'impulsion à partir de ce mouvement.

Il nous paraît possible d'exploiter leur méthode pour déterminer si une personne souffre de problèmes respiratoires, car lorsque le rythme cardiaque augmente lors des palpitations, de fortes oscillations de la tête sont constatées par rapport à la normale.

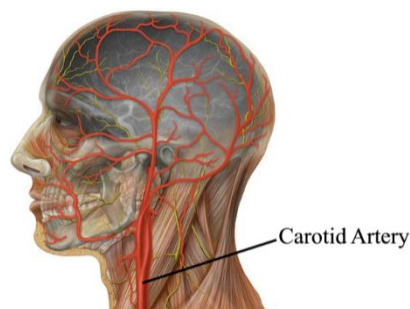


Figure 16 Le sang coule du cœur vers la tête via les artères carotides de chaque côté de la tête [17]



### 3.3. Détresse cardiaque

Le syndrome de détresse respiratoire est une forme sévère de défaillance pulmonaire, qui peut avoir de nombreuses causes (œdème pulmonaire, noyade, inhalation de fumée, crise d'asthme, etc.). Il se traduit par une impossibilité passagère de respirer normalement, lorsque les échanges gazeux de l'expiration et de l'inspiration ne sont plus efficaces. Ce phénomène entraîne un défaut d'oxygénation, dont les conséquences peuvent être fatales. Ce défaut d'oxygénation peut être lié à des facteurs extérieurs (manque d'oxygène dans l'air) ou internes (obstruction pulmonaire due à un aliment, maladie chronique, etc.). Globalement, le taux de mortalité relatif au syndrome de détresse respiratoire est d'environ 40%.

Une respiration irrégulière et bruyante, un pouls accéléré, des maux de têtes, une sensation d'angoisse, des lèvres et des extrémités bleutées (cyanose due à un manque d'oxygène dans le sang), des troubles de la conscience et du comportement, sont tous des symptômes de détresse respiratoire.

Si on utilise un capteur vidéo pour intercepter ces symptômes, on doit s'intéresser alors à l'ouverture buccale qui va être grande ouverte en raison de manque d'oxygène dans les poumons, ainsi que le changement de couleur des lèvres et le mouvement des épaules.



Figure 17 Illustration d'une difficulté respiratoire [18]

## 4. Conclusion

Dans ce chapitre, nous avons commencé par étudier les différents capteurs et détecteurs de chutes tels que : l'accéléromètre, le bouton d'aide et le détecteur basé sur les vibrations du sol, on a constaté qu'ils sont inefficaces et présentent des points faibles, par exemple, dans le cas de l'accéléromètre et du bouton d'aide, ils doivent être portés en permanence par la personne. Si celle-ci oublie de les porter, et que par la suite elle tombe, sa chute ne sera pas détectée ce qui ne déclenchera pas l'alerte pour la secourir. De plus, le bouton d'aide ne sert plus à rien si la personne est inconsciente après la chute. Le détecteur des chutes basé sur les vibrations du sol dépend principalement de la dynamique du sol et peut aussi confondre d'autres bruits présents dans l'environnement avec celui émis par une chute, ce qui augmente les fausses alarmes.

Face, à ces inconvénients, nous nous sommes intéressés à d'autres systèmes qui se basent sur la vidéo pour détecter les chutes, par exemple : à l'aide du rapport largeur sur hauteur du cadre (*bounding box*) délimitant la silhouette d'une personne ou en calculant la vitesse horizontale et verticale de déplacement

de la tête. Cependant, même ces systèmes présentent des points faibles, pour le premier système, lorsque la personne est allongée avec les mains et les pieds collés au corps, le rapport largeur sur hauteur est similaire à celui lorsque la personne est debout. Quand la personne est assise, le rapport largeur sur hauteur est similaire à celui de la personne qui est allongée dans une forme assise. Donc cette méthode n'est pas fiable pour reconnaître les postures et par conséquent, même les chutes. La vitesse horizontale et verticale d'une personne qui s'assoit brusquement sur une chaise ou qui s'allonge brusquement sur un lit, peut être similaire à celle d'une chute, ce qui va augmenter les fausses alarmes.

Face à ces points faibles que nous avons remarqués dans ces systèmes, nous pouvons conclure qu'une autre approche se basant principalement sur le concept du *Machine Learning* et sur la vision par ordinateur doit être élaborée afin d'avoir un système plus fiable et robuste pour détecter les chutes.

# **Chapitre 3 : Machine Learning**

## 1. Introduction

Pourquoi ne pas essayer de construire des cerveaux artificiels en copiant comment les vrais cerveaux biologiques fonctionnent ? De vrais cerveaux avec des neurones au lieu de portes logiques, un raisonnement plus doux et plus organique au lieu d'algorithmes traditionnels absolus et durs.

Les scientifiques se sont inspirés de l'apparente simplicité du cerveau d'une abeille ou d'un pigeon par rapport aux tâches complexes qu'ils pouvaient accomplir. Les cerveaux d'une fraction de gramme semblaient capables de faire des choses comme diriger le vol, s'adapter au vent, identifier la nourriture et les prédateurs et décider rapidement de se battre ou de s'échapper.

Aujourd'hui, les ordinateurs sont dotés de ressources massives et bon marché, mais, pourraient-ils imiter et améliorer ces cerveaux ? Une abeille possède environ 950 000 neurones, les ordinateurs d'aujourd'hui avec des gigaoctets et des téraoctets de ressources pourraient-ils surpasser les abeilles ? La réponse est non, avec des approches traditionnelles pour résoudre les problèmes, ces ordinateurs avec un stockage massif et des processeurs ultra-rapides ne pouvaient pas réaliser ce que les cerveaux relativement minuscules des oiseaux et des abeilles pouvaient faire. Les réseaux de neurones ont émergé de cette tendance à l'informatique intelligente d'inspiration biologique, et sont devenus l'un des concepts les plus puissants et les plus utiles dans le domaine de l'intelligence artificielle.

## 2. Définition

L'apprentissage automatique (*Machine Learning*) est une application de l'intelligence artificielle (IA), qui offre aux systèmes la capacité d'apprendre et de s'améliorer automatiquement à partir de l'expérience sans être explicitement programmé. L'apprentissage automatique se concentre sur le développement de programmes informatiques qui peuvent accéder aux données et les utiliser pour apprendre par eux-mêmes.

Le processus d'apprentissage commence par des observations ou des données, tels que des exemples, une expérience directe ou des instructions, afin de rechercher des modèles dans les données et de prendre de meilleures décisions à l'avenir en fonction des exemples que nous fournissons. L'objectif principal est de permettre aux ordinateurs d'apprendre automatiquement sans intervention ou assistance humaine et d'ajuster les actions en conséquence.

## 3. Types de Machine Learning

### 3.1. Apprentissage supervisé

Un algorithme d'apprentissage supervisé, apprend à partir d'exemples de données et de réponses cibles associées, qui peuvent consister en des valeurs numériques ou des étiquettes de chaîne, telles que des classes ou des balises. Après la phase d'apprentissage, il est évalué sur de nouveaux exemples, où il va prédire ou non la bonne réponse. Cette approche est en effet similaire à l'apprentissage humain sous la

supervision d'un enseignant. L'enseignant fournit de bons exemples à mémoriser, et l'élève tire ensuite des enseignements généraux de ces exemples spécifiques.

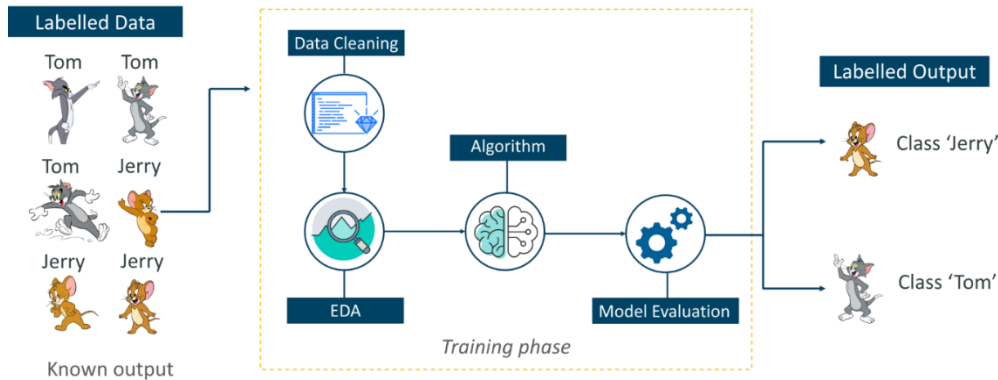


Figure 18 Exemple d'apprentissage supervisé [19]

### 3.2. Apprentissage non supervisé

Dans l'apprentissage non supervisé, l'algorithme est fourni avec un ensemble de données non étiqueté. Il apprend à partir d'exemples simples sans réponse associée, laissant à l'algorithme le soin de déterminer lui-même les modèles de données. Ce type d'algorithme a tendance à restructurer les données en quelque chose d'autre, comme de nouvelles fonctionnalités qui peuvent représenter une classe ou une nouvelle série de valeurs non corrélées. Il est très utile pour fournir aux humains un aperçu de la signification des données et de nouvelles entrées utiles pour les algorithmes d'apprentissage automatique supervisé.

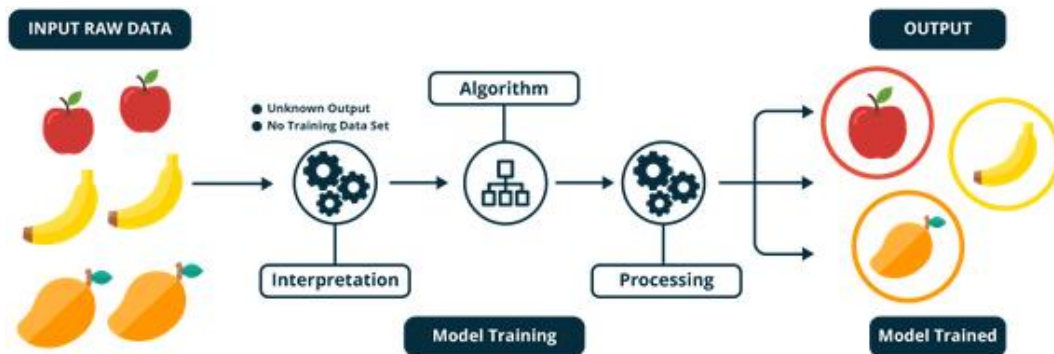


Figure 19 Exemple d'apprentissage non supervisé [19]

### 3.3. Apprentissage par renforcement

L'apprentissage par renforcement consiste à présenter l'algorithme avec des exemples sans étiquettes, comme dans l'apprentissage non supervisé, cependant, on peut accompagner un exemple de rétroaction positive ou négative selon la solution proposée par l'algorithme, qui est liée aux applications pour lesquelles l'algorithme doit prendre des décisions (le produit est donc prescriptif, pas seulement descriptif, comme dans l'apprentissage non supervisé), et les décisions ont des conséquences. Dans le monde humain, c'est comme apprendre par essais et erreurs. Les erreurs nous aident à apprendre car elles ont une pénalité supplémentaire (coût, perte de temps, regret, douleur, etc.), elles nous enseignent qu'un certain plan d'action a moins de chances de réussir que d'autres. Un exemple intéressant

d'apprentissage par renforcement se produit lorsque les ordinateurs apprennent à jouer seuls à des jeux vidéo.

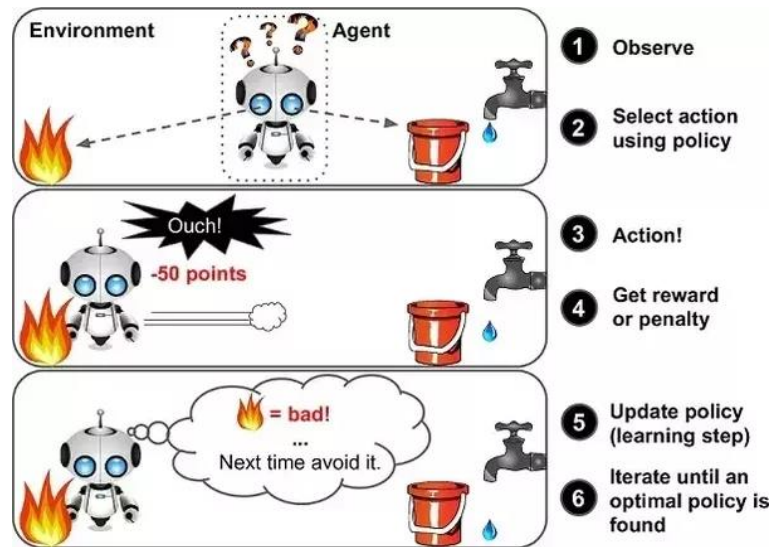


Figure 20 Exemple d'apprentissage par renforcement [19]

## 4. Réseaux de neurones

### 4.1. Le neurone biologique

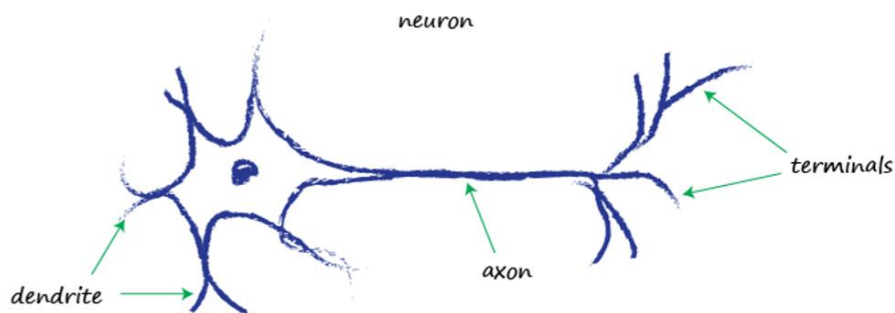


Figure 21 L'unité de base d'un cerveau biologique : le neurone [20]

Les neurones, bien qu'ils existent sous diverses formes, transmettent tous un signal électrique d'un bout à l'autre, des dendrites le long des axones aux terminaux. Ces signaux sont ensuite transmis d'un neurone à un autre. C'est ainsi que notre corps perçoit la lumière, le son, la chaleur, etc. Les signaux des neurones sensoriels spécialisés sont transmis le long de notre système nerveux au cerveau, lui-même principalement constitué de neurones.

Comment fonctionne un neurone ? Il prend une entrée électrique et sort un autre signal électrique. En effet, Il supprime l'entrée jusqu'à ce qu'elle soit devenue si grande qu'elle déclenche une sortie. On peut considérer cela comme un seuil qui doit être atteint avant de produire une sortie. C'est comme de l'eau dans une tasse, l'eau ne déborde pas avant d'avoir rempli la tasse pour la première fois. Intuitivement, cela a du sens, les neurones ne veulent pas transmettre de minuscules signaux de bruit, seulement des signaux intentionnels fortement forts. Les figures 22 et 23 illustrent cette idée de produire un signal de sortie uniquement si l'entrée est suffisamment commutée pour passer un seuil (*threshold*).

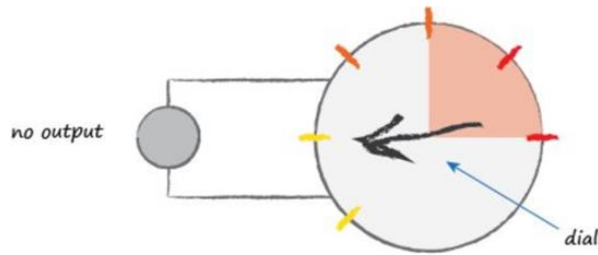


Figure 22 L'entrée n'a pas encore atteint le seuil [20]

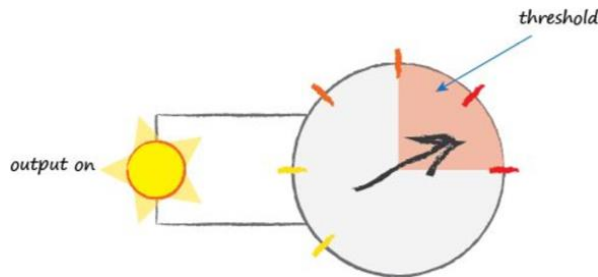


Figure 23 L'entrée a atteint le seuil [20]

## 4.2. La fonction d'activation

Une fonction qui prend le signal d'entrée et génère un signal de sortie, mais qui prend en compte une sorte de seuil est appelée fonction d'activation (*activation function*). Mathématiquement, il existe de nombreuses fonctions d'activation de ce type qui pourraient obtenir cet effet. Une simple fonction échelon unité (*step function*) pourrait le faire :

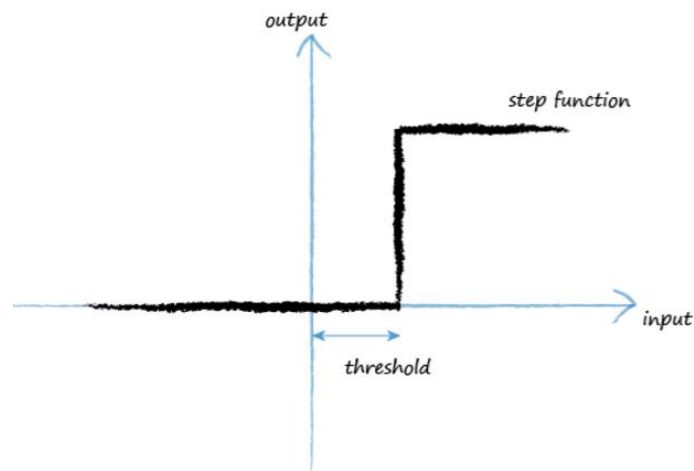


Figure 24 La fonction échelon unité [20]

On peut voir pour les valeurs d'entrée faibles, la sortie est nulle. Cependant, une fois le seuil d'entrée atteint, la sortie saute. Un neurone artificiel se comportant ainsi serait comme un véritable neurone biologique, il se déclenche lorsque l'entrée atteint le seuil.

La fonction en forme de S illustrée ci-dessous est appelée fonction sigmoïde (*sigmoid function*). Elle est plus douce que *step function* qui a une variation brusque, ce qui la rend plus naturelle et réaliste. La fonction sigmoïde, parfois aussi appelée fonction logistique (*logistic function*), est exprimée par

$$l'expression suivante : y = \frac{1}{1+e^{-x}} .$$

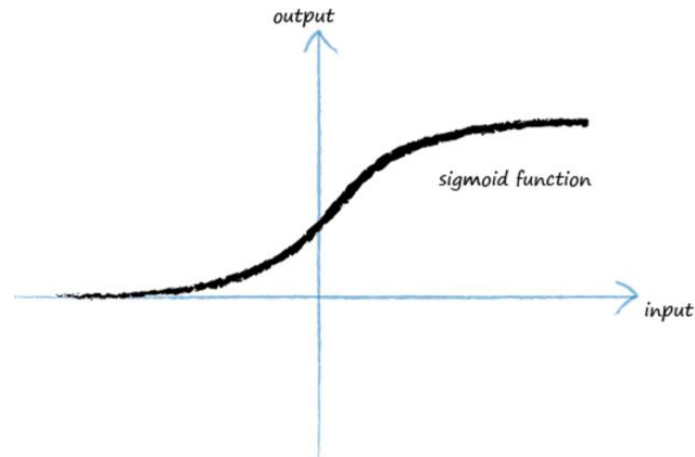


Figure 25 La fonction sigmoïde [20]

Cette fonction sigmoïde en forme de S est ce que nous continuerons à utiliser pour créer notre propre réseau de neurones. Les chercheurs en intelligence artificielle utiliseront également d'autres fonctions d'aspect similaire, mais le sigmoïde est simple et en fait très courant aussi. Il existe une autre raison très puissante de choisir cette fonction sigmoïde parmi les nombreuses autres fonctions en forme de S que nous aurions pu utiliser pour la sortie d'un neurone. La raison en est que cette fonction sigmoïde est beaucoup plus facile à faire des calculs que les autres fonctions en forme de S.

D'autres fonctions d'activations peuvent être utilisées au niveau des nœuds de certaines couches d'un réseau de neurones, parmi eux la fonction rampe (*rectifier function*) :  $y = \max(x, 0)$ .

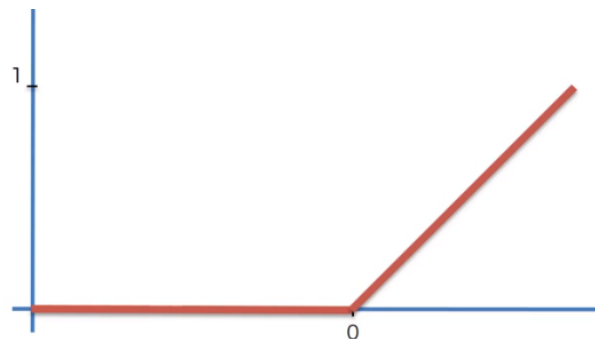
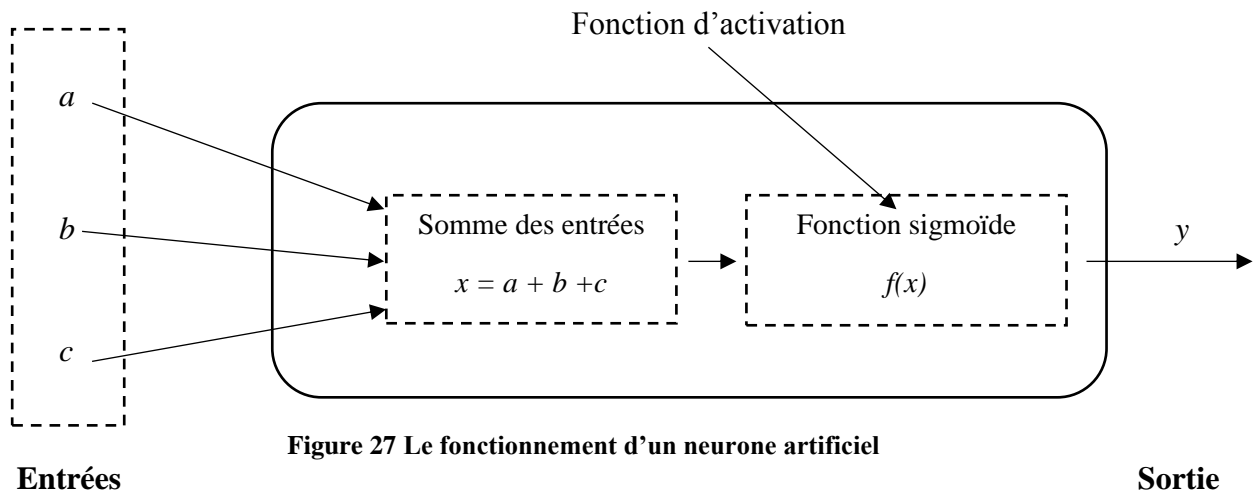


Figure 26 La fonction rampe [21]

### 4.3. Le neurone artificiel

Comment modéliser un neurone artificiel ? La première chose à réaliser est que les vrais neurones biologiques prennent de nombreuses entrées, pas seulement une. Que faisons-nous de toutes ces entrées ? Nous les combinons simplement en les additionnant, et la somme résultante est l'entrée de la fonction sigmoïde qui contrôle la sortie. Cela reflète le fonctionnement des vrais neurones. Le diagramme suivant illustre cette idée de combiner les entrées, puis d'appliquer la fonction d'activation à la somme obtenue.

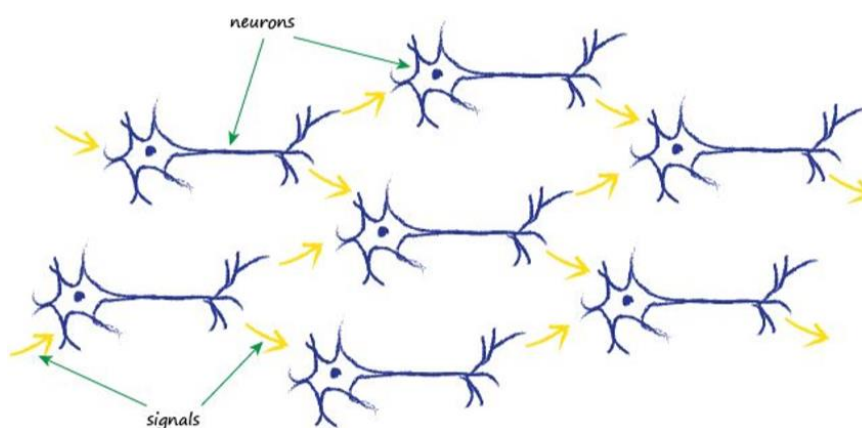




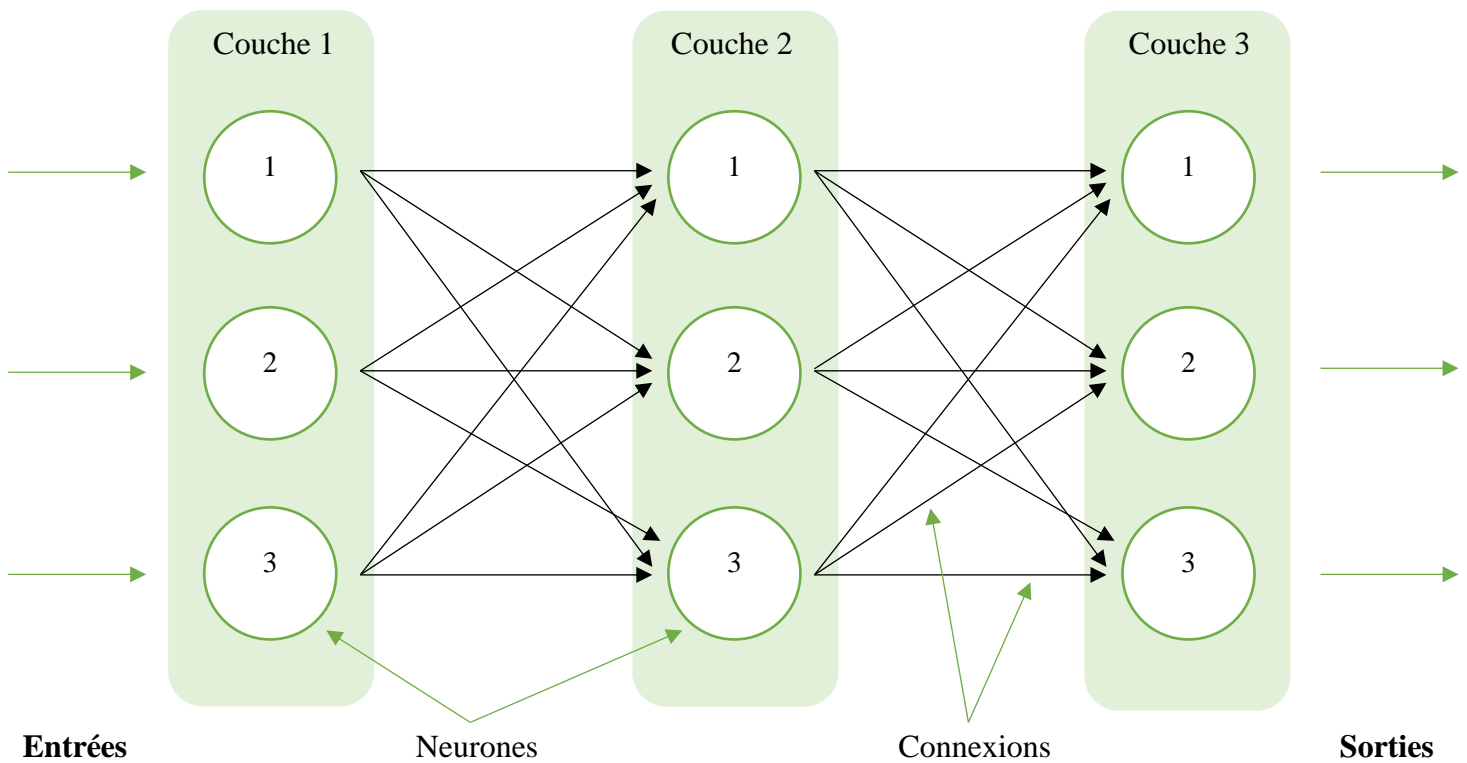
Si le signal combiné  $x$  n'est pas suffisamment grand, alors la fonction de seuil sigmoïde a pour effet de supprimer le signal de sortie. Si la somme  $x$  est suffisamment grande, l'effet du sigmoïde est de déclencher le neurone. Fait intéressant, si une seule des nombreuses entrées est grande et les autres petites, cela peut être suffisant pour déclencher le neurone. De plus, le neurone peut se déclencher si certaines des entrées sont individuellement presque, mais pas tout à fait, assez grandes, car lorsqu'elles sont combinées, le signal est suffisamment grand pour dépasser le seuil. De manière intuitive, cela nous donne une idée des calculs plus sophistiqués et dans un sens flou que ces neurones peuvent faire.

#### 4.4. Un réseau de neurones artificiels

Les signaux électriques sont collectés par les dendrites et ceux-ci se combinent pour former un signal électrique plus fort. Si le signal est suffisamment fort pour franchir le seuil, le neurone envoie un signal le long de l'axone vers les terminaux pour passer sur les dendrites du neurone suivant. Le schéma suivant montre plusieurs neurones connectés de cette manière.



La chose à noter est que chaque neurone prend l'entrée de plusieurs neurones avant lui, et fournit également des signaux à d'autres, s'il se déclenche. Une façon de reproduire cela de la nature à un modèle artificiel est d'avoir des couches de neurones, chacun étant connecté à l'autre dans la couche précédente et suivante. Le schéma suivant illustre cette idée.



**Figure 29 Un réseau de neurones artificiels**

Nous pouvons voir les trois couches, chacune avec trois neurones artificiels ou nœuds. Nous constatons également que chaque nœud est connecté à tous les autres nœuds dans les couches précédentes et suivantes (le réseau n'est pas encore entraîné). Mais quelle partie de cette architecture fait l'apprentissage ? Que modifions-nous en réponse aux données d'apprentissage (*training data*) ? Y a-t-il un paramètre que nous pouvons affiner ? La chose la plus évidente est d'ajuster le poids des connexions entre les nœuds. Au sein d'un nœud, nous aurions pu ajuster la somme des entrées, ou nous aurions pu ajuster la forme de la fonction de seuil sigmoïde, mais c'est plus compliqué que d'ajuster simplement le poids des connexions entre les nœuds.

Le poids  $W_{2,3}$  est simplement le poids associé au signal qui est passé entre le nœud 2 dans une couche au nœud 3 dans la couche suivante. Donc  $W_{1,2}$  est le poids qui diminue ou amplifie le signal entre le nœud 1 et le nœud 2 dans la couche suivante.

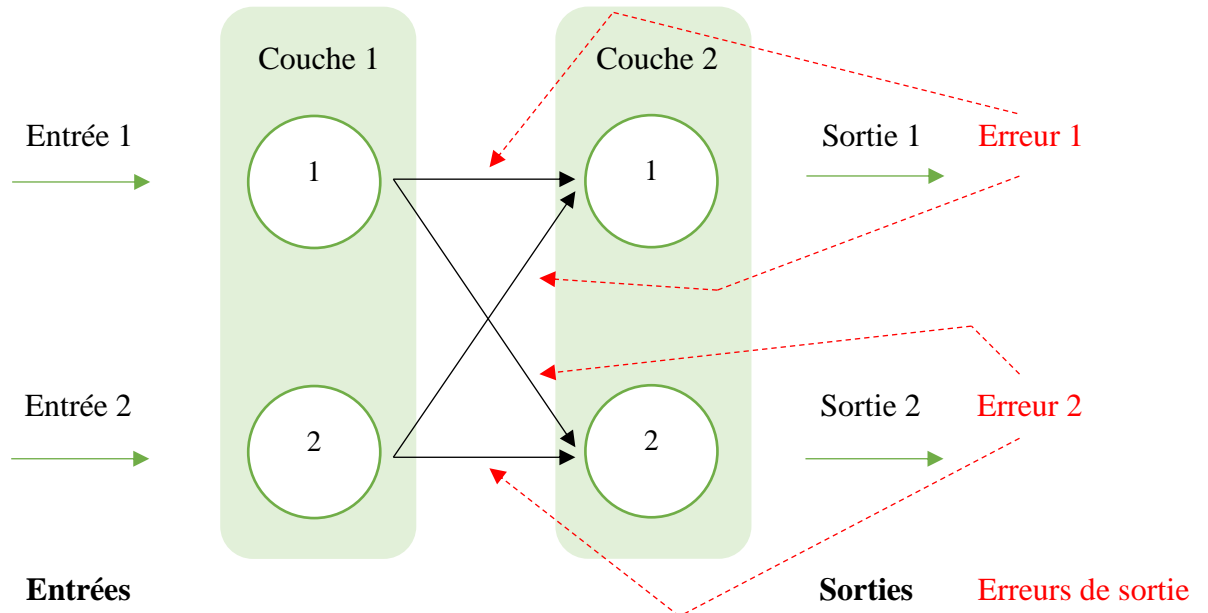
Le réseau apprend à améliorer ses sorties en affinant les poids de liaison à l'intérieur du réseau, certains poids deviennent nuls ou proches de zéro. Un poids nul ou presque nul signifie que ces liens ne contribuent pas au réseau car les signaux ne passent pas. Un poids nul signifie que les signaux sont multipliés par zéro, ce qui donne zéro, de sorte que le lien est effectivement rompu.

#### 4.5. La rétropropagation (Backpropagation)

Nous constatons que nous utilisons les poids de deux manières. Premièrement, nous utilisons les poids pour propager les signaux vers l'avant depuis la couche d'entrée vers la couche de sortie dans un réseau

neuronal. Deuxièmement, nous utilisons les poids pour propager l'erreur vers l'arrière depuis la sortie vers le réseau. Cette méthode est appelée rétropropagation (*backpropagation*).

Si la couche de sortie avait 2 nœuds, nous ferions de même pour le deuxième nœud de sortie. Ce deuxième nœud de sortie aura sa propre erreur, qui est également répartie sur les liens de connexion. Le diagramme suivant montre un réseau simple avec 2 nœuds d'entrée, mais maintenant avec 2 nœuds de sortie.



**Figure 30** Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds (avec les erreurs de sorties calculées et propagées vers le réseau)

Les deux nœuds de sortie peuvent avoir une erreur, en fait, cela est extrêmement probable lorsque nous n'avons pas entraîné le réseau de neurones. Nous pouvons voir que ces deux erreurs doivent informer le raffinement des poids des liens internes dans le réseau. Nous pouvons utiliser la même approche que précédemment, où nous divisons l'erreur d'un nœud de sortie parmi les liens contributifs, d'une manière proportionnée à leur poids.

## 5. CNN (Convolutional Neural Network)

### 5.1. Définition

Un réseau neuronal convolutif (ConvNet/CNN) est un algorithme d'apprentissage en profond qui peut prendre en entrée une image, attribuer une importance (poids et biais) à divers aspects/objets de l'image et être capable de différencier les uns des autres. Le prétraitement requis dans un ConvNet est beaucoup plus faible par rapport à d'autres algorithmes de classification. Alors que dans les méthodes primitives, les filtres sont conçus à la main, avec une formation suffisante, les ConvNets ont la capacité d'apprendre ces filtres/caractéristiques.

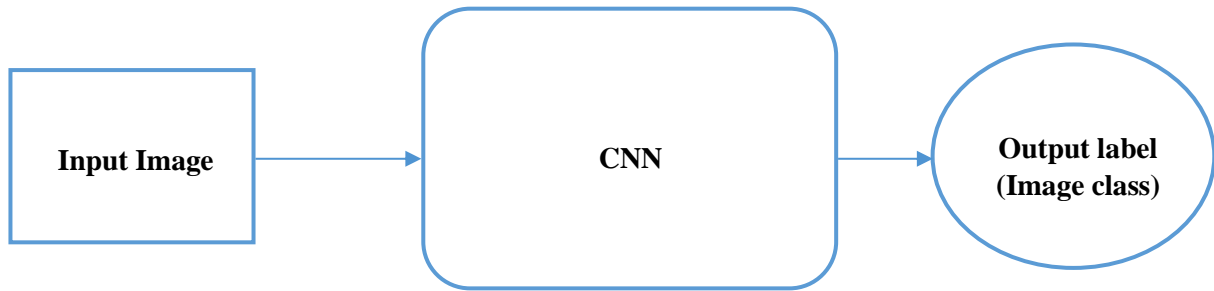


Figure 31 Schéma général d'un CNN

Les réseaux de neurones convolutifs sont une sous-catégorie des réseaux de neurones, qui se sont révélés très efficaces dans les domaines de la reconnaissance et la classification d'images. Les ConvNets (*Convolutional Neural Networks*) ont réussi à identifier les visages, les objets et les panneaux de signalisation, en plus d'alimenter la vision des robots et des voitures autonomes. Les réseaux de neurones convolutifs fonctionnent car ils constituent une bonne extension de l'algorithme standard d'apprentissage en profondeur. Si nous possédons une machine contenant assez de ressources et capacité de calcul, on n'aura pas besoin de la convolution, car, l'algorithme standard fonctionnera également. Cependant, la convolution est plus efficace, puisqu'elle réduit le nombre de paramètres.

## 5.2. L'opération de convolution

Le but principal de la convolution est de trouver des caractéristiques dans une image en utilisant un détecteur de caractéristiques et en les mettant dans une carte de caractéristiques, les relations spéciales entre les pixels seront préservées. C'est très important de comprendre que les caractéristiques dans un réseau de neurones vont être détectées et utilisées pour reconnaître la classe à laquelle appartient l'image en entrée du réseau neuronal convolutif.

Pour construire un réseau de neurones convolutif on doit passer par certaines étapes ou couches comme montre la figure ci-dessus.

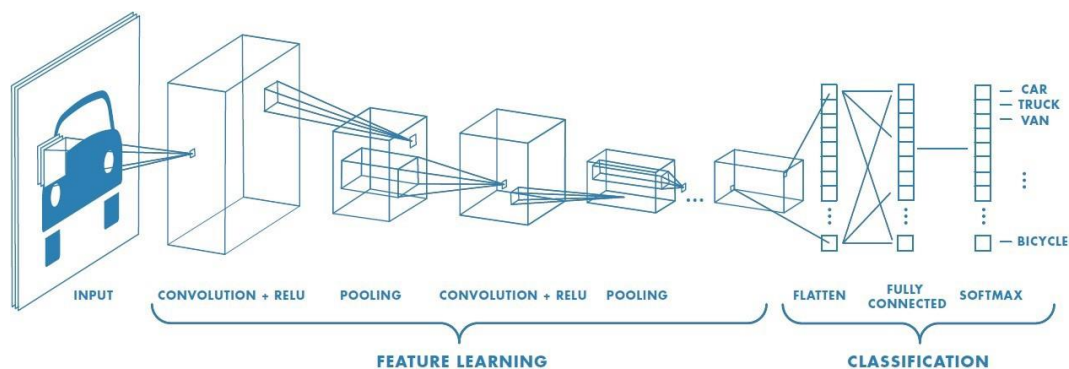


Figure 32 Schéma détaillé d'un CNN [22]

## 5.3. Couche d'entrée (Input layer)

La couche d'entrée contient les données de l'image. Dans la figure suivante, la couche d'entrée se compose de trois neurones. Dans une couche entièrement connectée, les neurones entre deux couches

adjacentes sont entièrement connectés par paire mais ne partagent aucune connexion au sein d'une couche. En d'autres termes, les neurones de cette couche ont des connexions complètes à tous les neurones de la couche précédente. Par conséquent, leurs activations peuvent être calculées avec une simple multiplication matricielle, en ajoutant éventuellement un terme de biais. La différence entre une couche entièrement connectée et une couche convolutionnelle est que les neurones d'une couche convolutionnelle sont connectés à une région locale en entrée et qu'ils partagent également des paramètres.

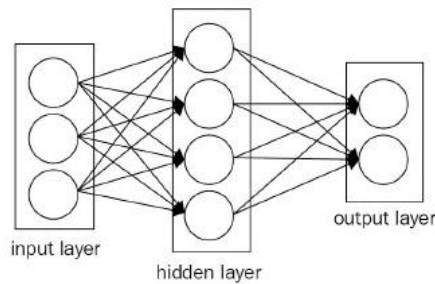


Figure 33 Les trois couches d'un CNN [23]

#### 5.4. Les étapes du traitement effectué dans la couche cachée (Hidden layer)

Pour déterminer le contenu d'une image ou sa classe, on doit passer par les étapes suivantes :

##### Étape 1 :

- **Couche de convolution** : on doit ressortir des caractéristiques depuis l'image en entrée du réseau neuronal convolutif, si elle est en couleur (*RGB* pour *Red*, *Green* et *Blue*) alors on appliquera des filtres pour chaque canal *RGB*. Comme exemple on a pris un visage dessiné à la main puis on représente l'image sous forme de matrice en noire et blanc, les cases noires correspondent à la valeur 1 et celles en blanc à la valeur 0.

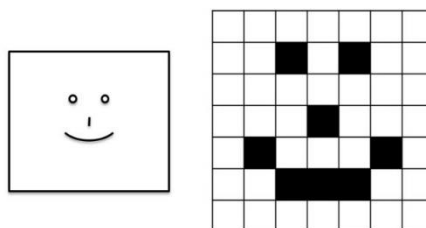


Figure 34 Transformation d'une image en une matrice [24]

Après avoir représenté l'image sous forme d'une matrice on la multiplie par une autre matrice nommée détecteur de caractéristique (*Feature Detector*), cette multiplication correspond à une opération de convolution. Le résultat de la multiplication est une matrice appelé carte de caractéristique (*Feature Map*).

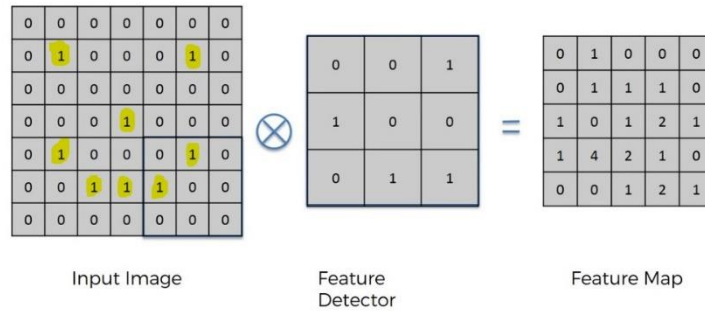


Figure 35 opération de convolution [24]

Cette étape a permis de réduire la taille de l'image ce qui va permettre de faciliter son traitement et de le rendre plus rapide.

- **ReLU (Rectified Linear Unit)** : la fonction d'activation ReLU est extrêmement utile dans les ConvNets il a pour but d'augmenter la non linéarité dans notre réseau de neurones convolutif, elle est appliquée juste après une couche de convolution.

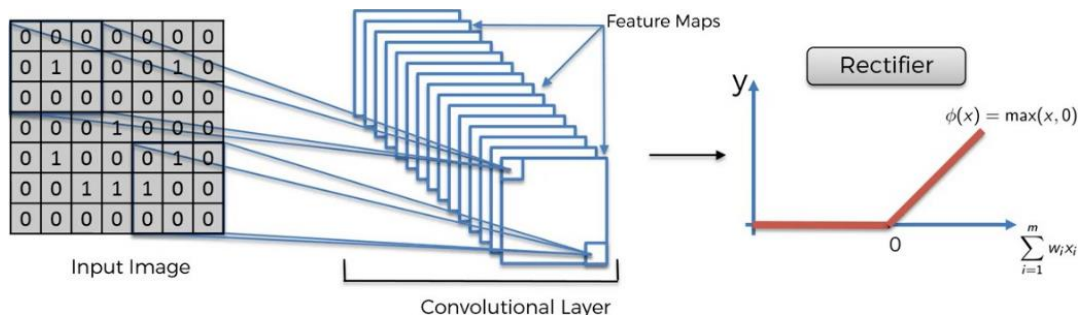


Figure 36 Application de la fonction d'activation ReLU aux feature maps en sorties d'une couche de convolution [24]

**Etape 2 :**

- **Couche de Pooling** : l'opération de *Pooling* sera appliquée aux *Feature Maps*, cette couche nous permet de réduire encore plus le nombre de paramètres qui vont être transmis à l'étape final. Des images peuvent contenir le même objet mais avec différents angles de vue et positions dans l'image, l'opération de *Pooling* nous permet de chercher les caractéristiques peu importe sa position dans l'image ou son angle de vue, cette propriété est appelée « Special invariance ».

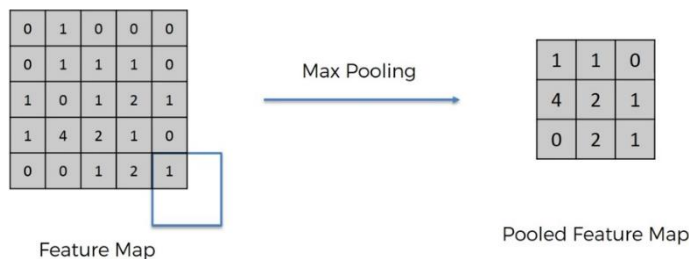


Figure 37 Appliquer l'opération de pooling aux Feature Maps [24]

**Etape 3 :**

- **Couche de Flattening** : le flattening est appliqué aux *feature maps* obtenues après

l'application de l'opération de *pooling*, elles sont aplaties en un vecteur à une seule dimension.

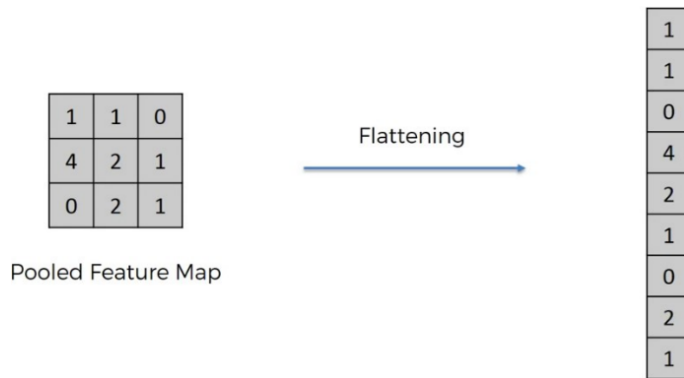


Figure 38 Aplatir la matrice Pooled Feature Map en un vecteur à une seule dimension [24]

**Étape 4 :**

- **Fully connected layer** : le dernier composant d'un réseau de neurones convolutif est un réseau de neurones artificiels composés de couches entièrement connectées (couche d'entrée, couches cachées et une couche de sortie). Le résultat de la couche de *flattening* alimente le réseau de neurones artificiels qui pilote la décision finale de la classification. En résumé, le réseau de neurones artificiels prend les entrées de l'analyse des entités et applique des pondérations pour prédire la bonne classe.

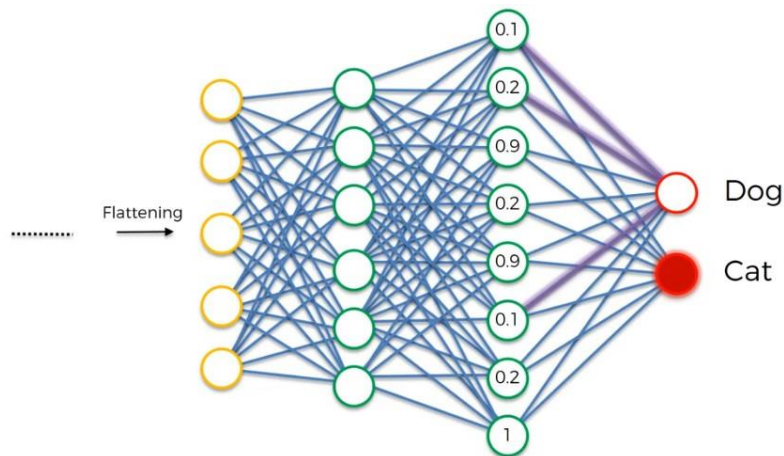


Figure 39 Représentation de la couche Fully connected [25]

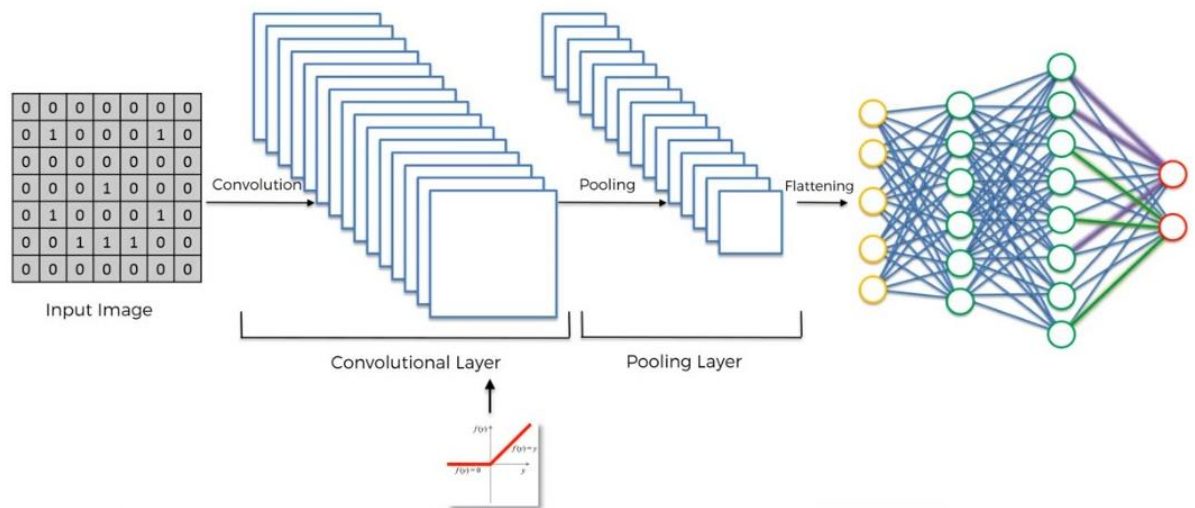


Figure 40 Une vue globale sur les différentes étapes effectuées par un réseau de neurones convolutif pour classer une image en entrée du réseau [25]

### 5.5. Architectures CNN pour la vision par ordinateur

Dans cette partie, on enchaîne avec les algorithmes d'apprentissage en profondeur qui ont résolu quelques-unes des tâches de la vision par ordinateur, et qui se résument comme suit :

- *R-CNN (Region-based Convolutional Neural Network)* [26]
- *Fast R-CNN* [27]
- *Faster R-CNN* [28]

Au cours de la dernière décennie, les réseaux de neurones convolutifs ont acquis une grande popularité dans le domaine de la détection d'objets. Cependant, l'utilisation d'un réseau neuronal convolutif classique avec une couche entièrement connectée (*fully connected layer*) à la fin pour détecter différents objets dans une image n'est pas possible en raison de deux obstacles principaux. Premièrement, le nombre d'objets à détecter dans une image n'est pas connu auparavant, ce qui entraîne la variabilité de la longueur de la couche de sortie (*output layer*). Deuxièmement, même si nous avons décidé d'appliquer un réseau neuronal convolutif à chaque région d'intérêt de l'image pour détecter si elle contient un objet, les objets ont généralement des tailles et des emplacements spatiaux différents à l'intérieur de l'image, ce qui nous oblige à étudier une énorme quantité de régions d'intérêt. Cela le rend impraticable sur le plan des calculs. Par conséquent, des algorithmes comme *R-CNN* ont été développés pour trouver ces occurrences d'une façon rapide.

L'algorithme *R-CNN* a été proposé en 2014 pour contourner le problème de la sélection d'un grand nombre de régions, où il utilise la recherche sélective pour extraire seulement 2000 régions de l'image et il les a appelés propositions de région (*region proposals*), c'est les régions d'une image contenant potentiellement des objets. Par conséquent, maintenant, au lieu d'essayer de classer un grand nombre de



régions, nous pouvons simplement travailler avec 2000 régions. Ces 2000 propositions de région sont générées en utilisant l'algorithme de recherche sélective qui est écrit ci-dessous :

- Générer une sous-segmentation initiale : nous générons de nombreuses régions candidates.
- Utilisez un algorithme pour combiner récursivement des régions similaires en régions plus grandes.
- Utilisez les régions générées pour produire les propositions finales de régions candidates.

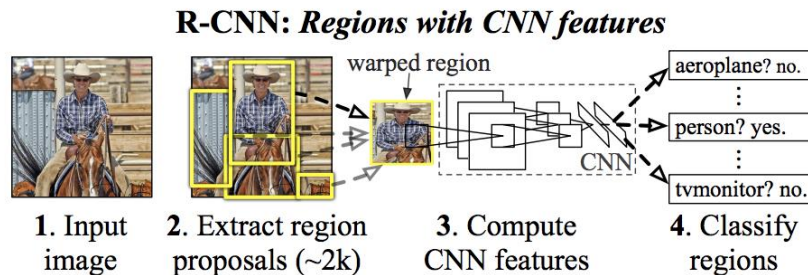


Figure 41 Principe de fonctionnement de l'algorithme R-CNN - 1 [29]

Ces 2000 propositions de régions candidates sont réajustées en un carré et utilisées comme entrée d'un réseau neuronal convolutif qui produit comme sortie un vecteur de *feature* à 4096 dimensions. Le réseau neuronal convolutif agit comme un extracteur de *feature* et la sortie de la couche *dense* sont ces *features* extraites de l'image, puis ces dernières sont introduites dans un *SVM (Support Vector Machine)* [30] pour classer la présence de l'objet dans cette proposition de région candidate. En plus de prédire la présence d'un objet dans les propositions de région, l'algorithme prédit également quatre valeurs qui sont des valeurs de décalage pour augmenter la précision du cadre de délimitation. Par exemple, étant donné une proposition de région, l'algorithme aurait prédit la présence d'une personne, mais le visage de cette personne dans cette proposition de région aurait pu être réduit de moitié. Par conséquent, les valeurs de décalage aident à ajuster le cadre de délimitation de la proposition de région.

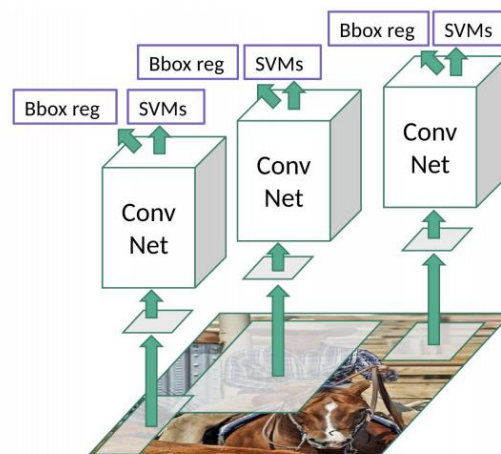


Figure 42 Principe de fonctionnement de l'algorithme R-CNN - 2 [29]

*R-CNN* avait trois inconvénients évidents. Premièrement, Il faut encore énormément de temps pour entraîner (*train*) le réseau, car il faudrait classer 2000 propositions de région par image. Deuxièmement, Il ne peut pas être implémenté en temps réel car il faut environ 47 secondes pour chaque image de test. Troisièmement, L'algorithme de recherche sélective est un algorithme fixe. Par conséquent, aucun apprentissage ne se produit à ce stade. Cela pourrait conduire à la génération de mauvaises propositions de régions candidates.

Les mêmes auteurs de *R-CNN* ont résolu certains des inconvénients de ce dernier pour construire un algorithme de détection d'objet plus rapide et il s'appelait *Fast R-CNN*. L'approche est similaire à l'algorithme *R-CNN*, mais, au lieu de transmettre les propositions de régions au réseau neuronal convolutif, il reçoit comme entrée l'image entière pour générer un *feature map*. A partir du *feature map* nous identifions les propositions de régions et les réajustant en un carré, puis en utilisant une couche *RoI pooling*, nous les remodelons en une taille fixe afin qu'elle puisse être introduite dans une couche entièrement connectée. A partir du vecteur *RoI feature*, nous utilisons une couche *softmax* pour prédire la classe de la région proposée ainsi que les valeurs de décalage pour le cadre de délimitation.

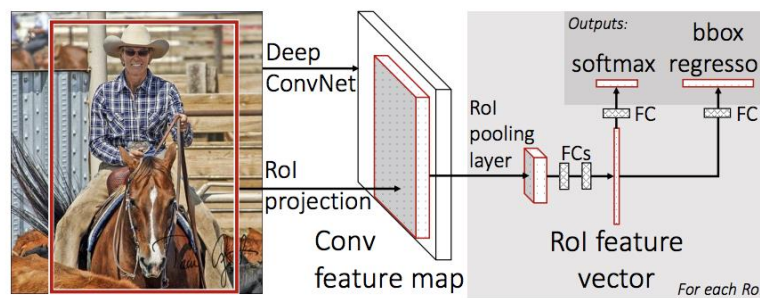


Figure 43 Principe de fonctionnement de l'algorithme Fast R-CNN [29]

La raison pour laquelle *Fast R-CNN* est plus rapide que *R-CNN* est parce que nous n'avons pas à envoyer à chaque fois 2000 propositions de région au réseau neuronal convolutif. Au lieu de cela, l'opération de convolution n'est effectuée qu'une seule fois par image (au lieu de 2000 fois) et un *feature map* est générée à partir de celle-ci.

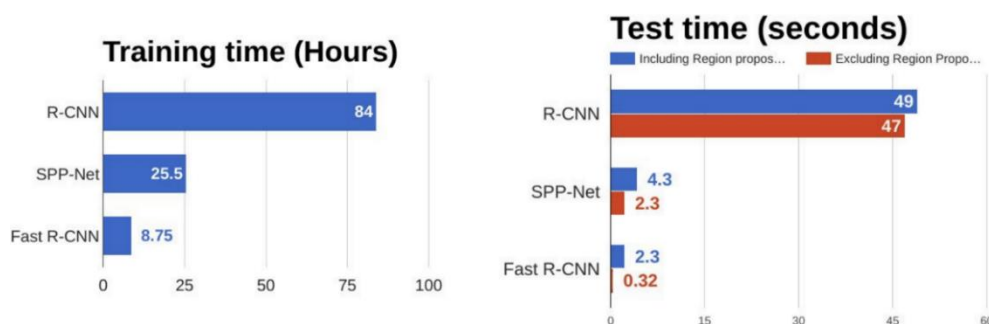
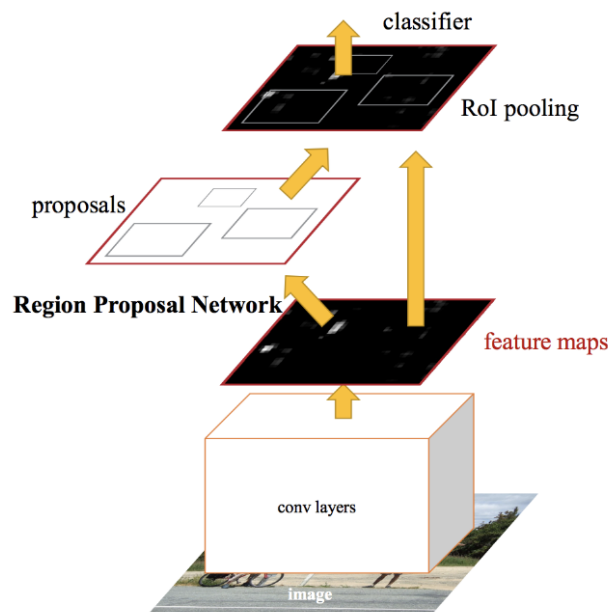


Figure 44 Comparaison des algorithmes de détection d'objets [29]

À partir des graphiques ci-dessus, nous pouvons déduire que *Fast R-CNN* est beaucoup plus rapide dans les sessions d'entraînement (*training*) et de test que *R-CNN*. Lorsque nous regardons les performances

de *Fast R-CNN* pendant le temps de test en incluant les propositions de région, nous constatons qu'elles ralentissent considérablement l'algorithme par rapport à leur non utilisation. Par conséquent, les propositions de région deviennent des goulots d'étranglement (*bottlenecks*) dans l'algorithme *Fast R-CNN* affectant ses performances.

Les deux algorithmes ci-dessus (*R-CNN* et *Fast R-CNN*) utilisent la recherche sélective pour trouver les propositions de région. La recherche sélective est un processus lent et long qui affecte les performances du réseau. Par conséquent, un algorithme de détection d'objets appelé *Faster R-CNN* a été proposé pour éliminer l'algorithme de recherche sélective et permettre au réseau d'apprendre les propositions de région.



**Figure 45 Principe de fonctionnement de l'algorithme Faster R-CNN [29]**

Semblable à *Fast R-CNN*, l'image est fournie comme entrée à un réseau de neurones convolutif qui fournit une *feature map*. Au lieu d'utiliser un algorithme de recherche sélective sur le *feature map* pour identifier les propositions de région, un réseau neuronal supplémentaire (réseau de proposition de région) connu sous le nom de *Region Proposal Network (RPN)* est utilisé pour prédire les propositions de région. Les propositions de région prédites sont ensuite remodelées (à l'aide d'une couche *RoI pooling*) puis utilisées pour classer l'image dans la région proposée et prédire les valeurs de décalage pour les cadres de délimitation.

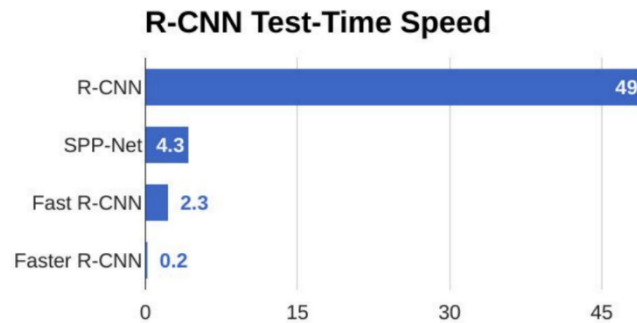


Figure 46 Comparaison du temps de test effectué par les algorithmes de détection d'objets [29]

Sur le graphique ci-dessus, vous pouvez voir que *Faster R-CNN* est beaucoup plus rapide que ses prédécesseurs. Par conséquent, il peut même être utilisé pour la détection d'objets en temps réel.

## 5.6. Le Framework Mask R-CNN

*Mask R-CNN* [31] a été proposé comme une extension de *Faster R-CNN*. Il est destiné à résoudre les problèmes de segmentation d'instances dans la vision par ordinateur. En d'autres termes, il peut séparer différents objets dans une image ou une vidéo. On lui donne une image, il nous retourne les cadres de délimitation qui englobent les objets, les étiquettes de classes auxquelles appartient chaque objet, mais aussi les masques qui s'appliquent sur les objets obtenant ainsi la forme de chaque objet. Les auteurs de l'article ont eu l'idée d'ajouter une branche supplémentaire à l'architecture de *Faster R-CNN*. La branche d'origine était destinée à la classification et au raffinement des boîtes délimitant (*bounding box*) un objet donné, obtenant comme sortie les cadres de délimitations et les étiquettes de classe des objets qu'ils contiennent. D'un autre côté, la nouvelle branche vise à prédire les masques de segmentation dans chaque région d'intérêt, donnant en sortie le masque de l'instance contenue dans chaque cadre de délimitation. Le *Mask R-CNN* s'appuie sur un modèle qui est entraîné (*trained*) sur un jeu de données (*dataset*) connu sous le nom de *COCO* [32] pour prédire la classe et le masque de segmentation de l'objet contenu dans la région d'intérêt. *COCO* signifie *Common Objects in Context*, comme l'indique son nom, les images du jeu de données *COCO* sont prises à partir de scènes complexes quotidiennes contenant des objets communs dans leur contexte naturel. Il contient des photos de 80 types d'objets (personne, vélo, voiture, chaise, livre, etc.) avec un total de 1,5 millions d'instances étiquetées, segmentées et délimitées par un cadre de délimitation dans 330 000 images.

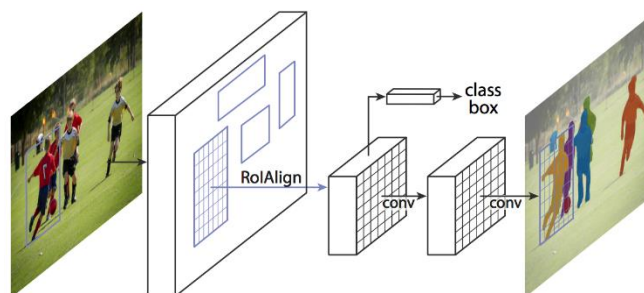


Figure 47 Le framework Mask R-CNN pour la segmentation d'instance [33]

*Mask R-CNN* se compose des modules suivants :

### 5.6.1. Backbone

Il s'agit d'un réseau neuronal convolutif standard (généralement *ResNet50* ou *ResNet101*) qui sert d'extracteur de *feature*. Les premières couches détectent les *features* de bas niveau (bords et coins) et les dernières couches détectent successivement les *features* de niveau supérieur (voiture, personne, ciel).

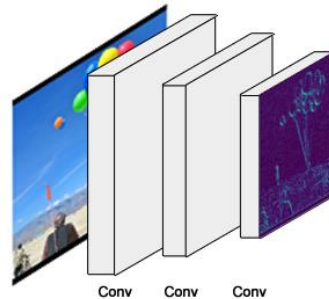


Figure 48 Illustration simplifiée du réseau backbone [33]

En passant par le réseau *backbone*, l'image est convertie de  $1024 \times 1024$  pixels  $\times 3$  (*RGB*) en un *feature map* de forme  $32 \times 32 \times 2048$ . Ce *feature map* devient l'entrée pour les étapes suivantes.

### 5.6.2. Feature Pyramid Network

Bien que le réseau *backbone* décrit ci-dessus fonctionne très bien, il peut être amélioré. Le *Feature Pyramid Network (FPN)* a été introduit par les mêmes auteurs de *Mask R-CNN* comme une extension qui peut mieux représenter des objets à plusieurs échelles.

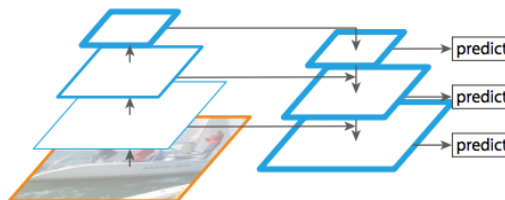


Figure 49 Feature Pyramid Network [33]

*FPN* améliore la pyramide d'extraction de *feature* standard en ajoutant une deuxième pyramide qui prend les *features* de haut niveau de la première pyramide et les transmet aux couches inférieures. En faisant cela, il permet aux *features* de tous les niveaux d'avoir accès aux *features* de niveau inférieur et supérieur.

### 5.6.3. Region Proposal Network (RPN)

Le réseau de proposition de région (*Region Proposal Network (RPN)*) est un réseau de neurones léger, qui balaye l'image à la manière d'une fenêtre coulissante et génère des propositions de région (zones susceptibles de contenir un objet). Les régions parcourues par ce réseau sont appelées ancres (*anchors*), ce sont des boîtes réparties sur la zone d'image, il existe environ 200 000 ancres de tailles et de rapports d'aspect (*aspect ratios*) différents, et elles se chevauchent pour couvrir autant d'image que possible.

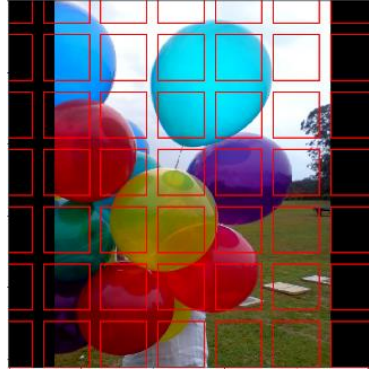


Figure 50 Illustration simplifiée montrant 49 boîtes d'ancrage [33]

Réellement, le réseau de proposition de région ne balaye pas directement l'image mais il analyse le *feature map* issue du réseau *backbone*, cela permet au *RPN* de réutiliser efficacement les *features* extraits et d'éviter la duplication des calculs. Avec ces optimisations, le réseau de proposition de région fonctionne en environ 10 ms selon l'article de *Faster R-CNN* qui l'a introduit. Dans *Mask R-CNN*, nous utilisons généralement des images plus grandes et donc plus d'ancres, par conséquent, cela peut prendre un peu plus de temps. Le réseau de propositions de région génère deux sorties pour chaque ancre :

- La classe à laquelle appartient l'ancre (*anchor class*) : l'une des deux classes, premier plan ou arrière-plan. La classe premier plan implique qu'il y a probablement un objet dans cette boîte.
- Le raffinement du cadre de délimitation : une ancre de premier plan (également appelée ancre positive) peut ne pas être parfaitement centrée sur l'objet. Ainsi, le *RPN* estime un delta (un pourcentage de changement en x, y, largeur, hauteur) pour affiner la boîte d'ancrage pour mieux s'adapter à l'objet.

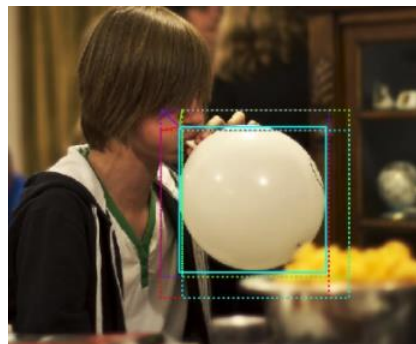


Figure 51 3 boîtes d'ancrage en pointillé et le décalage/redimensionnement qui leur est appliqué pour s'adapter précisément à l'objet (boîtes d'ancrage en solide). Plusieurs ancres peuvent correspondre au même objet [33]

En utilisant les prédictions du *RPN*, nous sélectionnons les ancres susceptibles de contenir des objets et affinons leur emplacement et leur taille. Si plusieurs ancres se chevauchent trop, nous conservons celle qui a le score d'appartenance à la classe premier plan le plus élevé et éliminons le reste (ce procédé est appelé suppression non maximale (*Non-maximum Suppression*)). Après cela, nous obtenons les propositions finales (régions d'intérêt (*Region of Interest* ou *RoI*)) que nous passons à l'étape suivante.



### 5.6.4. RoI Classifier et Bounding Box Regressor

Cette étape opère dans les régions d'intérêt proposées par le *RPN*. Et tout comme le *RPN*, elle génère deux sorties pour chaque région d'intérêt :

- La classe de l'objet dans la région d'intérêt. Contrairement au *RPN*, qui a deux classes (premier plan ou arrière-plan), ce réseau est plus profond et il a la capacité de classer les régions en classes spécifiques (personne, voiture, chaise, etc.). Il peut également générer une classe *background* pour une région d'intérêt donnée, ce qui cause le rejet de cette dernière.
- Affinement du cadre de délimitation : très similaire à la façon dont cela se fait dans le *RPN*, et son objectif est d'affiner davantage l'emplacement et la taille du cadre de délimitation pour qu'il encapsule l'objet.

Il y a un petit problème à résoudre avant de continuer vers la dernière étape. Les classificateurs ne gèrent pas très bien la taille d'entrée variable. Ils nécessitent généralement une taille d'entrée fixe. Mais, en raison de l'étape de raffinement du cadre de délimitation issue du *RPN*, les boîtes *RoI* peuvent avoir différentes tailles. C'est là que le *RoI pooling* entre en jeu.

### 5.6.5. RoI pooling

Le *RoI pooling* fait référence au recadrage d'une partie du *feature map* et à son redimensionnement à une taille fixe. C'est en principe similaire au recadrage d'une partie d'une image, puis à sa redimensionnement (mais il existe des différences dans les détails d'implémentation). Les auteurs de *Mask R-CNN* suggèrent une méthode qu'ils ont nommée *RoI Align*, dans laquelle ils échantillonnent le *feature map* à différents points et appliquent une interpolation bilinéaire.

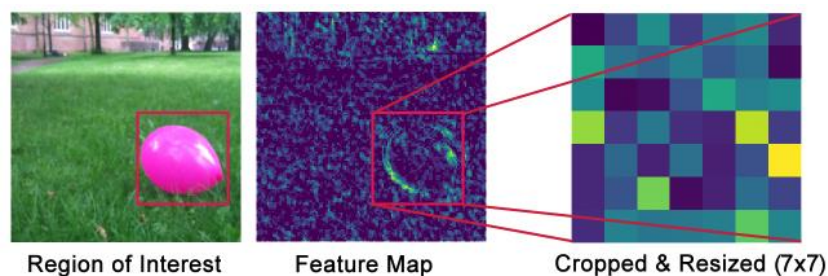


Figure 52 Le feature map ici est issue d'une couche de bas niveau, à titre d'illustration, pour faciliter la compréhension de cette phase [33]

### 5.6.6. Les masques de segmentation

Si nous nous arrêtons là, nous disposons d'un *framework Faster R-CNN* pour la détection d'objets. *Mask R-CNN* ajoute un réseau de neurones convolutif qui prend les régions sélectionnées par le classificateur *RoI* et génère le masque de segmentation pour l'objet contenu dans cette région. Les masques générés ou prédits sont en basse résolution : 28x28 pixels, puis ils sont adaptés à la taille du cadre de délimitation représentant la région d'intérêt et cela nous donne les masques finaux.

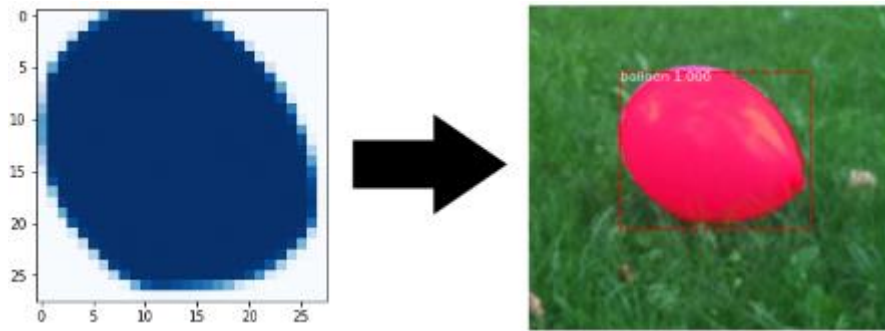


Figure 53 L'adaptation du masque de segmentation 28x28 pixels représentant l'objet au cadre qui le délimite [33]

## 6. Conclusion

Dans ce chapitre, nous avons présenté une vision générale du concept *Machine Learning*, la définition, les types de *Machine Learning*, les réseaux de neurones et les réseaux de neurones convolutif. Ensuite, nous nous sommes intéressés aux concepts de base de la vision par ordinateur qui sont : la classification d'image, la localisation d'objets, la détection d'objets, la segmentation sémantique et d'instance. Enfin, nous avons étudié en détails les différents algorithmes d'apprentissage en profondeur qui ont résolu quelques-unes de ces tâches de la vision par ordinateur, et qui se résument comme suit : *R-CNN*, *Fast R-CNN*, *Faster R-CNN* et *Mask R-CNN*.



# **Chapitre 4 : Étude de la partie matérielle du projet**

## 1. Introduction

D'après ce que nous avons étudié comme capteurs et systèmes de détection des fuites de gaz et des situations de détresse, ainsi que les différents points faibles que nous avons constaté dans ces derniers, qui peuvent ne pas être efficace pour permettre au secours d'agir au temps opportun. La situation peut devenir plus dangereuse dans le cas où la personne est âgée et vive seule ou elle est malvoyante ou malentendant. Donc, il est nécessaire de développer un système de contrôle en temps réel capable d'intervenir d'une manière automatique (exécuter un ensemble d'actions correctives) au niveau du domicile en cas d'une anomalie, en intégrant aussi la possibilité de consulter : l'historique des chutes/changements des taux des gaz nocifs, les causes ayant déclenchés une alerte, de plus, l'installation des capteurs va nous permettre de contrôler et d'améliorer les conditions de l'air.

Dans ce chapitre, nous effectuons en premier lieu, une étude comparative entre les différents types de cartes programmables disponible sur le marché dont le but de sélectionner la plus adaptée à nos besoins. Par la suite, nous présentons les différents capteurs et actionneurs qui nous permettent de contrôler et de gérer les taux des différents gaz toxiques. Enfin, nous clôturons ce chapitre par une étude socioéconomique pour estimer le coût de revient de notre projet.

## 2. Les choix de la solution électronique

Durant notre recherche sur la partie électronique qui va être implémentée pour automatiser les différents actionneurs, nous avons constaté qu'au cours des dernières années, de nombreuses cartes de développement modulaire sont apparues. Chaque carte présente des avantages et des inconvénients. Elles se différencient par leurs fonctionnalités, leurs complexités et leurs prix. On peut citer parmi ces cartes :

### 2.1. Carte de développement Arduino

L'Arduino est un circuit intégré à base d'un microcontrôleur programmable qui peut analyser et produire des signaux électriques de façon à effectuer des tâches précises comme (le pilotage d'un robot, la détection de présence, le contrôle des appareils domestiques, etc.).



Figure 54 Une carte Arduino [34]

### 2.2. Carte de développement Raspberry Pi

Le Raspberry pi est un nano ordinateur de la taille d'une carte de crédit que l'on peut brancher à un écran et utilisé comme un ordinateur standard. Il est développé par une organisation britannique dans le but de fournir aux étudiants des ordinateurs à faible coût.

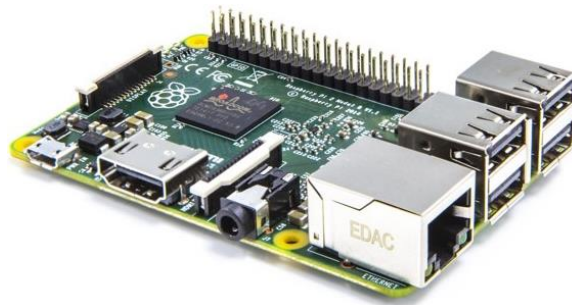


Figure 55 Une carte Raspberry Pi [35]

### 2.3. Carte de développement NodeMCU

Une carte NodeMCU est une petite (3,00 x 5,50 cm) carte électronique équipée d'un microcontrôleur avec une carte wifi intégré pour établir des connexions TCP/IP. Les NodeMCU ont révolutionné le domaine de l'électronique à l'échelle mondiale depuis leur apparition en 2014 en raison de leur très grande capacité et leur faible prix.

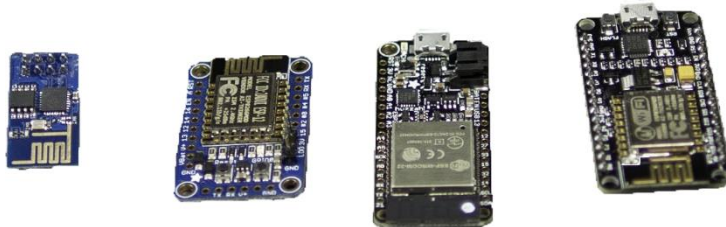


Figure 56 Des cartes NodeMCU [36]

### 2.4. Les Critères de choix de la solution

On a terminé la présentation des cartes existantes sur le marché qui peuvent être utilisées dans notre projet, ces cartes représentent le cerveau du système qui contrôle et gère la collection de l'information et réalise des actions en se basant sur l'information collectée. Nous passons maintenant vers l'une des étapes les plus importantes à la réalisation de notre projet qui est la sélection de la carte la plus adaptée à nos objectifs, il existe plusieurs critères de sélection dont nous devons tenir compte comme :

- Le nombre d'entrées/sorties analogiques ;
- Le nombre d'entrées/sorties numériques ;
- Puissance de calcul suffisamment élevée pour gérer des algorithmes en temps réel ;
- Taille de la mémoire programmée (pour contenir l'ensemble du programme) ;
- La taille de la mémoire RAM (pour les calculs que le microcontrôleur doit effectuer) ;
- La mémoire EEPROM (si on a besoin que certaines données soient sauvegardées si l'alimentation se coupe) ;
- Le prix et la disponibilité sur le marché (le prix est très important car la solution sera déployée pour des gens de majorité ordinaire) ;
- Idéalement : la possibilité de se connecter à internet pour envoyer et recevoir des données depuis un serveur distant.

### 3. Étude comparative

#### 3.1. Table comparative

Dans le tableau qui suit on résume les différentes caractéristiques techniques des cartes de développement qui existent sur notre marché, avec une comparaison, afin de trouver la mieux adaptée à notre système et qui remplit les critères précédemment cités :

Nom de la carte	Arduino UNO	ESP8266	ESP32	Raspberry Pi 3
Processeur	ATmega328P	Xtensa L106 Single-Core	<b>Xtensa Lx6 Dual-Core</b>	Broadcom BCM2837B0
Fréquence	16 MHz	160 Mhz	<b>240 Mhz</b>	1.4 GHz
RAM	32 KB	160 KB	<b>512 KB</b>	1 GB
ROM	2 KB	512 KB (UP to 4 MB)	<b>4 MB (UP TO 16 MB)</b>	MicroSD
EEPROM	1 KB	1 KB	<b>1 KB</b>	MicroSD
E/S Digital	14	16	<b>23</b>	40
E/S Analogique	6	1	<b>18</b>	0
WIFI	NON	OUI	<b>OUI</b>	OUI
Bluetooth	NON	NON	<b>OUI</b>	OUI
Prix – DA	2300	1300	<b>2700</b>	10000

Tableau 6 Étude comparative de quelques cartes disponibles sur le marché

#### 3.2. Discussion du tableau comparative

- En analysant le tableau, on remarque que la carte Arduino UNO est la plus faible coté fréquence du processeur, ainsi qu'elle ne dispose pas de connectivité sans fil, car nous avons besoin d'une communication sans fil dans notre système pour envoyer et recevoir les données vers et depuis un point d'accès Wi-Fi. Donc, cette carte est éliminée de notre choix.
- D'après une recherche réalisée, nous avons trouvé que Le prix de la carte Raspberry PI 3 est le plus élevé par rapport aux autres cartes, et avec les frais de (MicroSD, chargeur, ventilateur, convertisseur Analog to Digital) il peut atteindre jusqu'à 18000 DA ! De plus, il ne possède pas d'entrée analogique (dans notre cas, le capteur de gaz est analogique).
- Selon les spécifications de notre système les cartes **ESP32** et **ESP8266** seront les mieux adaptées, le prix est raisonnable par rapport à ce qu'ils offrent comme hardware, ainsi que les deux possèdent une carte wifi de plus par rapport aux autres, mais nous avons besoin d'un espace ROM important plus que 4 MB pour contenir le programme, plus de 3 pin analogique et plus de 16 pin digital pour des extension future et pour une solution extensible et durable, ce qui nous a conduit à éliminer l'**ESP8266**, donc nous choisissons la carte **ESP32**.

D'après l'analyse du tableau comparatif, on déduit que l'**ESP32** représente un outil plus performant et mieux adapté à nos besoins. L'**ESP32** possède le nombre de ports analogique le plus élevé, une fréquence pouvant atteindre jusqu'à 240 Mhz grâce à un processeur Dual-Core et une grande capacité mémoire ROM et RAM qu'ils lui permettront de gérer des algorithmes complexes, de plus, son prix est plus qu'abordable. Il a aussi la capacité de se connecter à un réseau Wifi pour envoyer et recevoir des données par internet, ce qui le rend la solution la plus adaptée pour la réalisation de notre projet.

## 4. La carte ESP32

### 4.1. Présentation de la carte

L'**ESP32** est une série de microcontrôleurs de type système sur une puce (SoC) produit par la société chinoise Espressif Systems, intégrant la gestion du Wi-Fi et du Bluetooth 802.11b/g/n et permettant d'établir des connexions TCP/IP, en mode client/serveur HTTP. Son support Wi-Fi et Bluetooth, le rend un système apprécié dans le domaine de l'internet des objets. Ce SoC connaît un certain succès depuis quelques années à la fois pour son coût, ses capacités et son intégration dans un nombre croissant de systèmes.

### 4.2. Caractéristiques principales

- Processeur Dual-Core Xtensa LX6 32 bits cadencé jusqu'à 240 MHz.
- Mémoire flash entre 4 MB et 16 MB selon les modèles.
- 520 Ko de RAM pour les instructions et les données.
- Puce Wifi 2.4 GHz (802.11 b/g/n) avec antenne intégré.
- Bluetooth Low Energy (BLE, BT4.0, Bluetooth Smart) intégré.
- Un capteur de température intégré.
- Un capteur à effet Hall intégré.
- Un capteur tactile intégré.
- Compatible avec les sécurités WFA, WPA/WPA2 et WAPI.
- 23 Entrées/sorties numériques GPIO.
- 18 Entrées/sorties Analogiques GPIO.
- Alimentation 2,2 V à 3,6 V par l'intermédiaire du connecteur micro USB.

### 4.3. Programmation de la carte ESP32

On peut utiliser certains langages de programmation pour programmer la carte ESP32 :

- En script Lua, avec le firmware NodeMCU.
- En C, avec le SDK d'Espressif.
- En C, avec l'IDE Arduino.
- En JavaScript, avec le firmware Espruino.

- En MicroPython, avec le firmware MicroPython.

Pour la programmation de l'ESP32, nous allons utiliser l'environnement de développement d'Arduino car cet IDE présente beaucoup d'avantages (gratuit, open-source, multiplateforme, etc.).

## 5. Capteur utilisé

Pour atteindre notre objectif qui est la réalisation d'un system intelligent capable de détecter les fuites de gaz dangereux, nous avons besoin d'un capteur de gaz qui sera intégré dans notre system, Il y a beaucoup de choix dans le marché mais on a choisi le MQ-2 car c'est le mieux adapter à notre projet, il détecte plusieurs gaz nocifs ainsi que la fumée, cela va nous éviter d'utiliser plusieurs capteurs pour chaque gaz spécifique.

### 5.1. Capteur de gaz MQ-2

MQ2 est l'un des capteurs de gaz le plus utilisé dans la série de capteurs MQ. Il s'agit d'un capteur de gaz de type semi-conducteur à oxyde métallique (MOS) également connu sous le nom de Chemiresistor, car la détection est basée sur le changement de résistance du matériel de détection lorsque le gaz entre en contact avec le matériel. À l'aide d'un simple réseau de diviseurs de tension, les concentrations de gaz peuvent être détectées.



Figure 57 Capteur de gaz MQ-2 [37]

#### ➤ Caractéristiques techniques :

- Alimentation : 5V.
- Sortie : une analogique et une numérique.
- Plage de détection : 200 ~ 10000 PPM.
- Gaz détectés : GPL, fumée, alcool, propane, monoxyde de carbone.
- Le temps de réponse :  $\leq 1$  s.
- Dimensions : 32x22x27mm.
- La durée de vie : 5 ans.

## 6. Actionneurs utilisés

Après la détection des gaz toxiques on doit pouvoir exécuter un ensemble d'actions au niveau du domicile pour réagir d'une façon autonome, pour cela, nous avons besoin des actionneurs suivant :

### 6.1. MG996R Servo Motor

Le MG996R est un servomoteur à engrenages métalliques avec un couple de calage maximum de 11 kg/cm. Comme les autres servos RC, le moteur tourne de 0 à 180 degrés en fonction du rapport cyclique de l'onde PWM fournie à sa broche de signal.



Figure 58 Le servomoteur MG996R [38]

➤ **Caractéristiques techniques :**

- Alimentation : 5V.
- Maximum Couple de décrochage : 11 kg/cm (6V).
- Rotation : 0° - 180°.

### 6.2. Relais one channel

Un relais est un interrupteur électromagnétique actionné par un courant relativement faible qui peut contrôler un courant beaucoup plus important. Le module de relais est largement utilisé dans tous les domaines tels que le secteur industriel, contrôle PLC, contrôle d'une maison intelligente, etc. Dans notre prototype, nous avons choisi un module de relais à un seule canal pour contrôler l'alimentation électrique au niveau du domicile est couper l'électricité en cas d'une fuite de gaz.



Figure 59 Relais one channel [39]

➤ **Caractéristiques techniques :**

- Alimentation : 5V.
- Courant de charge AC maximum : 10A - 250/125V AC.
- Courant de charge DC maximum : 10A - 30/28V DC.

### 6.3. Piezo buzzer

Un haut-parleur piézoélectrique est un haut-parleur qui utilise l'effet piézoélectrique pour générer du son. Le mouvement mécanique initial est créé en appliquant une tension à un matériau piézoélectrique, ce mouvement est généralement converti en son audible à l'aide de diaphragmes et de résonateurs. Comparé à d'autres conceptions d'enceintes, les enceintes piézoélectriques sont relativement faciles à piloter, par exemple, ils peuvent être connectés directement aux sorties TTL, bien que des pilotes plus

complexes puissent donner une plus grande intensité sonore. En règle générale, ils fonctionnent bien dans la plage de 1 à 5 kHz et jusqu'à 100 kHz dans les applications à ultrasons.



Figure 60 Piezo buzzer [40]

➤ **Caractéristiques techniques :**

- Alimentation : 1.5V DC.
- Output : 85dB.
- Fréquence de résonance : 2048 Hz.

### 6.4. Module GSM SIM800L

Le module SIM800 est un téléphone GSM simple, sans clavier, écran, micro, ni haut-parleur, mais possédant une liaison série à connecter à un microcontrôleur local. Ce module prend en charge le réseau quadri bande GSM/GPRS et disponible pour la transmission et réception des SMS, de passer des appels, ce qui en fait la solution idéale dans notre projet pour l'envoi des notifications sous forme SMS aux utilisateurs lors d'une fuite de gaz nocifs ou une chute d'une personne.



Figure 61 Module GSM SIM800L [41]

➤ **Caractéristiques techniques :**

- Alimentation : 3,5 ~ 4,4 V.
- Fréquence : 780MHz ~ 960MHz, 1710MHz ~ 2170MHz
- Effectuer et recevoir des appels vocaux à l'aide d'un casque et microphone externe.
- Envoyer et recevoir des messages SMS.
- Envoyer et recevoir des données GPRS (TCP / IP, HTTP, etc.).
- Numériser et recevoir des émissions de radio FM.
- Dimensions : 2.5 cm x 2.3 cm x 0.7 cm.

### 6.5. Moteur driver l298

Le Moteur driver l298 va nous permettre de piloter deux moteurs pour l'ouverture et la fermeture des fenêtres, il est responsable de la direction de rotation des moteurs ainsi que leurs vitesses, le l298 est un



circuit monolithique intégré dans des boîtiers multi watt à 15 conducteurs et PowerSO20. Il s'agit d'un pilote à double pont complet à haute tension et à courant élevé conçu pour accepter les charges inductives d'entraînement de sable de niveau logique TTL standard telles que les relais, les solénoïdes, les moteurs CC et les moteurs pas à pas. Deux entrées d'activation sont fournies pour activer ou désactiver l'appareil indépendamment des signaux d'entrées. Les émetteurs des transistors inférieurs de chaque pont sont connectés ensemble et la borne externe correspondante peut être utilisée pour la connexion d'une résistance de détection externe. Une entrée d'alimentation supplémentaire est fournie pour que la logique fonctionne à une tension inférieure.

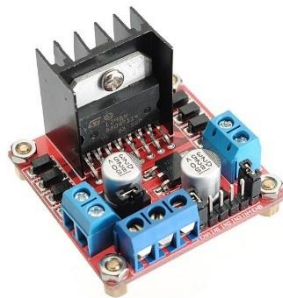


Figure 62 Moteur driver 1298 [42]

➤ **Caractéristiques techniques :**

- Alimentation max : 40 volts.
- Deux ponts en H, haute tension, grand courant, pilote de pont complet, qui peuvent être utilisés pour entraîner des moteurs à courant continu, des moteurs pas à pas, des bobines de relais et d'autres charges inductives.
- Utilisation d'un signal de niveau logique standard pour le contrôle.
- Capable de piloter un moteur pas à pas à deux phases ou un moteur à pas à quatre phases et des moteurs à courant continu à deux phases.
- Drive voltage : 5 - 35V.
- Logique voltage : 5V.
- Dimension : 4.2 x 4.2 cm.

## 6.6. Ventilateur

Le ventilateur permet l'extraction de l'air contaminé par le gaz (réduction du taux du gaz nocif dans l'air). Dans la réalisation de notre prototype, nous utiliserons un ventilateur 12 volts.



Figure 63 Ventilateur 12 volts [43]

## 7. Etude socioéconomique

Pour estimer le coût de fabrication de notre système, nous avons fait une étude socioéconomique afin d'estimer le prix de fabrication.

Composant	Prix unitaire (DA)	Quantité	Total (DA)
ESP32	2700	1	2700
Buzzer	350	1	350
Breadbord	800	1	800
Ventilateur	550	1	550
Driver L298	1000	1	1000
MG996r servo	1900	1	1900
MQ-2	600	1	600
Relay 1 channel	450	2	900
Wire male	20	50	1000
Wire female	20	50	1000
Led	10	2	20
Résistance	5	2	10
Module SIM800L	2200	1	2200
<b>TOTAL (DA)</b>		<b>13030</b>	

**Tableau 7 Estimation du prix de revient de notre projet**

D'après le tableau, le prix de revient de notre projet est de 13030DA, ce qui est très raisonnable par rapport aux avantages qu'il offre.

## 8. Conclusion

Dans ce chapitre, nous avons présenté une étude comparative entre les différentes cartes électroniques disponibles sur le marché en fonction de plusieurs critères pour le but de sélectionner la plus adéquate à nos besoins. Puis, nous avons cité les différents composants et modules ainsi que leurs caractéristiques qui nous permettent de contrôler le système. Enfin, nous avons conclu ce chapitre avec une étude socioéconomique afin de déterminer le coût de réalisation, ce coût paraît acceptable comparé à ce que nous aurons obtenus à la fin.

Le prochain chapitre, sera dédié à la Conception et à la Modélisation de notre système.

# **Chapitre 5 : Analyse et Conception du système**

## 1. Introduction

Dans le chapitre précédent on a exprimé l'objectif de notre travail ainsi que les solutions technologiques les plus adaptées à nos besoins, maintenant on peut passer à la phase de conception et modélisation du système, qui représente une activité clé dans le processus de développement de notre projet de fin d'études. En effet, elle formalise et détaille ce qui a été ébauché au cours de préliminaire, et permet de dégager l'étude fonctionnelle du système. Elle permet ainsi d'obtenir une idée sur ce que va réaliser le système en termes de métier (comportement du système).

D'abord, nous allons présenter la conception et la modélisation de notre système de détection des fuites de gaz, nous commencerons par définir les besoins fonctionnels et les besoins non fonctionnels de la solution que nous allons proposer, ainsi que son architecture logicielle. Ensuite, nous passerons par la modélisation et la conception de notre application à travers les différents diagrammes (diagramme de classes, cas d'utilisation, séquences et de composant).

Ensuite, nous clôturerons cette partie par la présentation d'une conception hardware de différents composants électroniques à travers les schémas de branchement des pièces.

Enfin, nous présenterons la conception de notre système de détection des chutes, en commençant par donner une vue globale et un résumé sur le principe de fonctionnement et les étapes par lesquelles passent notre système, et ce afin qu'il reconnaisse la posture exercée par la personne ou une éventuelle chute, puis, nous détaillerons chaque étape (Détermination de la zone représentant le sol, Extraction de la silhouette du corps humain, Classification de la silhouette dans l'une des deux classes *lie* ou *no lie* à l'aide d'un réseau de neurones convolutif entraîné et évalué sur notre jeu de données, Vérification si la personne est allongée (la classe *lie* est prédite par notre classificateur) dans la zone représentant le sol pendant une durée de 30 secondes ce qui indique qu'un évènement de chute s'est produit).

## 2. Conception et Modélisation du système de détection des fuites de gaz

### 2.1. Le principe de fonctionnement du système de détection des fuites de gaz

Le système peut fonctionner en deux modes qui sont :

**Mode automatique** : le système contrôle (à l'aide du capteur de gaz MQ-2) en permanence et en temps réel les taux actuels des gaz nocifs : GPL, monoxyde de carbone, gaz de ville ou méthane et la fumée, si le taux d'un de ces gaz toxiques dépasse le seuil de tolérance minimale qui est 300 PPM, alors le système intervient d'une manière automatique au niveau du domicile en effectuant les actions suivantes en respectant l'ordre suivant : couper l'alimentation électrique, fermer la vanne principale de gaz, ouvrir les fenêtres, activer la ventilation, envoyer une alerte via un appel ou SMS aux services de secours ainsi qu'aux occupants du domicile se trouvant en dehors de la maison et déclenchée une alarme sonore au niveau du domicile. Une fois le taux actuel du gaz qui a causé l'alerte revient à sa bonne valeur, le

système effectue les actions suivantes : activer l'alimentation électrique, ouvrir la vanne principale de gaz, fermer les fenêtres, désactiver la ventilation et désactiver l'alarme sonore.

**Mode manuel** : si le mode manuel est activé, alors le système ne contrôle plus la maison déléguant ce rôle aux occupants du domicile qui peuvent intervenir sur les actionneurs et les commutateurs qui contrôlent les différents équipements au niveau de la maison : la vanne principale de gaz, la ventilation, les fenêtres, l'alimentation électrique, etc.

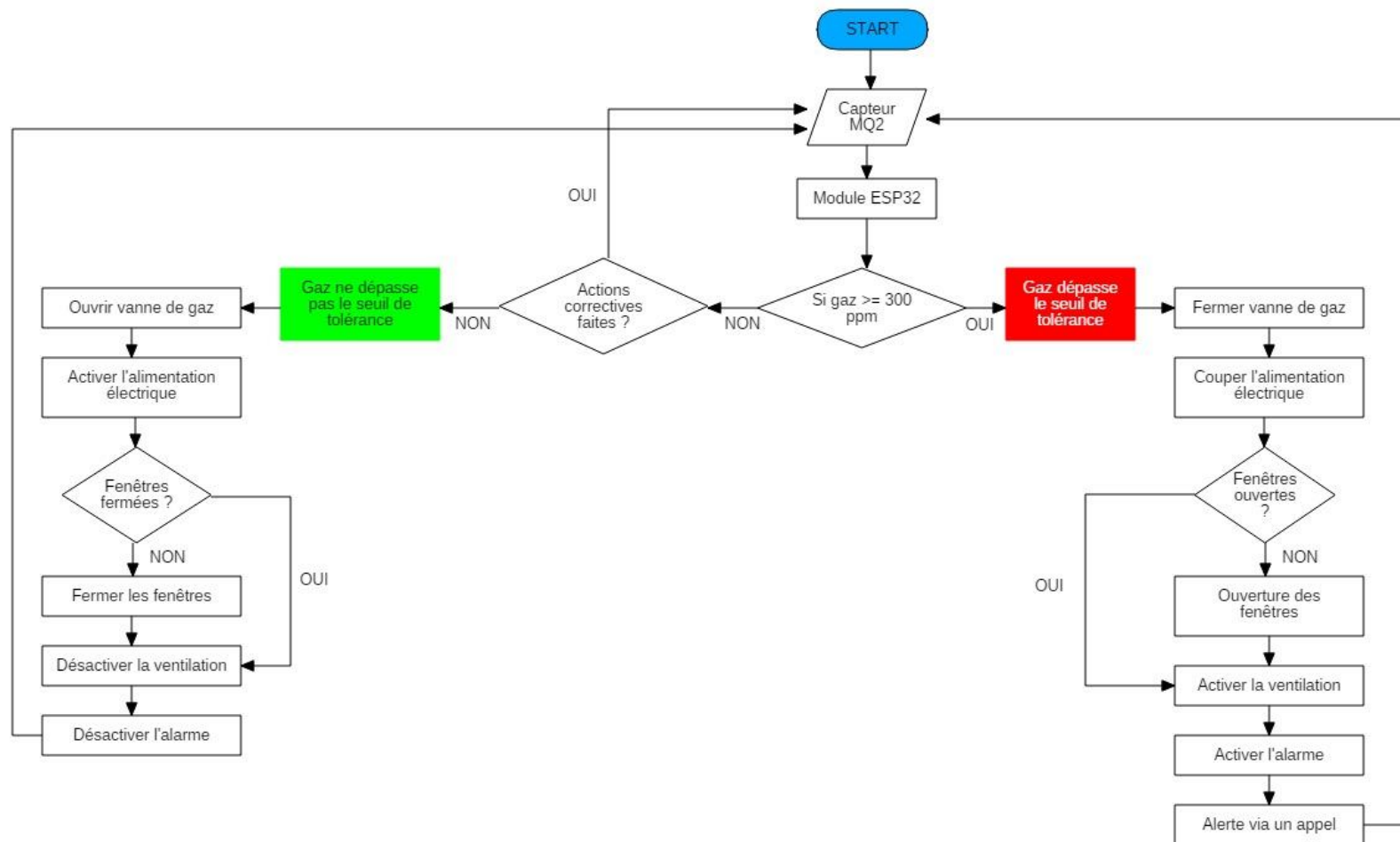


Figure 64 Diagramme expliquant d'une manière générale le principe de fonctionnement du système de détection de fuites de gaz

## 2.2. Spécification des besoins

### 2.2.1. Besoins fonctionnels

Afin de répondre aux besoins des utilisateurs de gaz domestique, qui risque chaque jour et surtout en période hivernale le danger de mort par asphyxie/explosion suite à l'émanation de gaz toxiques, notre système doit être intelligent et réagit d'une manière automatique en temps réel lors d'une anomalie (coupe l'alimentation électrique, ferme la vanne principale de gaz, ouvre les fenêtres, alerte via un appel/SMS les services de secours ainsi que les occupants du domicile se trouvant en dehors de la

maison, etc.), il doit être aussi extensible et permet de superviser et de contrôler le taux des gaz dans la maison.

Notre système comprendra trois (03) fonctionnalités principales que nous allons présenter ci-dessous de façon plus détaillée afin d'être plus clairs :

#### **2.2.1.1. Supervision du système**

Notre système doit permettre à l'utilisateur de suivre en temps réel l'état de l'appartement et de la personne elle-même, en offrant à l'utilisateur la possibilité de :

- Voir l'historique des changements des taux des gaz toxiques dans le temps.
- Accès à distance (à travers Internet).
- Voir l'état actuelle des taux des différents gaz toxiques.
- Historique des alertes.
- Historique des chutes.
- Historique des intrusions de personnes malveillantes au domicile de l'individu.

#### **2.2.1.2. Gestion des maisons et différents actionneurs**

Dans ce contexte, notre système offre la possibilité de gérer les différents équipements et dispositifs (extracteur d'air, vanne principale de gaz, les fenêtres, l'alimentation électrique, etc.) de manière automatique ou manuelle.

#### **2.2.1.3. Gestion des alertes**

Dans l'appartement nous avons un capteur de gaz ainsi qu'un capteur vidéo et actionneurs installés dans des endroits différents, permettent au système d'avoir en temps réel et en permanence des informations sur les taux actuels des gaz présents dans l'air (monoxyde de carbone, la fumée, GPL, etc.) et sur le comportement des personnes se trouvant au domicile. Le système peut analyser les données collectées par le capteur de gaz et lance l'alerte dans les cas anormaux (le taux actuel d'un gaz dépasse 300 PPM, une chute d'une personne, etc.). De plus, le système réagira de façon automatique et autonome en effectuant une liste d'actions préconfigurées pour remettre les taux actuels des gaz à leur bonnes valeurs, tout en envoyant des notifications aux utilisateurs (via SMS/Appel et une application Web responsive) qui peuvent eux même (à distance à travers Internet) intervenir.

#### **2.2.2. Besoins techniques**

Afin de mettre en place une solution adéquate aux attentes des utilisateurs, notre application doit nécessairement assurer les principaux besoins techniques suivants :

- **La sécurité** : l'application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde, c'est-à-dire que le site web est accessible par un identifiant et un mot de passe attribué à une personne physique.
- **L'interface** : avoir une application qui respecte les principes des interfaces Homme/Machine (IHM), ainsi qu'elle doit être responsive aux différentes tailles d'écran.
- **La portabilité et la compatibilité** : Notre application doit être portable sur tous les environnements logiciels (Windows, Mac OS, Linux), d'où compatible avec les autres services développés ou qui vont être développés.
- **Gestion des utilisateurs et droits d'accès** : on peut ajouter des utilisateurs avec différents droit d'accès.

## 2.3. Conception

Après avoir défini les fonctionnalités de notre système, il est indispensable de passer par la suite vers la conception qui s'intéresse à comment ces fonctionnalités seront implémentées et représente ainsi une ébauche de l'activité de développement. La conception passe par deux étapes : la conception software et la conception des composants électroniques (hardware).

### 2.3.1. La conception software

#### 2.3.1.1. Identification des acteurs du système

- **Administrateur distant** : est le responsable des différents équipements, Il a toutes les fonctionnalités du système telles que : la supervision, gestion des maison, gestion des alertes ainsi que la gestion des clients, il peut aussi gérer les différents module ESP32 des clients (ajouter, modifier, supprimer).
- **Client** : peut consulter les taux actuels des gaz présents dans l'environnement, leur historique de changements dans le temps, il peut aussi consulter l'historique des chutes, modifier le mode du système entre manuel ou automatique et peut recevoir des alertes sur différents évènements (une chute, une fuite de gaz, une intrusion).

#### 2.3.1.2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est interprété comme une suite de séquences d'actions produisant un résultat observable, c'est un diagramme qui sert à documenter le système étudié, définir ses limites et modéliser les besoins, et tout cela en dénombrant :

- Les fonctions majeures assurées par le système.
- L'ensemble des acteurs interagissent avec le système.

➤ Diagramme de cas d'utilisation global



Figure 65 Diagramme de cas d'utilisation global



### 2.3.1.3. Diagrammes de séquence

Le diagramme de séquence permet de répondre à la question : comment les éléments du système interagissent entre eux au cours du temps ? Il désigne une représentation séquentielle du déroulement des traitements et des interactions entre les acteurs du système étudié. Dans ce diagramme nous avons deux types d'interactions, la première étant l'interaction objet avec le système qui se fera à travers d'échanges de messages, et la deuxième est l'interaction acteurs système qui se fera dans ce cas par l'interaction de l'acteur avec l'interface.

➤ **Diagramme de séquence du cas Vérifier la connectivité des cartes ESP32**

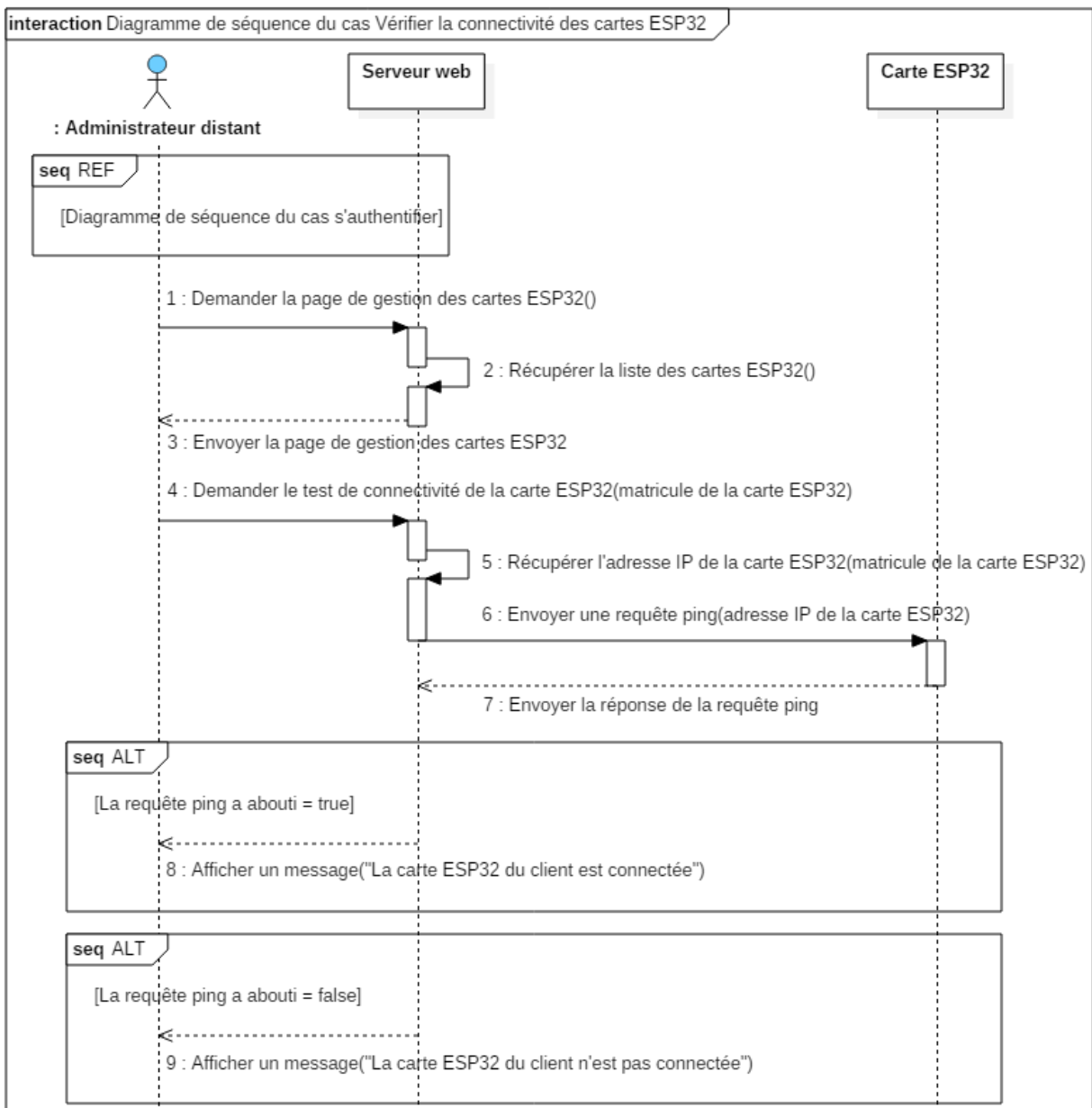
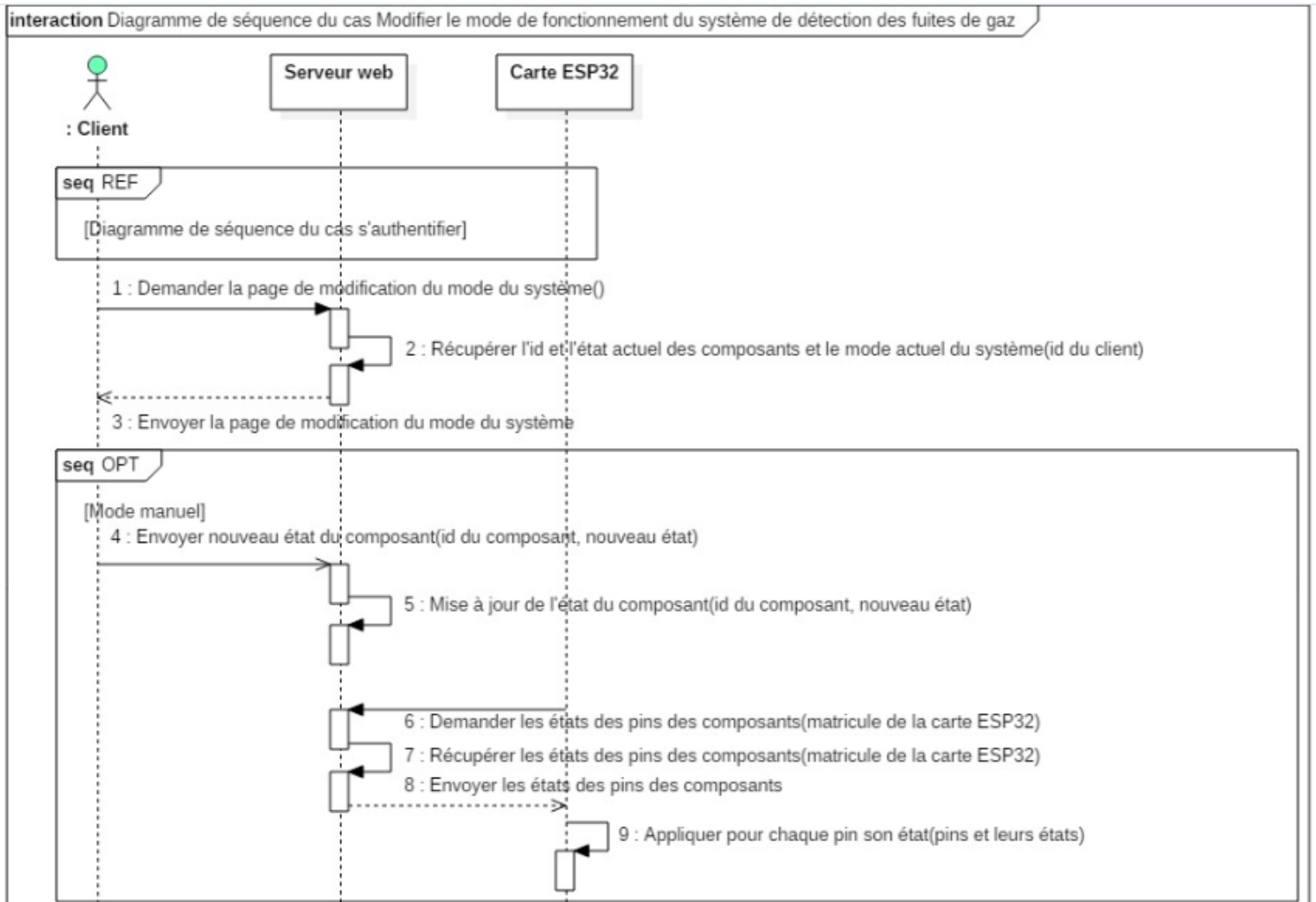


Figure 66 Diagramme de séquence du cas Vérifier la connectivité des cartes ESP32

➤ **Diagramme de séquence du cas Modifier le mode de fonctionnement du système de détection des fuites de gaz**



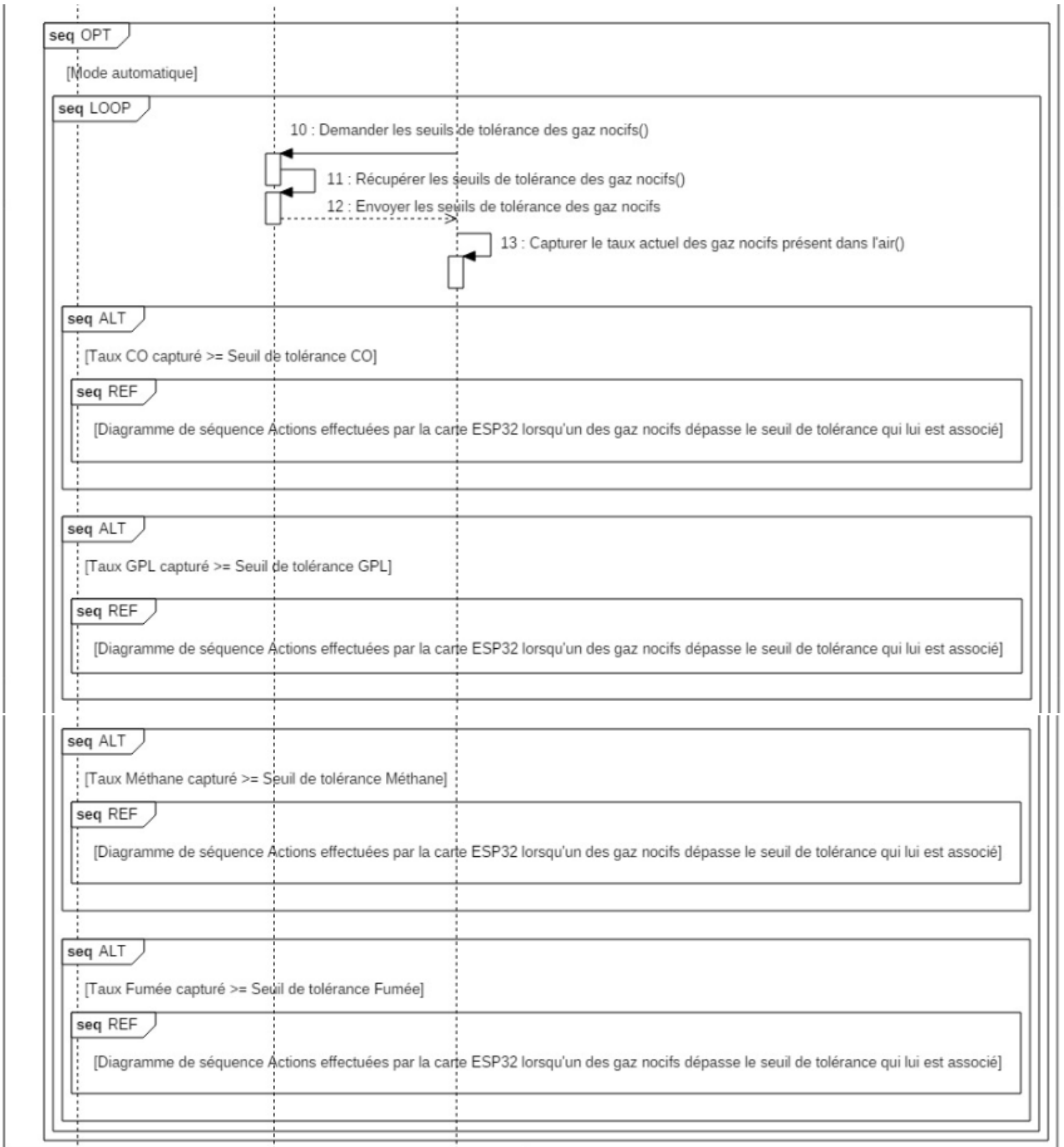


Figure 67 Diagramme de séquence du cas Modifier le mode de fonctionnement du système de détection des fuites de gaz

### 2.3.1.4. Diagramme de classes

Le diagramme de classes est considéré comme étant le diagramme le plus important dans le domaine de la modélisation orientée objet, c'est un diagramme structurel statique permettant de représenter les classes et les associations qui les lient, il fournit une vue globale de la structure interne du système. Une classe contenue dans ce diagramme est une représentation abstraite d'un ensemble d'objets, elle contient les attributs et les méthodes nécessaires à leurs existences.

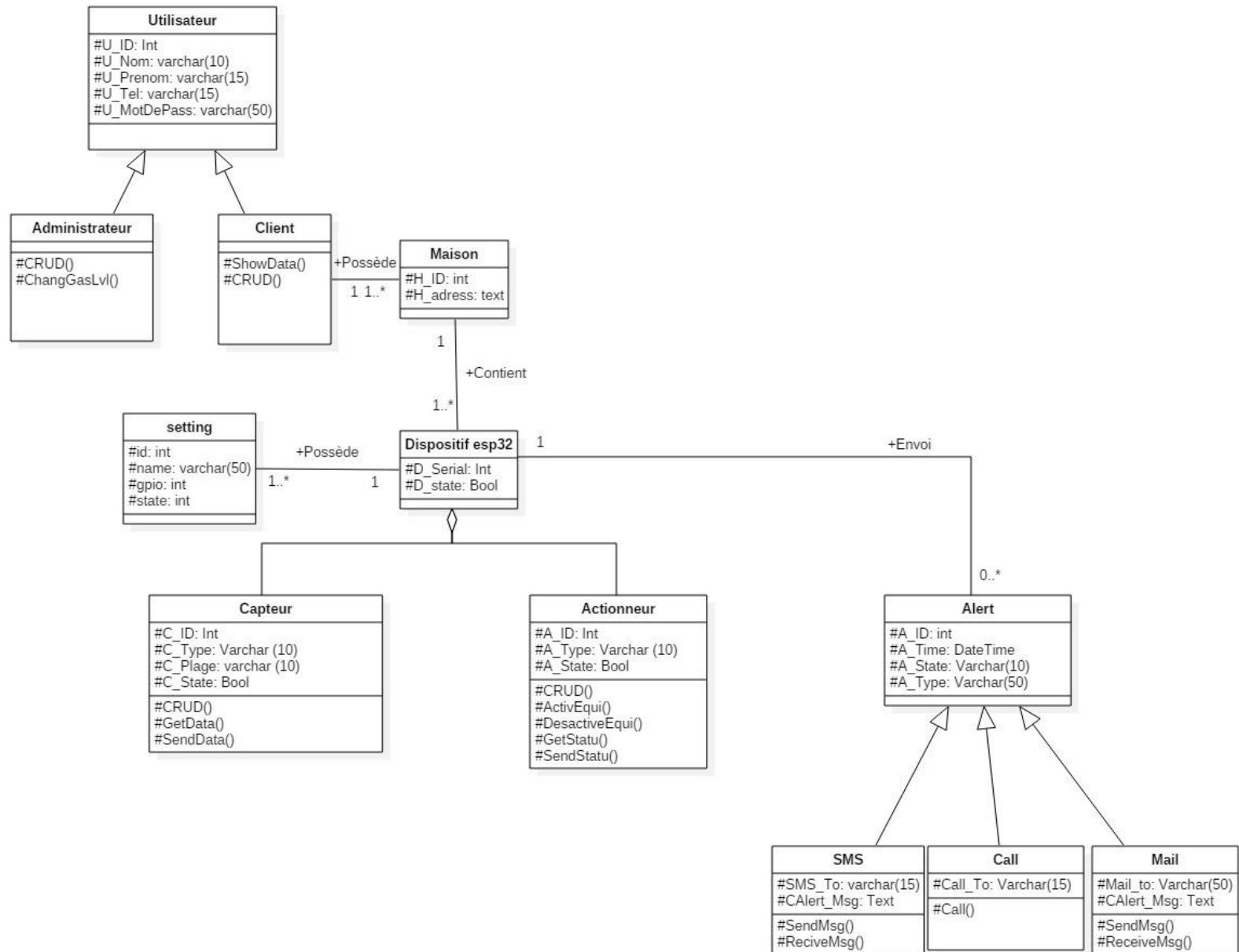


Figure 68 Diagramme de classes

➤ **La classe Utilisateur**

C'est la classe qui stocke les informations des comptes utilisateurs, pour leur permettre ensuite d'accéder et de gérer le système.

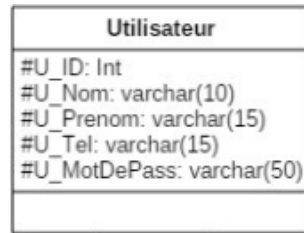


Figure 69 La classe Utilisateur

➤ **La classe Setting**

C'est la classe responsable du stockage des paramètres des cartes ESP32, pour pouvoir par la suite contrôler les différents actionneurs.

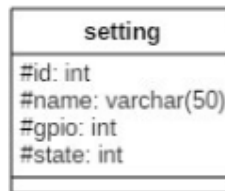


Figure 70 La classe Setting

➤ **La classe Alert**

C'est la classe responsable du stockage des alertes résultantes d'un évènement de fuites de gaz nocifs, chutes ou intrusions.

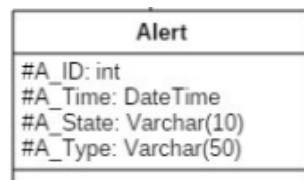


Figure 71 La classe Alert

### 2.3.1.5. Diagramme de composants

Le diagramme de composants permet de décrire les composants du système et les relations qui existent entre eux.

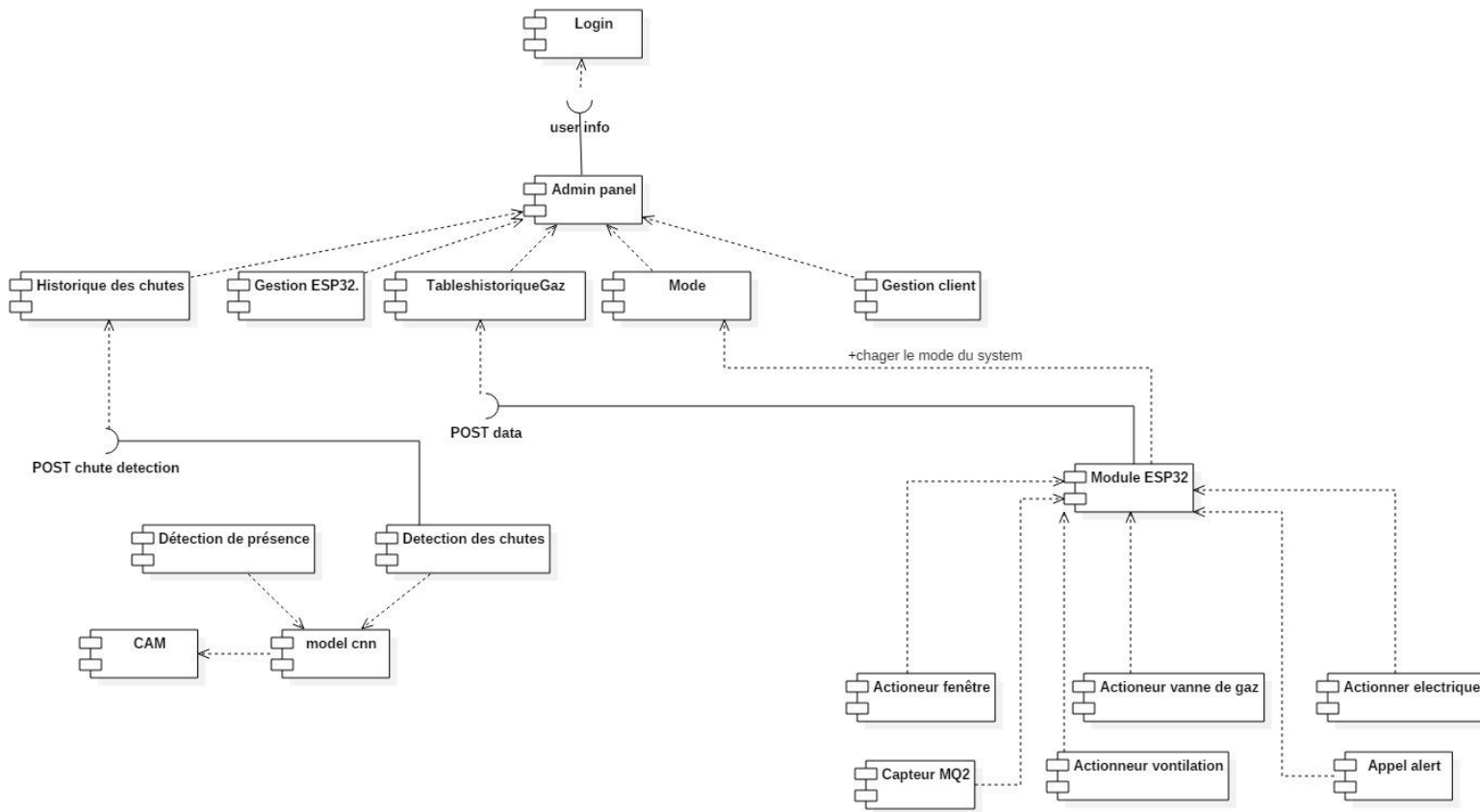


Figure 72 Diagramme de composants

## 2.3.2. Le déploiement du système

### 2.3.2.1. Architecture du système

Nous présentons ci-dessous, une architecture de notre système où on peut voir le réseau de capteurs qui va transmettre les données acquises à un centre de traitement composé d'un ou plusieurs ordinateurs connectés à Intranet (ou à Internet). L'analyse des données permettrait d'élaborer des décisions. Ces décisions seront appliquées soient automatiquement, semi automatiquement ou de manière manuelle. Le centre de traitement permettrait aux techniciens et responsables de suivre en temps réel et à distance le domicile de l'occupant. Il leur permettra aussi de prendre des décisions sur la base d'informations fournies par le centre de traitement. L'interaction du centre de traitement avec les responsables pourrait se faire en utilisant internet, ou Intranet ou les services d'appel d'un opérateur de télécommunication. A titre d'exemple une alerte pourrait être envoyée au responsable par appel. Le schéma synoptique suivant va nous permettre de mieux comprendre le fonctionnement global de notre système.

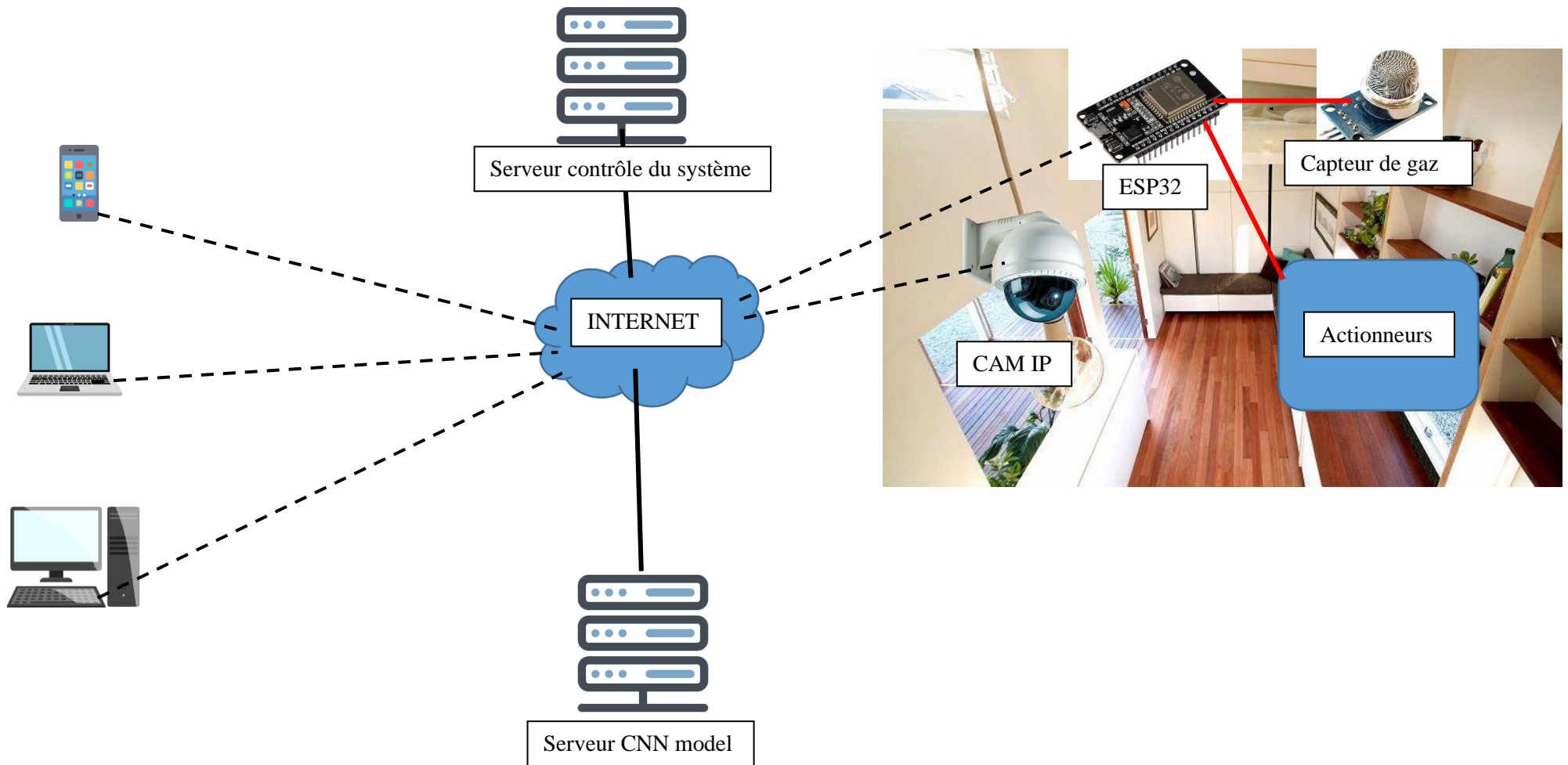


Figure 73 Architecture du système

### 2.3.2.2. La conception des composants électronique (hardware)

Dans ce qui suit nous allons décrire les éléments de la conception de notre système (capteurs, microcontrôleur, actionneur) et cela en utilisant les outils de conception électronique (fritzing). En effet, nous avons utilisé ce logiciel afin de mieux visualiser le bon déroulement du système ainsi que d'avoir une idée claire sur la partie matérielle et la conception des circuits imprimés avant de les mettre en place dans notre installation.

➤ **Schéma du système de détection des fuites de gaz nocifs et de la fumée**

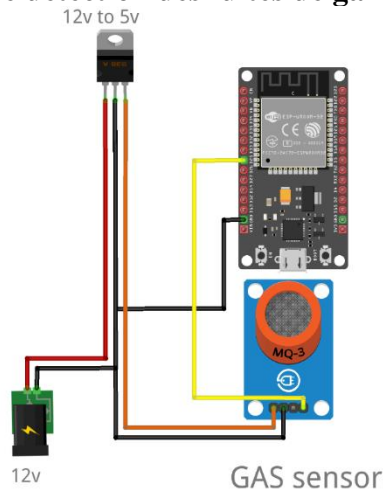


Figure 74 Schéma du système de détection des fuites de gaz toxiques

Le schéma de la figure 78 représente le branchement du capteur MQ-2 qui peut détecter différents gaz comme le monoxyde de carbone, GPL, l'i-butane, le propane, le méthane ainsi que la fumée, il se compose de 4 branchements le premier de couleur orange pour le 5V (+), le deuxième de couleur noire pour le GND (-), le jaune pour « DATA analogique », celui qui reste pour « DATA numérique ».

➤ **Schéma du système de contrôle de la vanne principale de gaz**

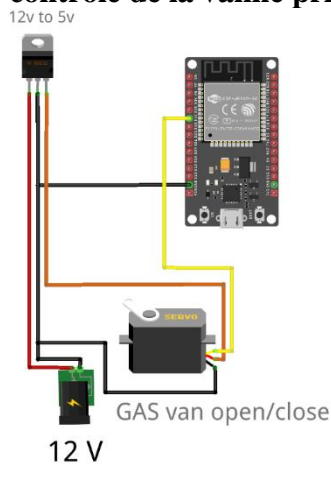


Figure 75 Schéma du système de contrôle de la vanne principale de gaz

Le schéma de la figure 79 représente le branchement du servomoteur qui va contrôler la vanne principale de gaz à l'aide de la carte ESP32, il comporte 3 connections, le rouge et la noire pour l'alimentation, le jaune pour le contrôle.



➤ Schéma du système de contrôle du courant électrique

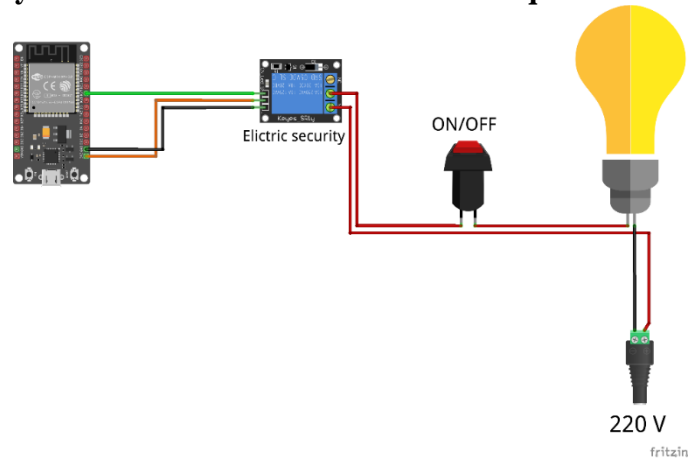


Figure 76 Schéma du système de contrôle de l'alimentation électrique

➤ Schéma du système d'envoi des alertes via SMS/Appel

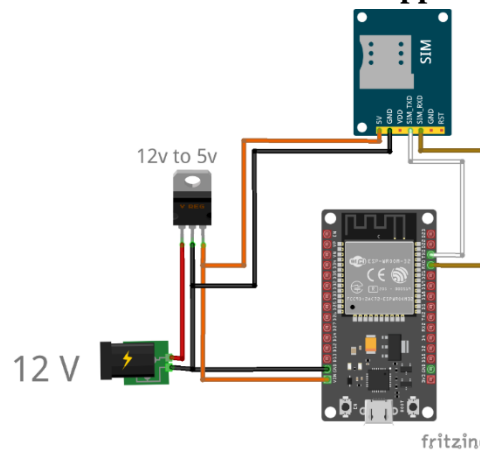


Figure 77 Schéma du système d'envoi des alertes via SMS/Appel

Le schéma de la figure 81 représente le branchement du Module GSM SIM800L avec la carte ESP32. Il se compose de 5 branchements les deux premiers de couleurs blanc et Jaune pour « DATA », le restes est pour l'alimentation « +, - ».

➤ Schéma du système de contrôle de l'alarme sonore

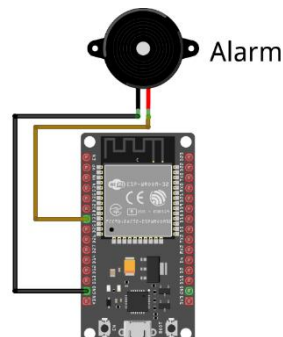


Figure 78 Schéma du système de contrôle de l'alarme sonore

Le schéma de la figure 82 représente le branchement de l'alarme sonore de type Piezo Buzzer avec la carte ESP32, le fil rouge sera connecté avec l'un des pins de l'ESP32 pour le control de l'état ON/OFF, le fil noir pour le GND.

➤ Schéma du système de contrôle des fenêtres et du ventilateur

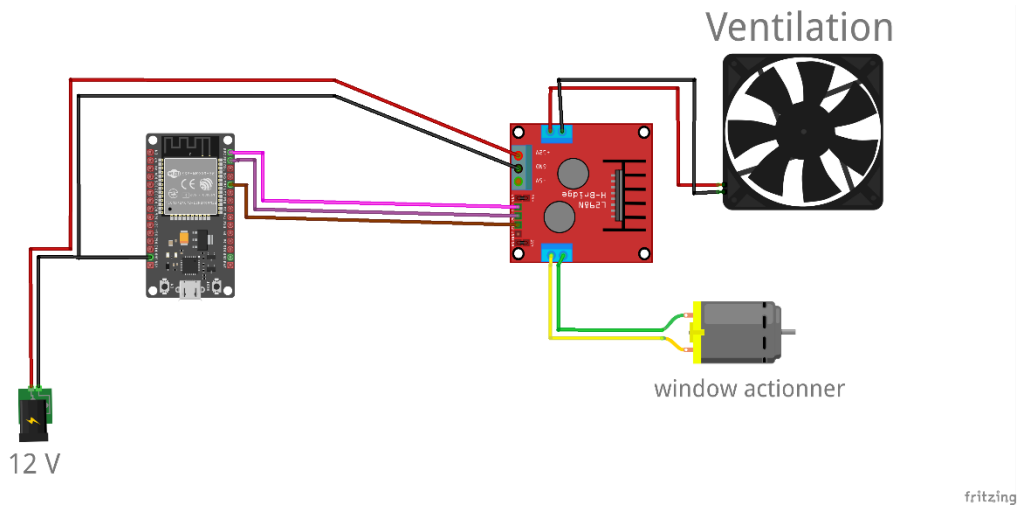


Figure 79 Schéma du système de contrôle des fenêtres et du ventilateur

Le schéma de la figure 83 représente le branchement de l'actionneur qui contrôle les fenêtres, il est représenté par le petit moteur qui tourne dans les deux sens, les fils de couleur rouge et noire pour l'alimentation et les fils de couleur mauve et rose pour contrôler le moteur des fenêtres et le fil gris pour contrôler la ventilation.

➤ Vue globale du système

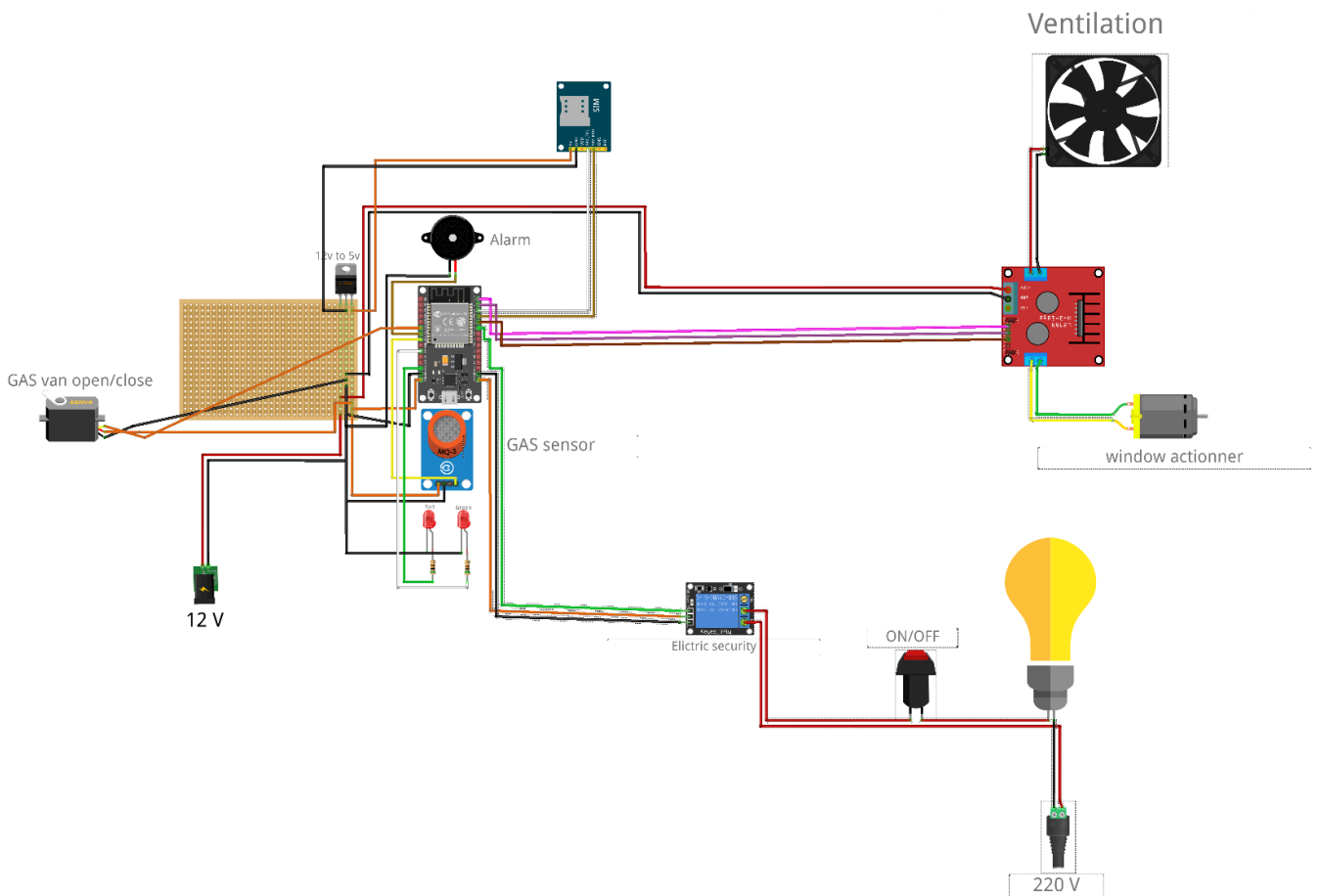


Figure 80 Le branchement du système global avec ses différents composants

➤ **Résumé des pins utilisés dans la carte ESP32**

Composant	Numéro du pin	Alimentation
MQ-2	34	5V
Servomoteur	32	6V
Relay	18	3.3V
LED 1	26	3.3V
LED 2	27	3.3V
Piezo buzzer	33	3.3V
Moteur 1	23/22	12V
GSM	04/02	5V

Tableau 8 Les pins utilisés dans la carte ESP32

### 3. Conception du système de détection des chutes

#### 3.1. Le principe de fonctionnement du système de détection des chutes

Nous proposons une nouvelle approche pour la détection des chutes basée sur le *framework Mask R-CNN* (*Mask Region-based Convolutional Neural Network*) et les réseaux de neurones convolutif. Selon nos recherches effectuées (étude de l'existant), nous sommes les premiers au niveau internationale à élaborer un système de détection des chutes qui se base principalement sur le *framework Mask R-CNN*. Depuis une séquence vidéo enregistrée par une caméra, la silhouette du corps humain est d'abord obtenue en appliquant le *framework Mask R-CNN*, puis, elle est mise dans une image sans arrière-plan (une image avec un fond noire où l'arrière-plan de la pièce où se trouve la personne est supprimé gardant ainsi que la silhouette) pour assurer d'une part la confidentialité de la personne, mais aussi d'éliminer les traitements inutiles pour le réseau neuronal convolutif qui va s'appuyer uniquement sur la forme de la silhouette (quel que soit l'environnement où elle se trouve) pour extraire un ensemble de *features* permettant d'identifier la classe à laquelle appartient la posture courante. La silhouette obtenue correspondant aux activités quotidiennes est introduite dans le réseau de neurones convolutif qui est appliqué pour la classification de la posture courante exercé par la personne. Deux classes sont possibles allongée *lie* ou n'est pas allongée *no lie*, si la personne est debout ou assise ou elle se penche alors la classe qui doit être prédite par le réseau neuronal convolutif est *no lie*, sinon, si la personne est allongée alors la classe qui sera prédite est *lie*. Un évènement de chute est détecté lorsqu'une personne est allongée (le modèle classe l'ensemble des images obtenues (par l'étape d'extraction de la silhouette du corps humain) pendant une durée de 30 secondes dans la classe *lie*) dans la région représentant le sol pendant une période donnée (30 secondes). Lors de la détection d'une chute, une alarme sonore et visuelle sera déclenchée au niveau du domicile de la personne ayant chuté et une alerte via un Appel/SMS est envoyée aux contacts (04 numéro de téléphones) introduits préalablement dans le système ainsi qu'aux services de la protection civile, de plus une notification est envoyée à l'application Web/Mobile (utilisée par un administrateur distant appartenant aux services de la protection civile et les occupants du domicile).

De plus, ce système offre à l'utilisateur la possibilité de choisir entre deux modes : mode détection des chutes qu'on vient d'expliquer précédemment et le mode surveillance qui permet la détection des intrusions de personnes malveillantes au domicile de l'utilisateur. Par exemple, si l'utilisateur ou l'un de ces proches est présent au domicile, il opte pour le mode détection des chutes, sinon, s'il n'y a aucune présence, l'utilisateur peut déployer le mode surveillance. Une intrusion est détectée lorsque le *framework Mask R-CNN* détecte la présence d'une personne or qu'aucun occupant du domicile n'y est présent. Les deux figures 85 et 86 illustrent parfaitement le principe de fonctionnement de ces deux modes.

L'autre avantage de notre système (grâce à l'utilisation du *framework Mask R-CNN*), est qu'il peut assurer ces tâches de jour comme de nuit (l'utilisation d'une caméra infrarouge).

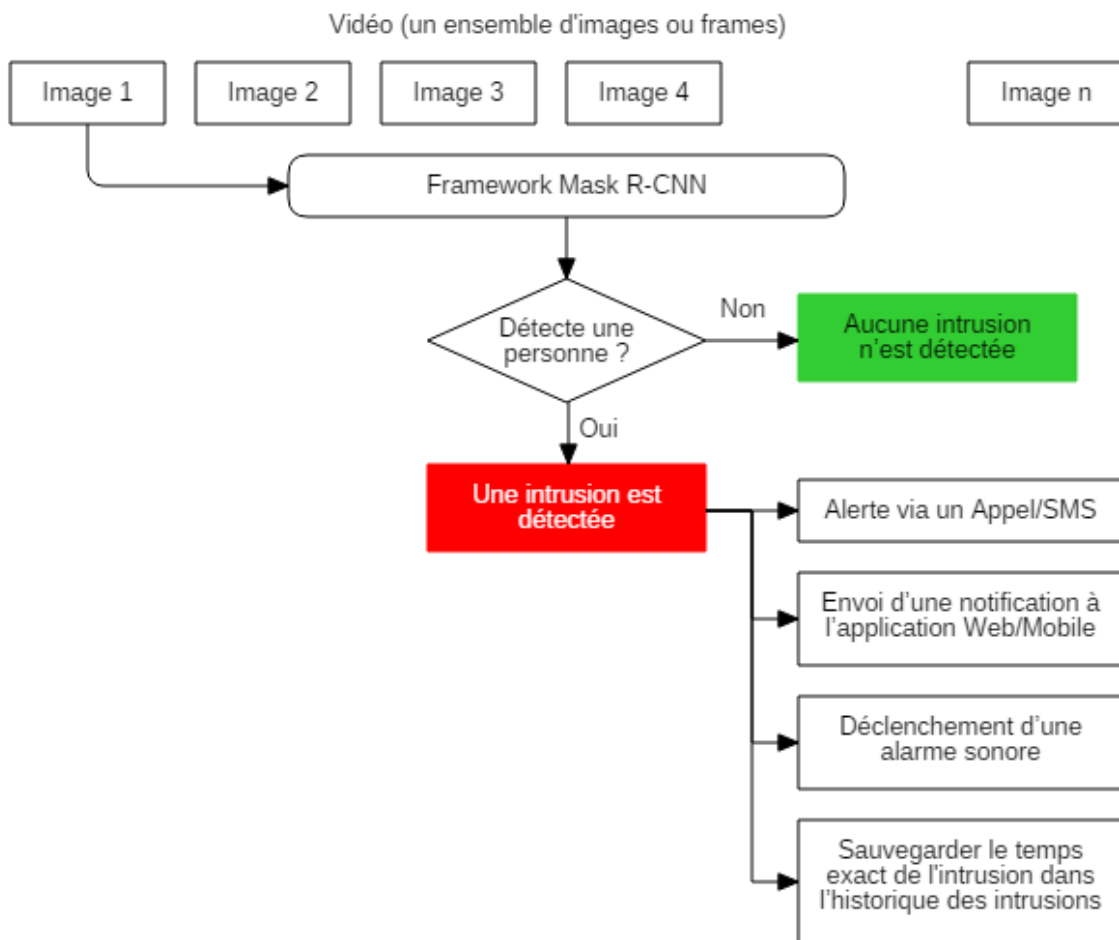


Figure 81 Diagramme expliquant d'une manière générale le principe de fonctionnement du mode surveillance

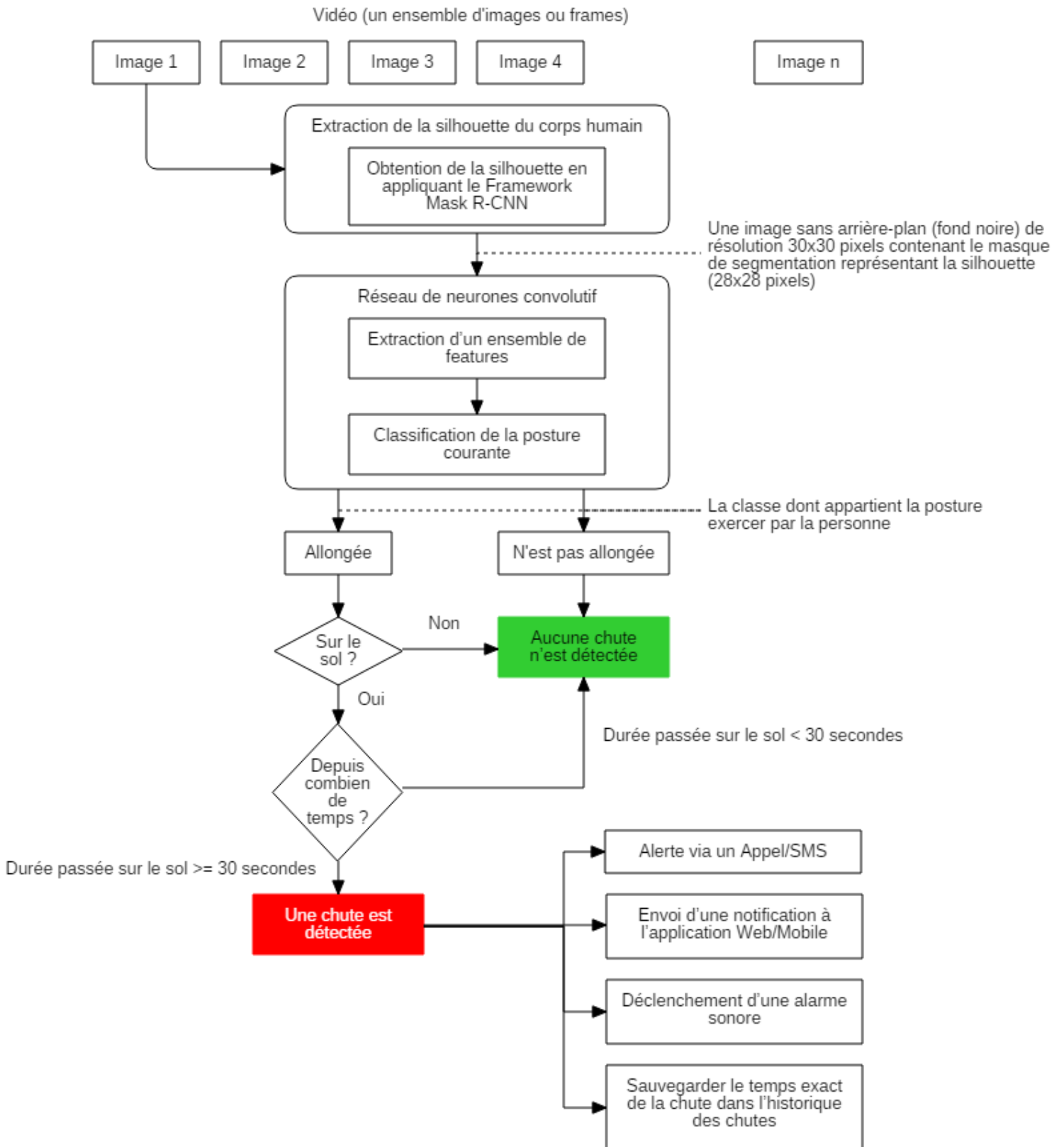


Figure 82 Diagramme expliquant d'une manière générale le principe de fonctionnement du mode détection des chutes

## 3.2. Extraction de la silhouette du corps humain

### 3.2.1. Obtention de la silhouette en utilisant le Framework Mask R-CNN

La forme de la silhouette du corps humain représente un point clé pour déterminer la classe (*no lie* ou *lie*) à laquelle appartient la posture courante, pour pouvoir l'obtenir on doit utiliser des techniques de segmentation d'image et plus précisément la segmentation d'instance, qui prédit et dessine un masque épousant la forme de chaque objet détecté dans l'image. Pour chaque objet on assigne une couleur différente pour le masque, contrairement à la segmentation sémantique où les objets faisant partie de la même classe sont considérés comme une seule instance et auront la même couleur de masque (par exemple : tous les objets appartenant à la classe personne auront un masque de segmentation de couleur verte).

Comme on l'a constaté dans le chapitre 3, le *framework Mask R-CNN* est le plus adapté pour réaliser cette tâche (la segmentation d'instance), contrairement aux autres méthodes : *R-CNN*, *Fast R-CNN* et *Faster R-CNN*, qui sont spécialisés uniquement dans la détection d'objets où ils peuvent seulement localiser des objets dans une image à l'aide d'un cadre de délimitation (*bounding box*) et leur attribuer une étiquette de classe, sans fournir d'information sur la forme de l'objet (ils ne dessinent pas de masque de segmentation).

Pour une image donnée, *Mask R-CNN* génère un ensemble de régions d'intérêt (zones contenant un objet), chaque région est délimitée par un cadre qui encapsule l'objet qu'elle contient, puis le *framework* utilise un modèle pré-entraîné sur le jeu de données *COCO* pour prédire la classe à laquelle appartient l'objet et générer un masque de segmentation à basse résolution 28x28 pixels qui ensuite est adapté à la taille du cadre de délimitation représentant la région d'intérêt, obtenant ainsi un masque qui épouse parfaitement la forme de l'objet. *Mask R-CNN* peut faire la détection et la segmentation de 80 types d'objets (personne, vélo, voiture, chaise, livre, etc.), cependant, on a apporté des modifications à celui-ci pour qu'il détecte et segmente seulement les objets appartenant à la classe personne qui est notre sujet d'intérêt.

#### L'entrée (une image issue de la séquence vidéo)



Générer les régions d'intérêt (délimitées par un cadre de délimitation) contenant un objet (personne)



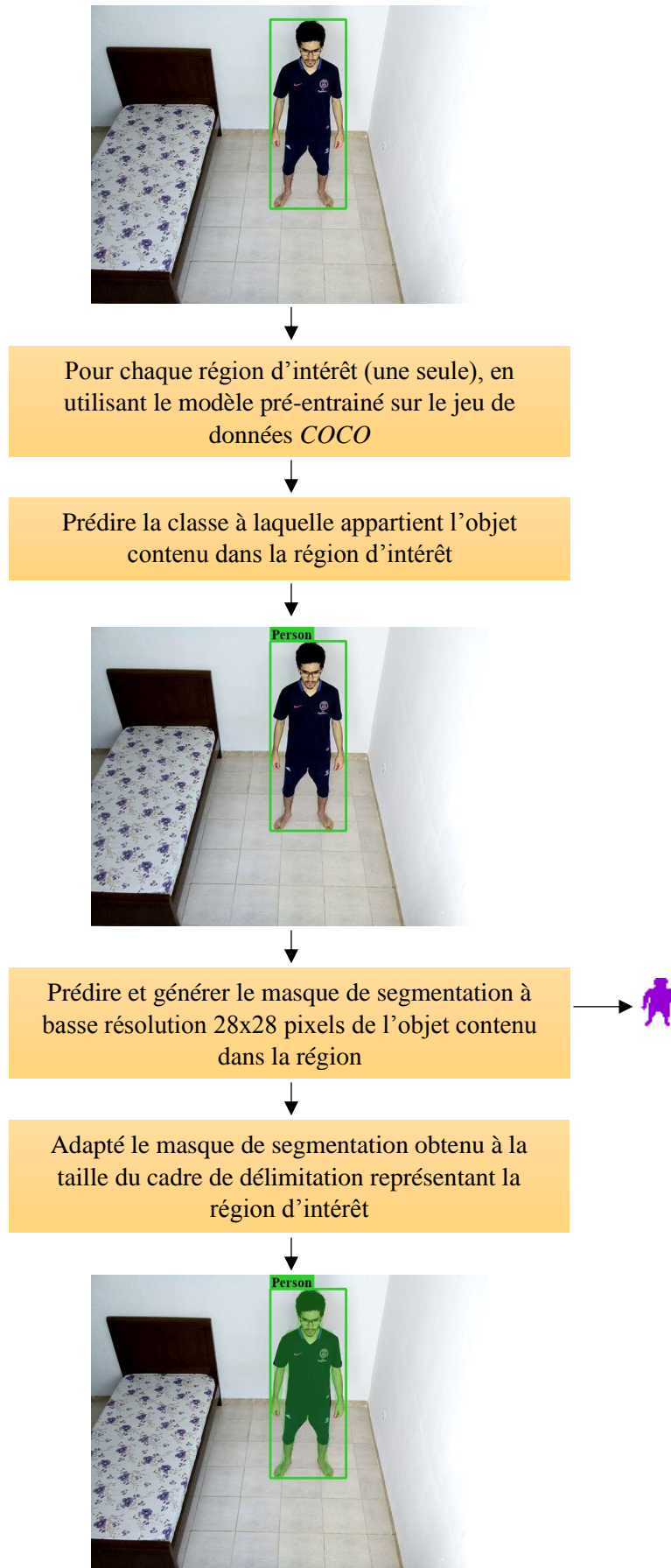


Figure 83 Illustration du principe de fonctionnement du Framework Mask R-CNN

Depuis la figure ci-dessus on peut constater qu'en utilisant le Framework *Mask R-CNN* et via le processus de segmentation d'instance, on peut facilement obtenir la forme de la silhouette d'une personne, qui va nous permettre par la suite d'identifier si elle est allongée ou non. Pour cela nous allons appliquer un réseau de neurones convolutif qui reçoit comme entrée une image de résolution 30x30 pixels contenant le masque de segmentation 28x28 pixels prédit et généré précédemment par le modèle pré-entraîné sur le jeu de données *COCO*. Le réseau va classifier l'image reçue en entrée dans l'une des deux classes : *non lie* ou *lie*.

L'image de résolution 30x30 pixels est obtenue simplement en insérant le masque de segmentation 28x28 pixels dans une image sans arrière-plan (fond noir) 30x30 pixels.



Figure 84 Insertion du masque de segmentation 28x28 pixels représentant la silhouette d'une personne dans une image sans arrière-plan 30x30 pixels

### 3.3. Le réseau de neurones convolutif

Voici le réseau de neurones convolutif que nous avons conçu :

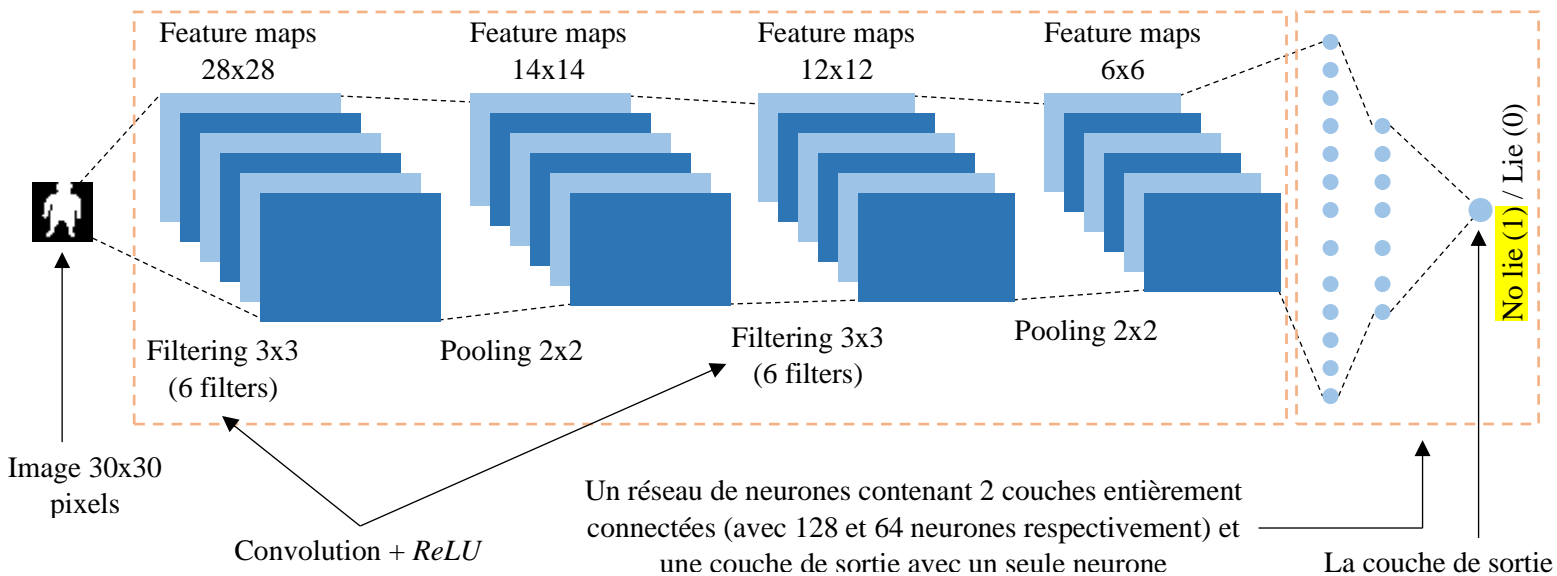


Figure 85 La structure du réseau neuronal convolutif développé

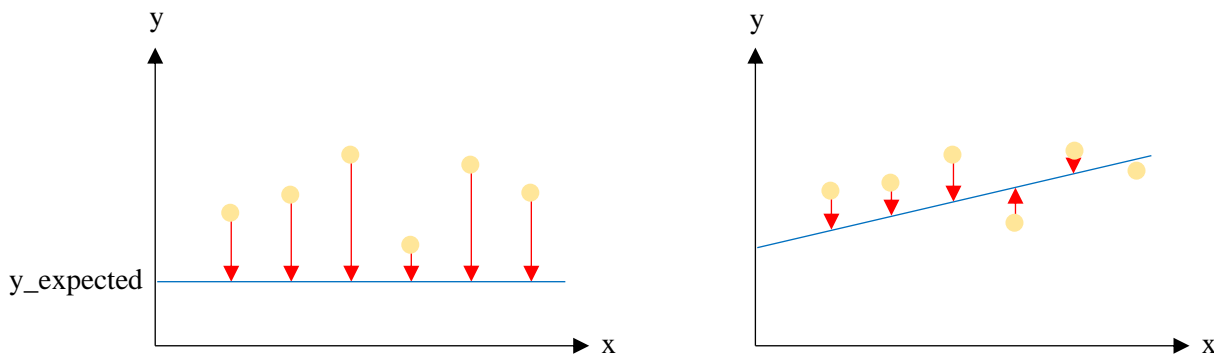
#### 3.3.1. La fonction de perte (loss function)

Pour un modèle, l'apprentissage signifie déterminer les bonnes valeurs pour toutes les pondérations (poids) à partir d'exemples étiquetés. Dans l'apprentissage supervisé, un algorithme de *Machine Learning* crée un modèle en examinant de nombreux exemples, puis en tentant de trouver un modèle qui minimise la perte. La perte correspond à la pénalité pour une mauvaise prédiction. Autrement dit, la perte est un nombre qui indique la médiocrité de la prévision du modèle pour un exemple donné. Si la prédiction du modèle est parfaite, la perte est nulle. Sinon, la perte est supérieure à zéro. Le but de



l'entraînement d'un modèle est de trouver un ensemble de pondérations pour lesquels la perte, en moyenne sur tous les exemples, est faible. Par exemple, la figure 90 présente à gauche un modèle dont la perte est élevée, et à droite un modèle dont la perte est faible.

- Les flèches en rouge représentent les pertes (la différence entre la sortie attendue ou l'observation et la sortie réelle qui est la sortie actuelle).
- La ligne en bleu représente la sortie attendue.



**Figure 86** La perte est élevée dans le modèle de gauche par contre elle est faible dans le modèle de droite.

La perte est calculée par une fonction appelée fonction de perte ou *loss function*. Une fonction de perte est pour un seul exemple d'entraînement, c'est la différence entre la sortie attendue et la sortie réelle (c'est l'erreur), cette fonction est définie par la formule suivante :  $loss = |y\_expected - y|$ .

D'autre part, une fonction de coût (*cost function*), est la perte moyenne sur l'ensemble du jeu de données (somme des pertes de chaque exemple que l'on divise par le nombre total d'exemples). Elle est définie par la formule suivante :  $cost = \frac{1}{n} \sum_{i=1}^n |y\_expected - y|$ , où  $n$  représente le nombre d'exemple d'entraînement dans un jeu de données. Les stratégies d'optimisation visent à minimiser la fonction de coût.

La fonction de perte que nous avons utilisé lors de la phase d'entraînement de notre réseau de neurones convolutif est appelée *binary\_crossentropy*, qui est couramment utilisée pour la classification binaire où on a seulement deux classes possibles (*lie* et *no lie*), elle est calculée après l'application de la fonction d'activation *sigmoid* dans la dernière couche du réseau de neurones artificiels. La fonction *binary\_crossentropy* est définie par la formule suivante :  $BCE = -\sum_{i=1}^2 p(x) \times \log(q(x)) = -p(lie) \times \log(q(lie)) - (1 - p(lie)) \times \log(1 - q(lie))$ , où  $x$  représente une classe donnée (*lie* ou *no lie*),  $p(x)$  représente la probabilité de la classe  $x$  avant l'application de la fonction *sigmoid* et  $q(x)$  après son application.

### 3.3.2. L'algorithme d'optimisation

Nous avons précédemment traité la fonction de perte, qui est un moyen mathématique pour mesurer à quel point nos prédictions sont erronées. Au cours du processus d'entraînement, nous ajustons et

modifions les paramètres (poids) de notre modèle pour essayer de minimiser cette fonction de perte et rendre nos prédictions aussi correctes et optimisées que possible.

Mais comment faisons-nous cela ? Comment modifions-nous les paramètres de notre modèle, de combien et quand ? C'est là qu'interviennent les optimiseurs (*optimizers*). Ils relient la fonction de perte et les paramètres du modèle en mettant à jour le modèle en réponse à la sortie de la fonction de perte. En termes plus simples, les optimiseurs façonnent et moulent notre modèle dans sa forme la plus précise possible en modifiant les poids. La fonction de perte est le guide du terrain, indiquant à l'optimiseur lorsqu'il se déplace dans la bonne ou la mauvaise direction.

Pour un modèle mental utile, nous pouvons penser à un randonneur essayant de descendre une montagne avec un bandeau sur les yeux. Il est impossible de savoir dans quelle direction aller, mais il y a une chose qu'il peut savoir : s'il descend (fait des progrès) ou monte (perd des progrès). Finalement, s'il continue de prendre des mesures qui le mènent vers le bas, il atteindra la base. De même, il est impossible de savoir dès le départ quels devraient être les poids de notre modèle. Mais avec quelques essais et erreurs basés sur la fonction de perte, nous pouvons finir par y arriver éventuellement.

L'optimiseur que nous avons utilisé pour minimiser la fonction de perte (trouver les poids optimaux) et ainsi améliorer notre modèle est appelé *Adam (Adaptive Moment Estimation)* avec les paramètres par défaut ( $learning\ rate = 0.001, beta1 = 0.9, beta2 = 0.999$  et  $epsilon = 10^{-8}$ ) comme indiqué dans le premier article ayant introduit cet algorithme [44].

### 3.3.3. Les métriques

Différentes métriques sont proposées pour mesurer les performances des modèles d'apprentissage en profondeur destinés à la tâche de classification, ces métriques sont : *accuracy, precision, recall, f1-score* ou *f-measure* et *loss*.

Chaque *epoch* le réseau de neurones convolutif est entraîné sur un ensemble d'exemples d'entraînement ou *training set*. Une fois entraîné, il est évalué sur un ensemble de données de validation (*validation set*) contenu dans le jeu de données (à la fin de chaque *epoch*), c'est à ce moment-là que les métriques sont calculées. Donc à la fin de chaque *epoch* on obtient de nouvelles valeurs pour les différentes métriques, le résultat final est la somme des valeurs obtenues divisée par le nombre d'*epoch*. Pour pouvoir calculer ces métriques, on doit d'abord ressortir la matrice de confusion ou *Confusion matrix*.

- *Confusion matrix* : une matrice de confusion est un tableau qui est souvent utilisé pour décrire les performances d'un modèle de classification (ou classificateur) évalué sur un ensemble de données de validation dont la classe d'appartenance est déjà connue. La ligne représente la classe attendue (devant être prédite par le modèle) et la colonne représente la classe prédite. Depuis la matrice de confusion, on peut calculer les métriques : *accuracy, precision, recall, f-*

*measure* et *loss*. Avant cela on doit aborder 4 termes essentiels pour pouvoir remplir la matrice de confusion. Pour chaque classe (*No lie* et *Lie*) on calcule le :

- Vrais positif (*True positive* ou *TP*) : le nombre d'exemples dans lesquels la classe attendue *No lie* est également celle prédite par le modèle (prédictions correctes).
- Vrais négatif (*True negative* ou *TN*) : le nombre d'exemples dans lesquels la classe attendue *Lie* est également celle prédite par le modèle (prédictions correctes).
- Faux positif (*False positive* ou *FP*) : le nombre d'exemples dans lesquels la classe attendue *Lie* n'est pas celle prédite par le modèle *No lie* (prédictions fausses).
- Faux négatif (*False negative* ou *FN*) : le nombre d'exemples dans lesquels la classe attendue *No lie* n'est pas celle prédite par le modèle *Lie* (prédictions fausses).

		La classe prédite	
		No lie (1)	Lie (0)
La classe attendue	No lie (1)	TP	FN
	Lie (0)	FP	TN

**Tableau 9 Représentation de la matrice de confusion**

- *Accuracy* : il s'agit du rapport entre le nombre de prédictions correctes ( $TP + TN$ ) et le nombre total d'images contenues dans l'ensemble de données de validation ou *validation set*,  $Accuracy = \frac{TP+TN}{Total\ number\ of\ image\ on\ the\ validation\ set}$ . Une valeur élevée de cette métrique prouve que le classificateur arrive à prédire la classe correcte (attendue) pour la plupart des images contenues dans le *validation set*.
  - Si nous choisissons la mauvaise métrique pour évaluer les performances de notre modèle, nous sommes susceptibles de choisir un modèle médiocre ou, dans le pire des cas, d'être induit en erreur sur les performances attendues de notre modèle.
  - La métrique *accuracy* est inappropriée lorsque on a un jeu de données déséquilibré (*imbalanced dataset*) où il existe une distribution inégale des données (images) entre les classes (*lie* et *no lie*) du *training set*.
  - La classe majoritaire (*lie*) est la classe ayant le plus d'exemples dans l'ensemble de données d'entraînement par rapport à la classe minoritaire (*no lie*).
  - Pour les problèmes de classification déséquilibrée, la classe majoritaire est généralement appelée la classe négative (0) et la classe minoritaire est appelée la classe positive (1).
  - Une alternative à l'utilisation de cette métrique pour évaluer les performances d'un modèle lors d'une classification déséquilibrée, est de calculer les métriques *precision*, *recall* et *f-measure*.
- *Precision* : elle quantifie le nombre de prédictions correctes pour la classe positives (*no lie*), donc elle calcule l'*accuracy* de la classe minoritaire *no lie*. C'est le rapport entre le nombre d'exemples positives (dont la classe attendue est *no lie*) correctement prédits (la classe prédite

est également *no lie*) divisé par le nombre total d'exemples dans la classe *no lie*,  $Precision = \frac{TP}{TP+FP}$ .

- *Recall* : aussi appelée *sensitivity*, elle fournit une indication des prédictions positives manquées (où le modèle prédit la classe *lie* or c'est *no lie* qui devrait être prédite), elle quantifie le nombre de prédiction correctes à partir de toutes les prédictions positives qui auraient pu être faites,  $Recall = \frac{TP}{TP+FN}$ .
- *F-measure* : elle combine les deux métriques *Precision* et *Recall* en une seule.  $Fmeasure = 2 \times \frac{Precision \times Recall}{Precision+Recall}$ . Une valeur élevée de cette métrique prouve qu'on a moins de faux positifs (*FP*) et de faux négatifs (*FN*) donc moins de prédictions fausses faites par le modèle.
  - Cette métrique est idéale pour évaluer les performances de notre modèle vue qu'on a un *dataset* déséquilibré.
- *Loss* : c'est le rapport entre le nombre de prédictions fausses ( $FP + FN$ ) faites par le modèle, divisé par le nombre total d'images contenues dans l'ensemble de données de validation ou *validation set*,  $Loss = \frac{FP+FN}{Total\ number\ of\ image\ on\ the\ validation\ set} = 1 - Accuracy$ . Une valeur élevée de cette métrique prouve que le classificateur est médiocre (il n'arrive pas à prédire la classe correcte (attendue) pour la plupart des images du *validation set*).

### 3.3.4. Image data augmentation

Les performances des réseaux de neurones d'apprentissage en profondeur s'améliorent souvent avec la quantité de données disponibles dans le *dataset*. *Image data augmentation* est une technique qui peut être utilisée pour augmenter artificiellement la taille d'un jeu de données (l'ensemble d'exemples d'entraînement ou *training set*) sans acquérir de nouvelles images, en créant simplement des versions transformées des images dans le *dataset*. Les transformations comprennent une gamme d'opérations du domaine de la manipulation d'images, tels que le décalage (*shift*), le retournement (*flip*), la rotation, le zoom et bien plus encore. L'objectif est d'élargir l'ensemble de données d'entraînement avec de nouveaux exemples plausibles (selon le domaine d'application du modèle), cela signifie des variations d'images du *training set* susceptibles d'être vues par le modèle. Par exemple, un retournement horizontal (*horizontal flip* pour retourner une image vers la droite ou la gauche) d'une photo d'une personne peut avoir un sens, car la photo peut être prise de gauche ou de droite (ou la personne s'oriente vers la droite ou la gauche). Un retournement vertical (*vertical flip* pour retourner une image vers le bas ou le haut) et une rotation de la photo d'une personne n'ont pas de sens dans certains cas (si la personne est debout, assise ou penchée) étant donné que le modèle est très peu susceptible de voir une photo d'une personne à l'envers, par contre, ils peuvent être utiles dans le cas où la personne est allongée. Les figures 91, 92 et 93 illustrent trois transformations (un retournement horizontal, un retournement vertical et une rotation d'angle compris entre  $-180^\circ$  et  $180^\circ$ , pour des raisons de simplification on a illustré seulement une

rotation d'angles  $45^\circ$ ,  $90^\circ$  et  $130^\circ$ ) que nous avons appliqué sur les images de l'ensemble d'entraînement de notre *dataset* afin d'entraîner le modèle sur plus d'exemples qu'il peut rencontrer lors de son utilisation dans un milieu réel (reconnaitre des postures depuis une caméra de surveillance).

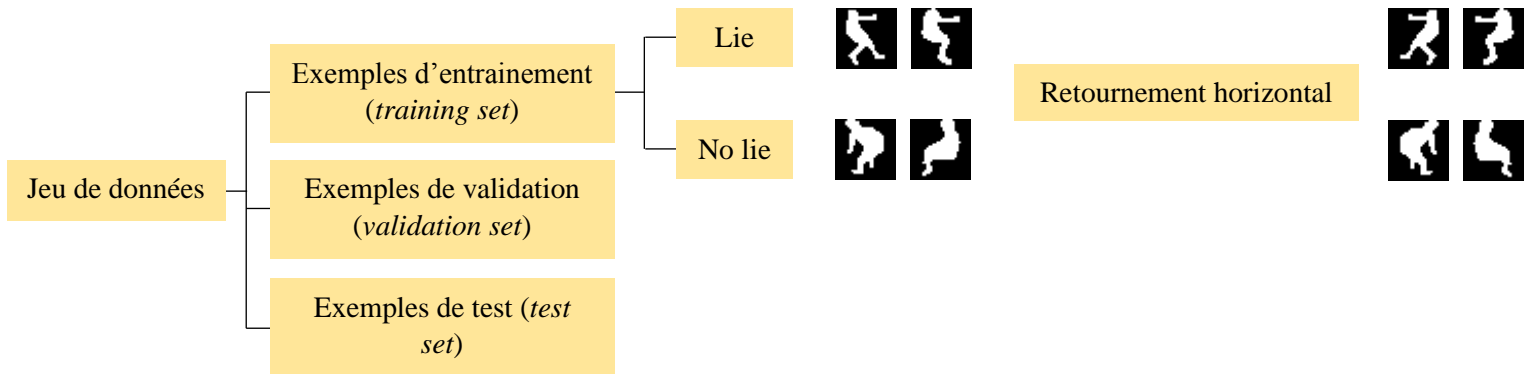


Figure 87 Application du retournement horizontal sur les images du training set

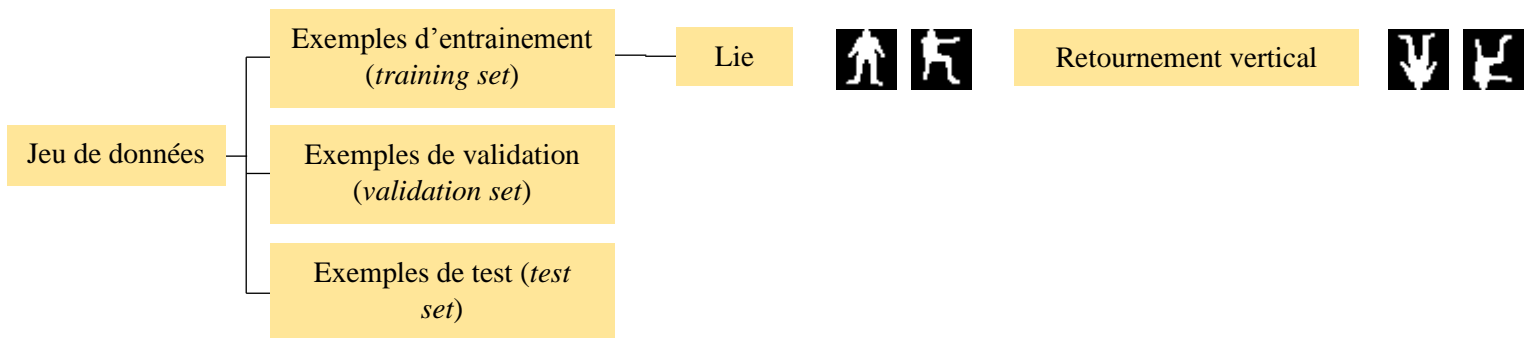


Figure 88 Application du retournement vertical sur les images du training set

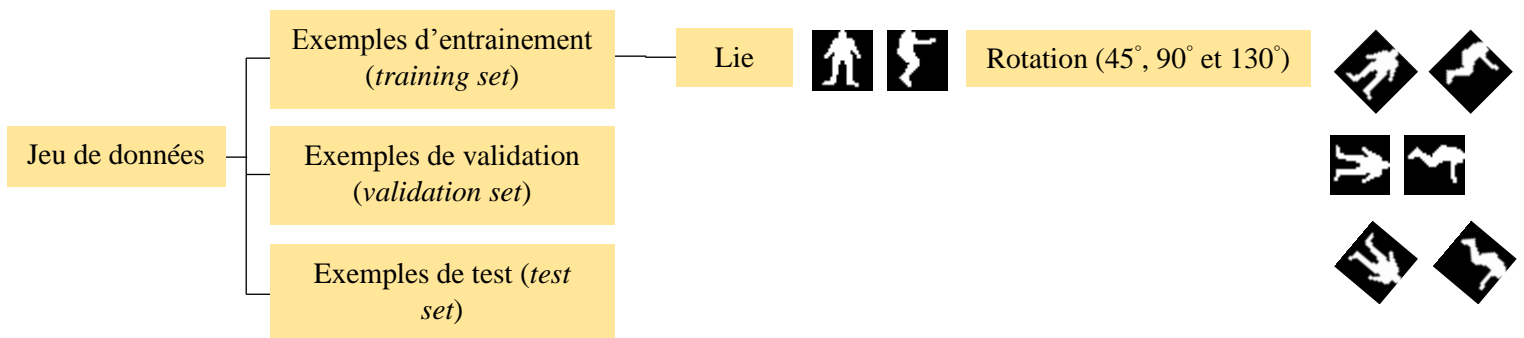


Figure 89 Application de la rotation sur les images du training set

### 3.4. Détermination de la zone représentant le sol

Cette zone constitue un point clé pour déterminer si une personne a chuté réellement, car, pour qu'un évènement de chute soit détecté, l'individu doit être allongé à l'intérieur de cette zone (pendant une durée de 30 secondes). Par contre, si la personne ne figure pas sur cette zone et qu'elle soit allongée (sur un tapis ou un lit), alors aucun évènement de chute n'est signalé. Pour la localiser, on a opté pour une méthode manuelle qui utilise la bibliothèque graphique *OpenCV* afin de dessiner sur chaque image ou

frame d'une vidéo un polygone (constitué de plusieurs points) représentant le sol. Ces points sont introduits dans le système par un administrateur distant une fois que la caméra de surveillance est installée au domicile de la personne.



Figure 90 La zone représentant le sol (polygone) définie par 04 points P1, P2, P3 et P4

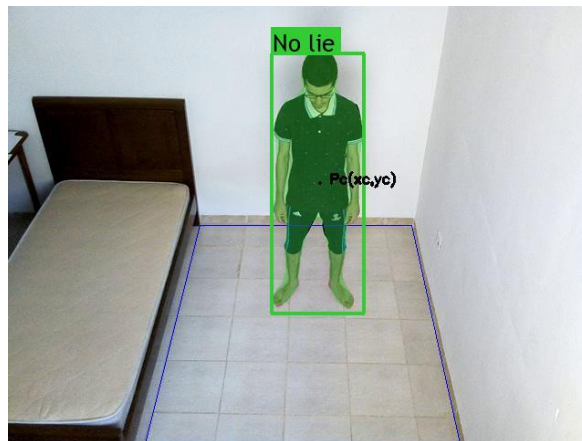


Figure 91 Le point  $P_c$  représentant le centre du cadre délimitant la personne

Comment peut-on reconnaître si la personne se trouve à l'intérieur de la zone représentant le sol (elle se trouve à l'intérieur du polygone) ? Pour chaque image de la vidéo faire :

- Dans la figure ci-dessus le polygone est dessiné en spécifiant 04 points :  $P_1(x_1, y_1)$ ,  $P_2(x_2, y_2)$ ,  $P_3(x_3, y_3)$  et  $P_4(x_4, y_4)$ ,
- Déterminer le point  $P_c(x_c, y_c)$  qui correspond au centre du cadre délimitant la personne (*bounding box*) à l'aide de l'algorithme suivant :
  - Calculer la largeur du cadre de délimitation :  $width_{bbox} = x_{2\_bbox} - x_{1\_bbox}$ ,
  - Calculer la hauteur du cadre de délimitation :  $height_{bbox} = y_{3\_bbox} - y_{1\_bbox}$ ,
  - Calculer la coordonnée  $x_c$  :  $x_c = x_{1\_bbox} + \frac{width_{bbox}}{2}$ ,
  - Calculer la coordonnée  $y_c$  :  $y_c = y_{1\_bbox} + \frac{height_{bbox}}{2}$ .
- Si  $\left[ x_c > \frac{x_1+x_3}{2} \text{ et } x_c < \frac{x_2+x_4}{2} \right]$  et  $[y_c \geq y_1 \text{ ou } y_c \leq y_3]$  alors la personne se trouve à l'intérieur de la zone représentant le sol.

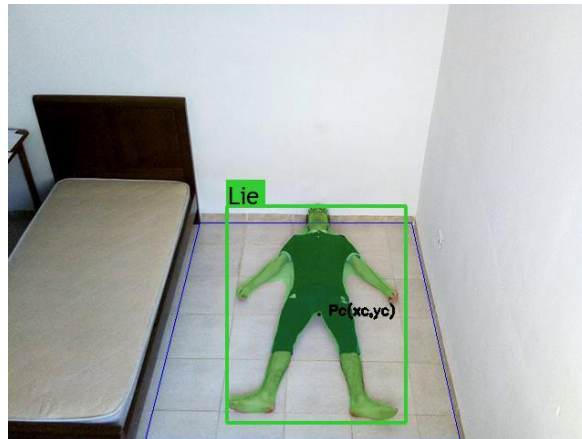


Figure 92 La personne est allongée à l'intérieur de la zone représentant le sol

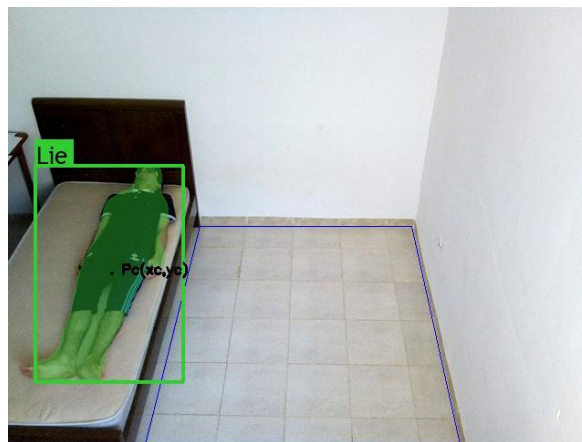


Figure 93 La personne n'est pas allongée à l'intérieur de la zone représentant le sol

À partir du moment où la personne est allongée dans la zone représentant le sol, un compte à rebours se met en marche. Si elle reste allongée dans cette zone pendant 30 secondes, une chute sera signalée, si par contre elle se remet debout, le compte à rebours sera réinitialisé et aucune chute ne sera signalée.

#### 4. Conclusion

Dans ce chapitre, nous avons défini les différents besoins fonctionnels et techniques. Ensuite, nous avons présenté la conception software de l'application à travers le diagramme de cas d'utilisation, diagramme de classes, diagramme de composants et les diagrammes de séquence. Puis, nous avons présenté la conception hardware des différents composants électroniques qui constituent notre système de détection des fuites de gaz toxiques, à travers les schémas de chaque branchement, ensuite, nous avons clôturé cette partie en présentant l'architecture globale de notre système. Enfin, nous avons abordé la conception de notre système de détection des chutes.

Dans le chapitre suivant, nous nous intéresserons à l'implémentation du système de détection des fuites de gaz ainsi que celui des chutes.

# **Chapitre 6 : Implémentation et Tests**



## 1. Introduction

Après une étude approfondie des concepts théoriques des techniques à implémenter, nous allons dans ce chapitre décrire le processus de réalisation de notre système en mettant en évidence l'ensemble des environnements de développement que nous avons utilisés, un aperçu sur les modules de notre application ainsi qu'une présentation des résultats obtenus des différents tests expérimentaux réalisés.

## 2. Implémentation et Tests du système de détection des fuites de gaz

### 2.1. Présentation de la plateforme

Afin d'avoir le contrôle des différents équipements de notre projet, nous avons développé une plateforme sous forme d'une application web, ce type d'application est très utilisé de nos jours car il offre la possibilité de contrôler un système depuis n'importe quel ordinateur (ou smartphone car notre plateforme est aussi conçue pour s'adapter à n'importe quelle taille d'écran) connecté à internet sans avoir besoin d'installer des logiciels ou pilotes supplémentaires. Pour avoir accès à cette application, il suffit juste d'ouvrir un navigateur web et d'y entrer l'adresse de l'application.

Après avoir accédé et s'être authentifié, chaque utilisateur est dirigé vers son espace personnel, en fonction de son rôle et ses droits d'accès (chaque utilisateur aura un menu différent). Dans ce qui suit, nous allons présenter les différents modules de notre plateforme en prenant en considération l'utilisateur de type « Administrateur distant » car c'est celui qui possède toutes les fonctionnalités que propose le système.

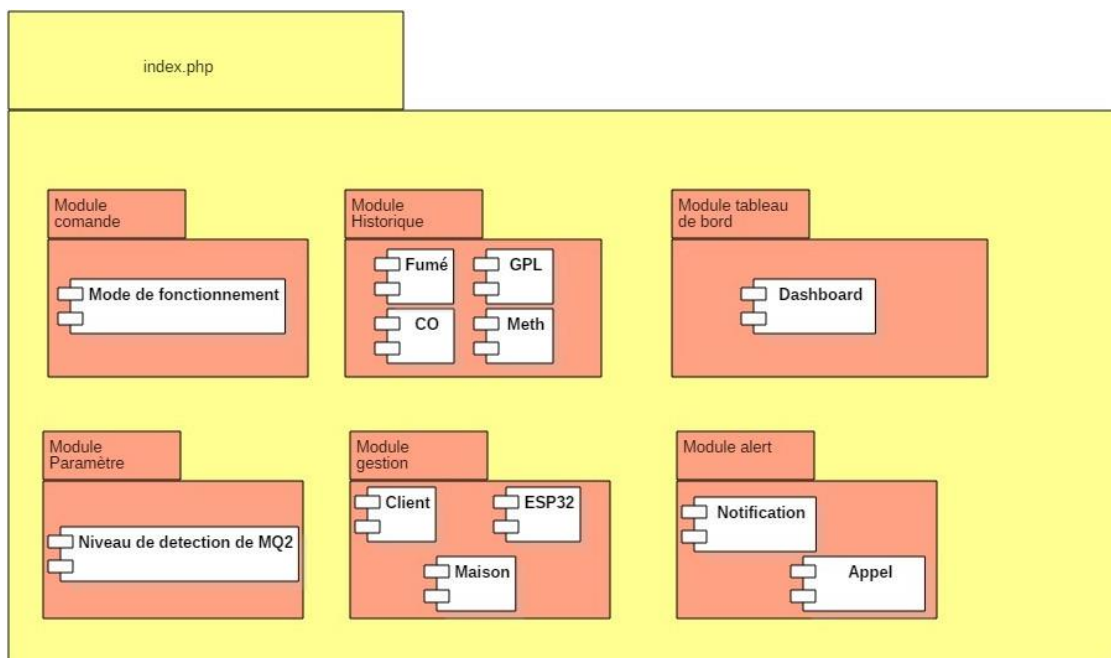


Figure 94 Vue globale sur les modules de la plateforme développée

Comme le montre la figure 98, notre plateforme est constituée de 6 modules : Module Tableau de bord, Module Historique, Module Commande, Module Gestion, Module Alerte, Module Paramètre, Ils sont tous accessibles depuis la page principale de la plateforme.

### 2.1.1. Module Tableau de bord

Ce module, nous donne beaucoup d'informations sur l'évolution des paramètres de notre système, Nous avons 4 jauges colorées qui nous donne la valeur actuelle de chaque taux de gaz toxique, ces jauges sont actualisées chaque seconde.

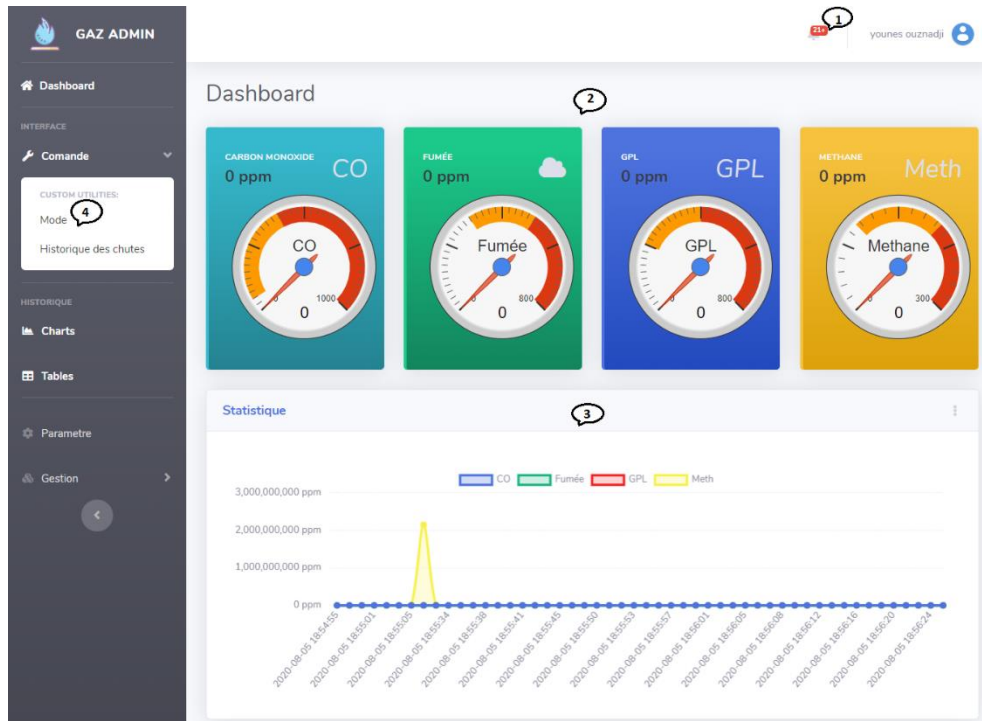


Figure 95 Module Tableau de bord qui montre l'évolution des paramètres du système

- A l'aide du menu numéro « 1 » on peut consulter les notifications d'alertes signalant la présence d'une fuite de gaz ou une chute d'une personne ou une intrusion de personnes malveillantes.
- Sur le numéro « 2 » on peut visualiser en temps réel les changements des taux actuels des différents gaz.
- Sur le numéro « 3 » on voit l'historique des 30 dernières captures (taux collectés par le capteur de gaz MQ-2) des différents gaz.
- Sur le numéro « 4 » où on peut contrôler les différents actionneurs ou commutateurs du système, ainsi que son mode de fonctionnement (manuel/automatique).

### 2.1.2. Module Historique

Ce module se compose de trois pages (Charts, Tables, Chutes) :

- Charts : pour visualiser l'historique des changements des taux de gaz toxiques dans le temps sous forme d'un graphe.
- Tables : pour visualiser l'historique des changements des taux de gaz toxiques dans le temps sous forme d'un tableau.
- Chutes : pour visualiser l'historique des chutes survenues dans le temps sous forme d'un tableau.

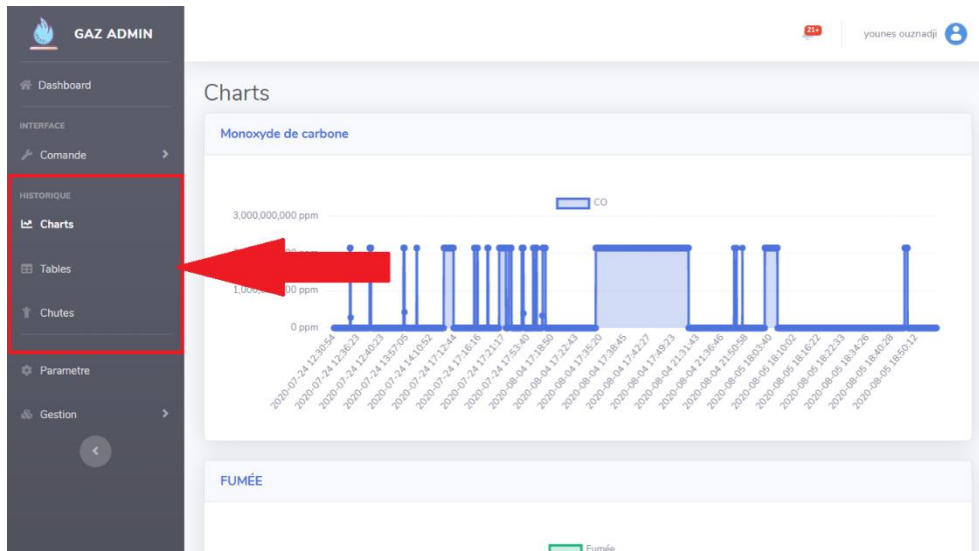


Figure 96 Module historique (la page Charts)

### 2.1.3. Module Commande

Ce module nous permet de choisir entre la commande automatique ou la commande manuelle :

- Commande automatique : Lorsque la commande automatique est activée, notre système intervient automatiquement au niveau du domicile lors de la détection d'un taux anormal d'un gaz donné :
  - Couper l'alimentation électrique pour éviter le contact électricité-gaz.
  - Fermer la vanne principale de gaz.
  - Aération de l'appartement grâce à l'ouverture des fenêtres et l'activation des extracteurs d'air.
  - L'activation de l'alarme sonore et visuelle.
  - Envoi d'une alerte via un Appel/SMS à l'aide d'un module GSM.
- C. Manuelle : Dans le cas où la commande automatique est désactivée, nous pouvons commander manuellement les différents actionneurs de l'appartement via les divers boutons, Ces boutons nous donnent l'état actuel de chaque actionneur (ON/OFF) (OPEN/CLOSE).

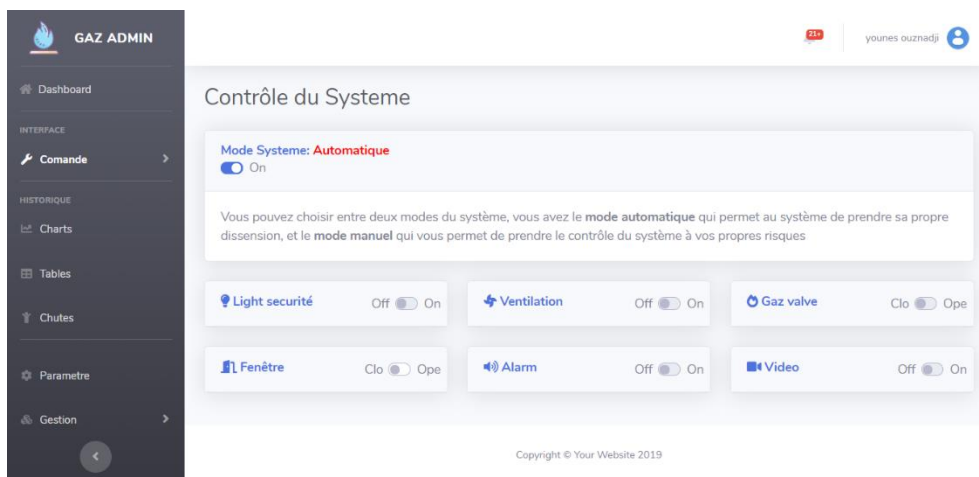


Figure 97 Module Commande

### 2.1.4. Module Gestion

Ce module est pour assurer la gestion des clients (ajouter un nouveau client, supprimer un client, modifier les informations relatives à un client).

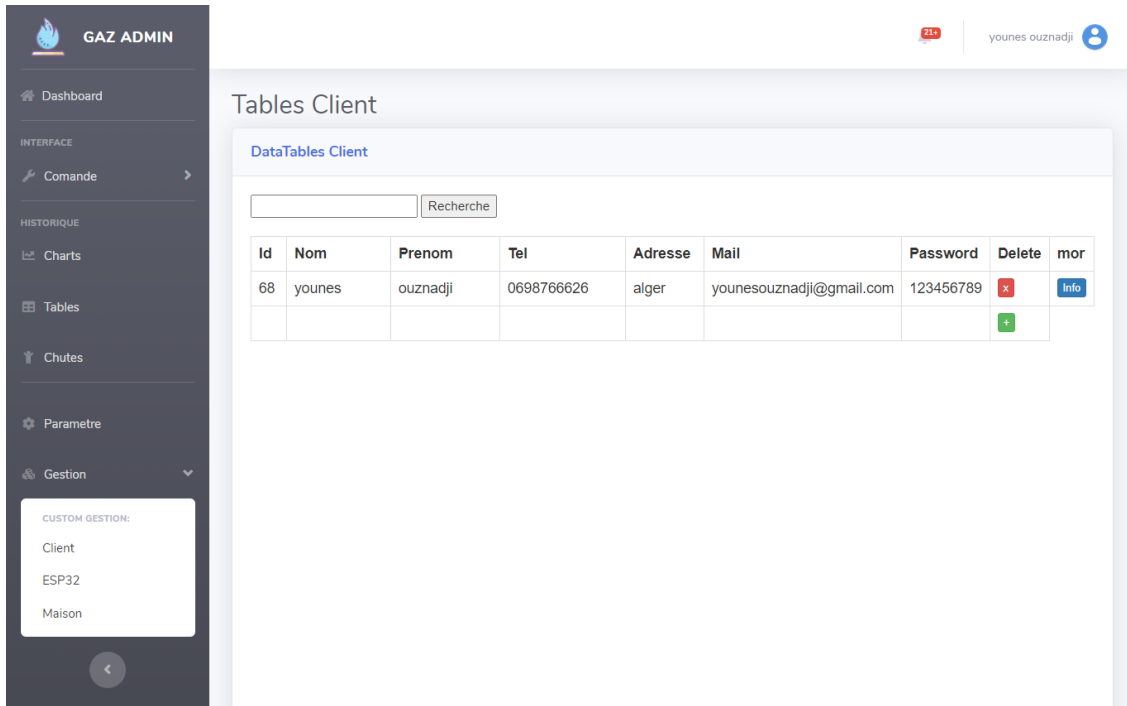


Figure 98 Module Gestion

### 2.1.5. Module Alerte

À travers ce module on peut consulter les notifications d’alertes suite à un évènement de fuite de gaz ou de chute ou d’une intrusion.

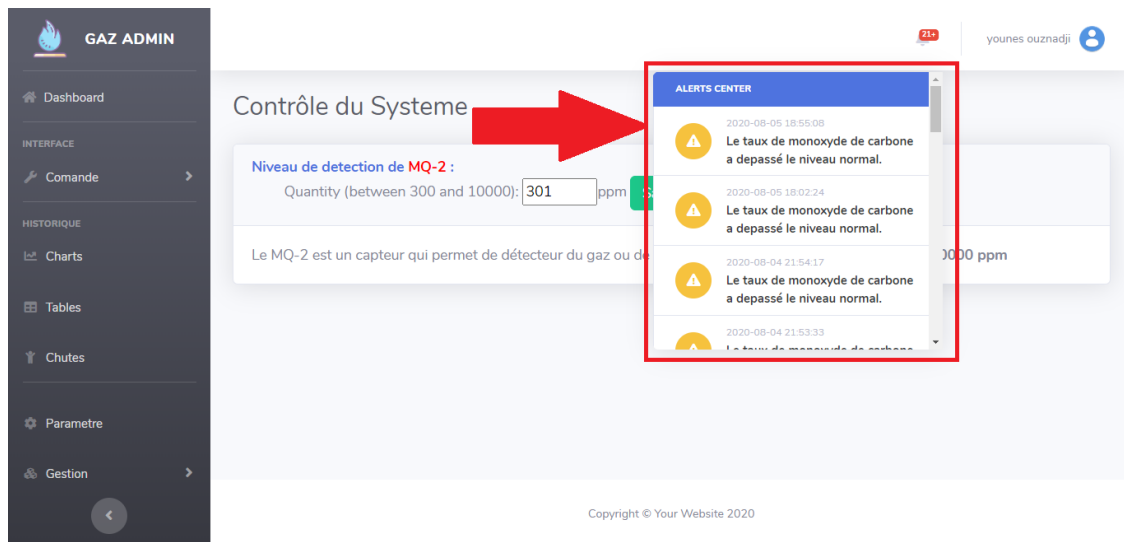


Figure 99 Module Alerte

### 2.1.6. Module Paramètre

Dans cette partie, l’administrateur distant peut modifier le seuil minimum de tolérance (entre une plage de valeur de 300 à 10000 ppm) que les gaz nocifs ne doivent pas dépasser. La nouvelle valeur sera mise à jour pour tous les utilisateurs.



Figure 100 Module Paramètre

## 2.2. Tests et Evaluation du système

### 2.2.1. Tests des fonctionnalités du système

- Fonction de contrôle : des fenêtres, d'extracteur d'air, de l'alimentation électrique, de la vanne principale de gaz et de l'alarme.
  - La fonction de contrôle des fenêtres est assurée via notre plateforme en appuyant sur le bouton approprié, qui va envoyer un signal par la suite vers le relais branché à la carte ESP32 (la carte ESP32 est liée à un routeur) afin d'ouvrir ou de fermer les fenêtre (de même dans le mode automatique) et de même pour les fonctionnalités d'extraction d'air, contrôle de l'alimentation électrique, contrôle de la vanne principale de gaz et contrôle de l'alarme.

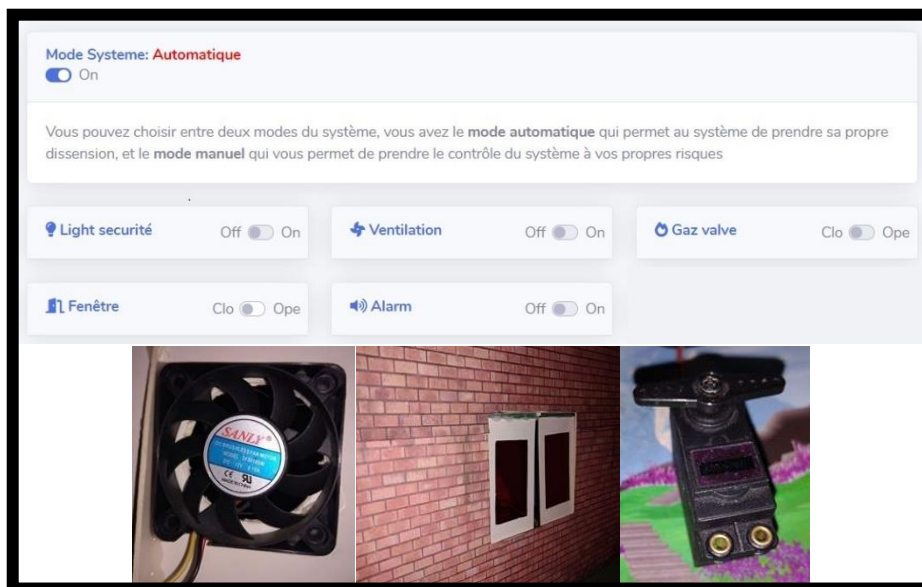


Figure 101 Fonction de contrôle du système

- Fonction d'acquisition des taux actuels des gaz toxiques présents dans l'air.
  - La fonction d'acquisition des taux actuels des gaz présents dans l'environnement est réalisée par le capteur de gaz MQ-2, ensuite, ces taux seront enregistrés dans une base de données et affichés sur la plateforme.



Figure 102 Fonction d'acquisition des taux actuels des gaz toxiques présents dans l'air

- Fonction d'alerte : nous avons 04 types différents d'alerte.
  - Alerte via une alarme sonore.
  - Alerte via une alarme visuelle (couleur des lampes LED).
  - Alerte via une notification envoyée à la plateforme.
  - Alerte via un appel effectué à l'aide du module GSM.



Figure 103 Alerte via une alarme sonore et visuelle (la LED rouge est allumée lors d'une situation dangereuse)

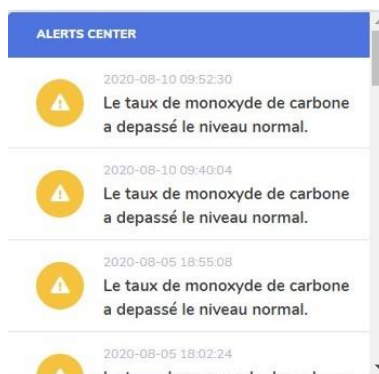


Figure 104 Alerte via une notification envoyée à l'application Web/Mobile

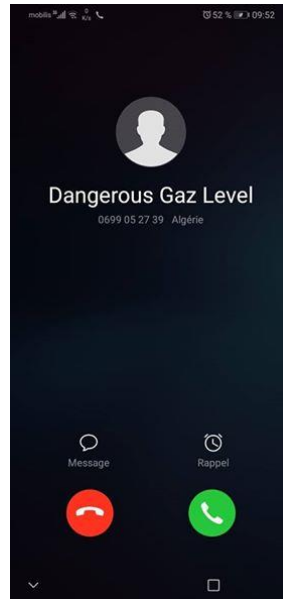


Figure 105 Alerte via un appel téléphonique

### 2.3. Evaluation des performances du système

Afin de tester le bon fonctionnement et les performances de notre système de contrôle et de détection des fuites de gaz, on a réalisé des tests sur le comportement du système lors de la détection des gaz nocifs ou de la fumée à l'aide d'un prototype d'une maison, qui représente des conditions presque similaires aux conditions réelles.

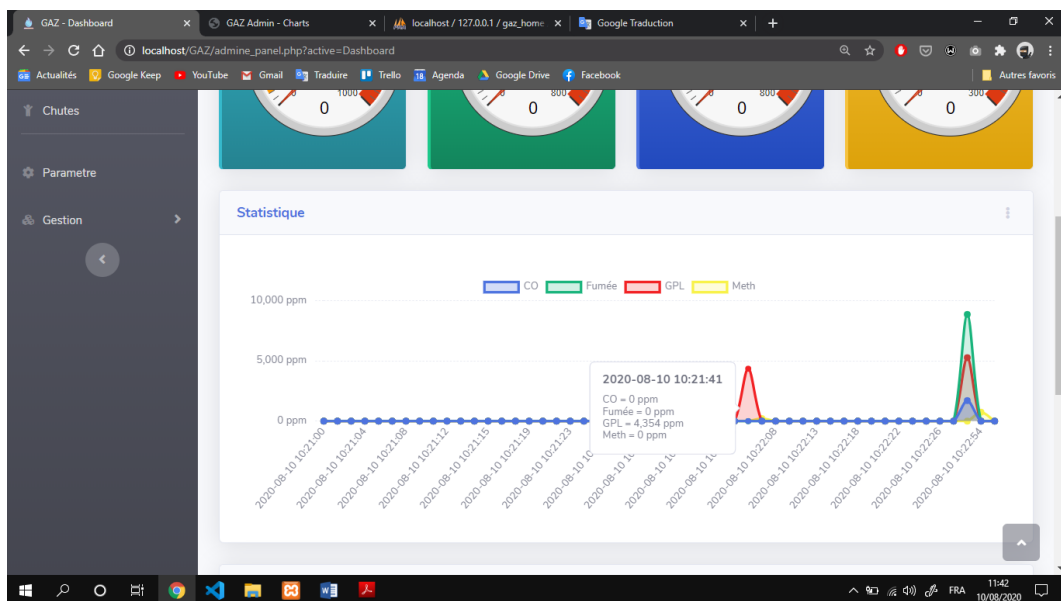


Figure 106 Graphe représentant l'évolution des taux des gaz nocifs dans le temps

- Interprétation : le seuil de tolérance minimum est configuré à 300 PPM, on remarque bien lors de la détection d'un des gaz nocifs ou de la fumée, le graphe affiche en temps réel la valeur du taux capturé par le capteur MQ-2, de plus la jauge représentant le gaz capturé est mise à jour avec la nouvelle valeur du taux. Ensuite, le système exécute des actions permettant de diminuer le taux du gaz nocif (ou de la fumée) présent dans l'air et donc sauver les personnes présentes à domicile d'une asphyxie mortelle.



- Interprétation et discussion des résultats :
  - Durant les tests réalisés, on a remarqué que notre système répond parfaitement et en temps réel lors de la détection d'une fuite de gaz ou de la fumée.
  - On a constaté que : pour assurer une précision de capture (lors de l'installation du système dans un milieu réel), il est nécessaire d'utiliser des capteurs industriels pour chaque gaz nocif. Pour notre prototype l'utilisation d'un capteur qui détecte plusieurs gaz à la fois est suffisant pour faire les tests.

### 3. Implémentation et tests du système de détection des chutes

#### 3.1. La phase d'entraînement (training) du réseau de neurones convolutif

##### 3.1.1. Le jeu de données (dataset)

Afin d'obtenir un classificateur ou un modèle robuste capable de reconnaître de manière sûre si une personne est allongée ou non, on a créé notre propre jeu de données (*dataset*) constitué de 297310 images de taille 30x30 pixels sans arrière-plan (fond noire) contenant seulement le masque de segmentation de résolution 28x28 pixels (représentant la silhouette de la personne exerçant une posture ou un mouvement donné) prédit et généré par le modèle (pré-entraîné sur le jeu de données *COCO*) utilisé par le *framework Mask R-CNN* pour réaliser la segmentation d'objets (personnes) dans une image. Le *dataset* est constitué d'un ensemble de données d'entraînement ou *training set* (sur lequel est entraîné le réseau neuronal convolutif durant chaque *epoch*) et un ensemble de données de validation ou *validation set* (sur lequel on évalue le réseau neuronal convolutif à la fin de chaque *epoch* après la phase d'entraînement ou *training*), dans chacun, deux classes sont présentes : *no lie* et *lie*. Le *training set* contient au total 208170 images (78975 images appartenant à la classe *no lie* et 129195 images à la classe *lie*). Le *validation set* contient au total 89140 images (65217 images appartenant à la classe *no lie* et 23923 images à la classe *lie*). Les images de la classe *no lie* représentent différentes postures à savoir : se tenir debout, marcher, s'asseoir et se pencher, ainsi que des mouvements transitoires entre ces différentes postures (lorsque la personne est debout puis elle s'assoie, lorsque la personne est assise puis elle se lève, lorsque la personne est debout ensuite elle se penche pour ramasser un objet, lorsque la personne est penchée ensuite elle se met debout). Les images contenues dans la classe *lie* ne représentent que la posture allongée.

Jeu de données contenant 297310 images			
Training set		Validation set	
No lie	Lie	No lie	Lie
78975 images	129195 images	65217 images	23923 images

Tableau 10 Représentation de la structure de notre jeu de données

Comment avons-nous créé notre jeu de données ? D'abord, on a enregistré 47 vidéos de différentes durées divisées comme suit : 33 vidéos (21 vidéos où la personne exerce deux postures : se tenir debout et



marcher, 8 vidéos où la personne se penche pour ramasser un objet puis se remet debout et 4 vidéos où la personne s’assoie puis se remet debout) pour construire la classe *no lie* (18 vidéos destinées au *training set* et 15 vidéos au *validation set*) et 14 vidéos pour construire la classe *lie* (11 vidéos destinées au *training set* et 3 vidéos au *validation set*) où la personne n’exerce que la posture allongée.

Vidéos			
Training set		Validation set	
No lie	Lie	No lie	Lie
18	11	15	3

Tableau 11 Le partitionnement des vidéos enregistrées portant création de notre jeu de données

Une vidéo est constituée d’un ensemble de *frames* ou images, pour chaque image contenant une personne, l’étape abordée précédemment ‘Extraction de la silhouette du corps humain’ est appliquée, où on obtient comme résultat une image de résolution 30x30 pixels sans arrière-plan (fond noire) contenant seulement un masque de segmentation de résolution 28x28 pixels représentant la silhouette de la personne. Puis, ces images obtenues sont classées dans l’ensemble de données (d’entraînement ou de validation) et la classe auxquels appartient la vidéo traitée.

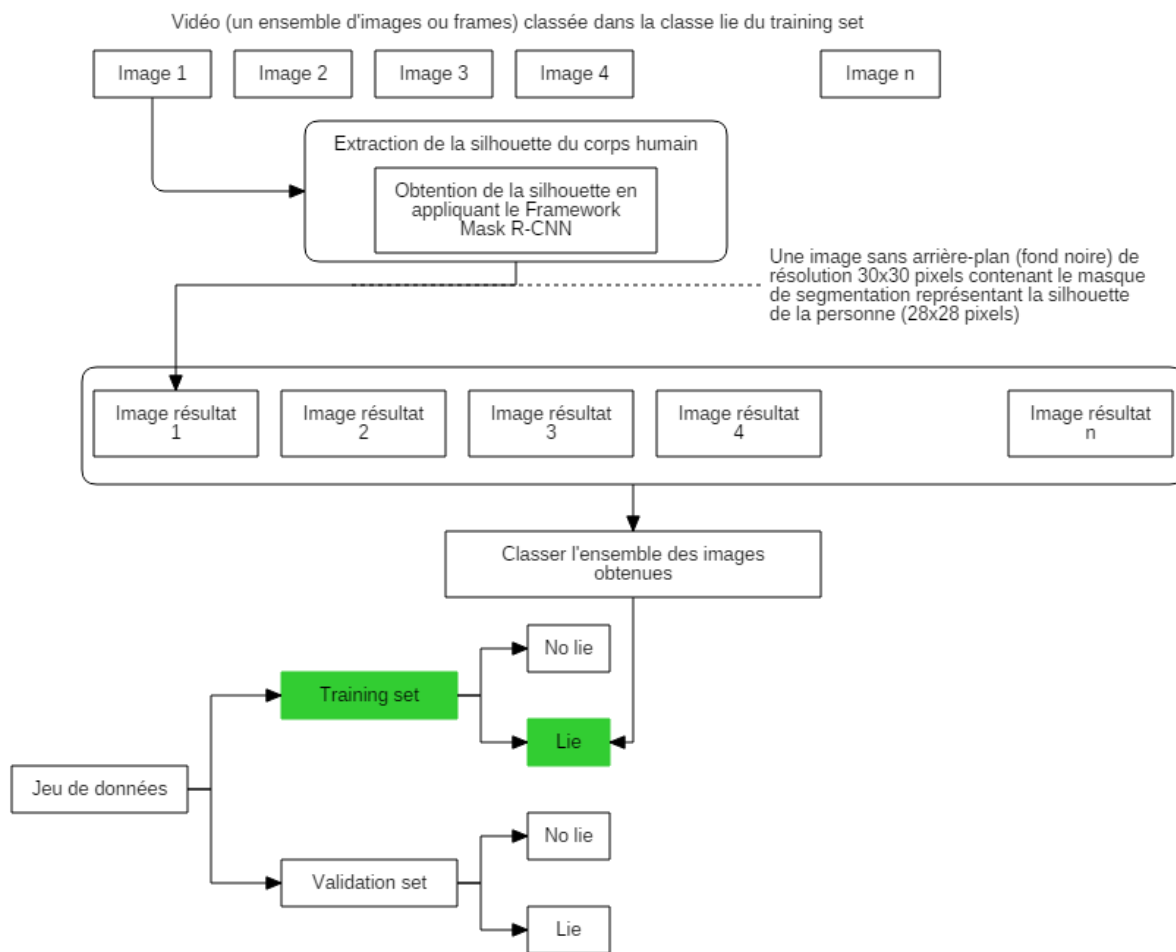


Figure 107 Illustration des étapes suivies dans la création de notre jeu de données

**Remarque :** vu qu'on a utilisé la technique *Image data augmentation* qui consiste à augmenter artificiellement la taille de l'ensemble d'exemples d'entraînement ou *training set* en lui appliquant des transformations (la rotation, le retournement horizontal et vertical), on a pu réduire le nombre de vidéo à enregistrer et à traiter pour créer notre *dataset*, par exemple :

- Enregistrer une vidéo où la personne marche de la droite vers la gauche puis une autre où elle marche dans le sens inverse (de la gauche vers la droite) n'est pas nécessaire, car, il suffit simplement d'enregistrer l'une des deux, puis, d'appliquer l'étape 'Extraction de la silhouette du corps humain' sur les *frames* de la vidéo pour obtenir un ensemble d'images contenant la silhouette de la personne qui marche dans un sens donné. Ensuite, le retournement horizontal est appliqué sur cet ensemble pour obtenir un autre ensemble d'images où la personne marche dans l'autre sens.
- Enregistrer des vidéos où la personne est allongée dans toutes les directions n'est pas nécessaire, car, il suffit seulement d'enregistrer une seule vidéo où elle est allongée dans une direction donnée, ensuite, on applique la rotation suivant plusieurs angles compris entre  $-180^\circ$  et  $180^\circ$  sur l'ensemble des images obtenues en appliquant l'étape 'Extraction de la silhouette du corps humain' sur les *frames* de la vidéo.

Quelques échantillons de notre jeu de données sont illustrés dans le tableau suivant.

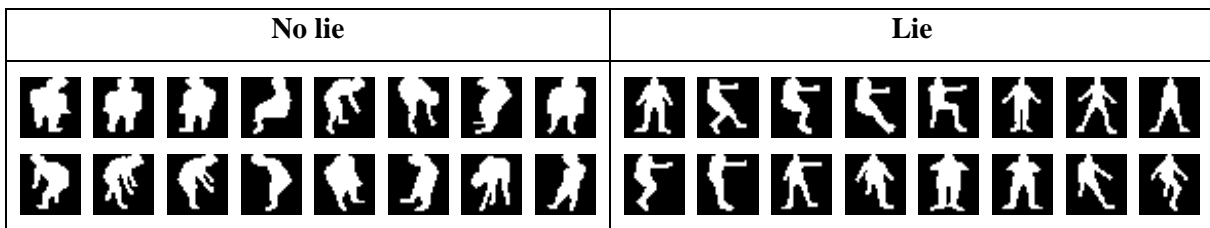


Tableau 12 Quelques exemples de notre jeu de données

### 3.1.2. Évaluation des performances de notre classificateur

Afin d'obtenir un classificateur fiable capable de reconnaître de manière sûre si une personne est allongée ou non, nous avons entraîné et évalué notre modèle sur notre jeu de données pendant 11 epochs puis 15 epochs avec un *batch* constitué de 32 images, donc nous avons obtenu deux classificateurs différents. En se basant sur la métrique *f-measure* nous avons choisi le meilleur classificateur avec lequel nous allons effectuer les tests de reconnaissance de postures et des chutes.

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.99	1.0	0.99	0.99	$1.22 \times 10^{-18}$

Tableau 13 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle pendant 11 epochs

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	1.0	1.0	1.0	1.0	$1.42 \times 10^{-13}$

Tableau 14 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle pendant 15 epochs

Nous pouvons constater que le classificateur entraîné et évalué pendant 15 epochs est plus performant que celui entraîné et évalué pendant 11 epochs.

		La classe prédite	
		No lie (1)	Lie (0)
La classe attendue	No lie (1)	65217	0
	Lie (0)	0	23923

Tableau 15 La matrice de confusion décrivant les performances de notre classificateur entraîné et évalué pendant 15 epochs

3.1.3. Graphes représentant l'évolution des métriques pendant 15 epochs

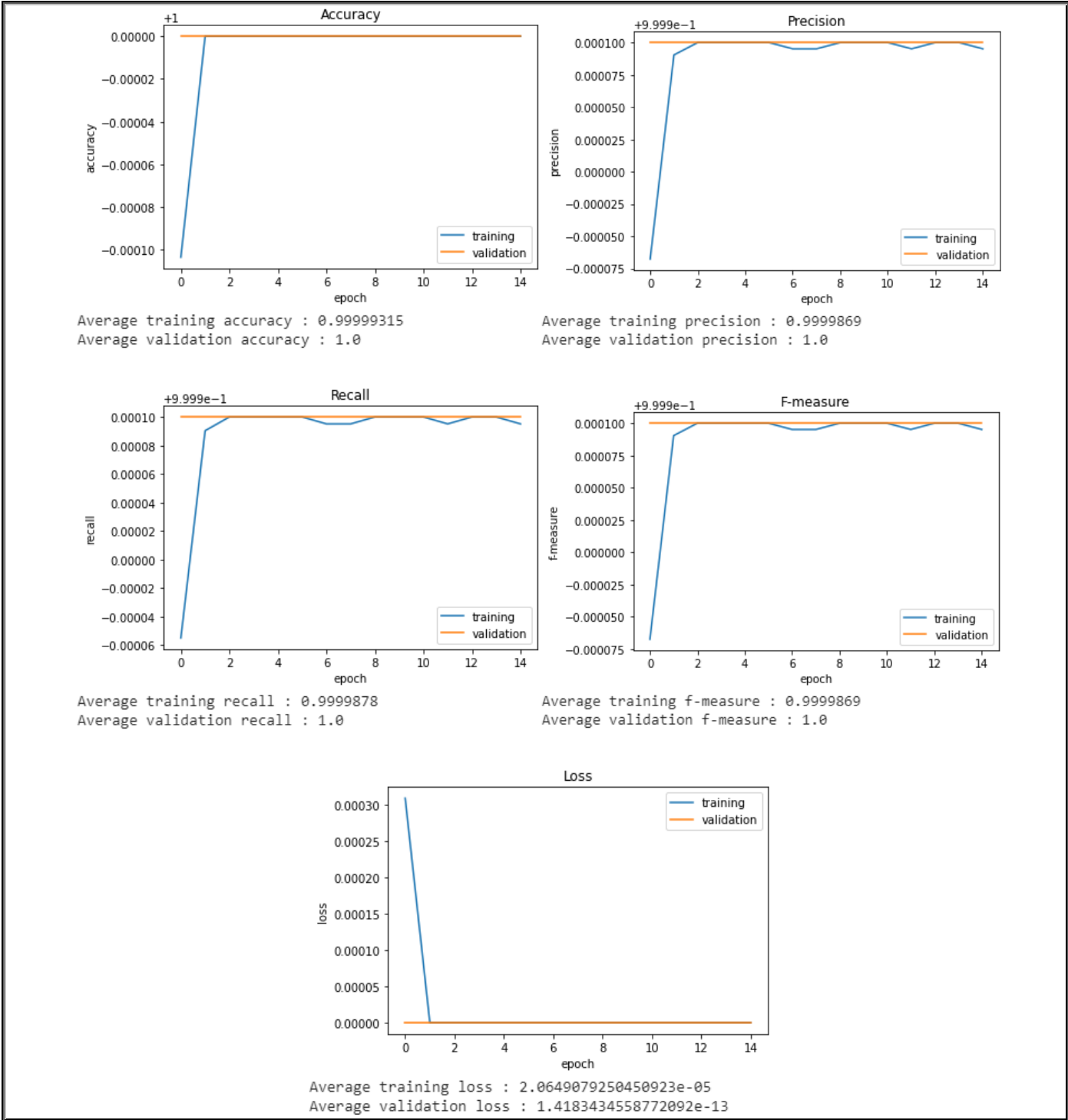
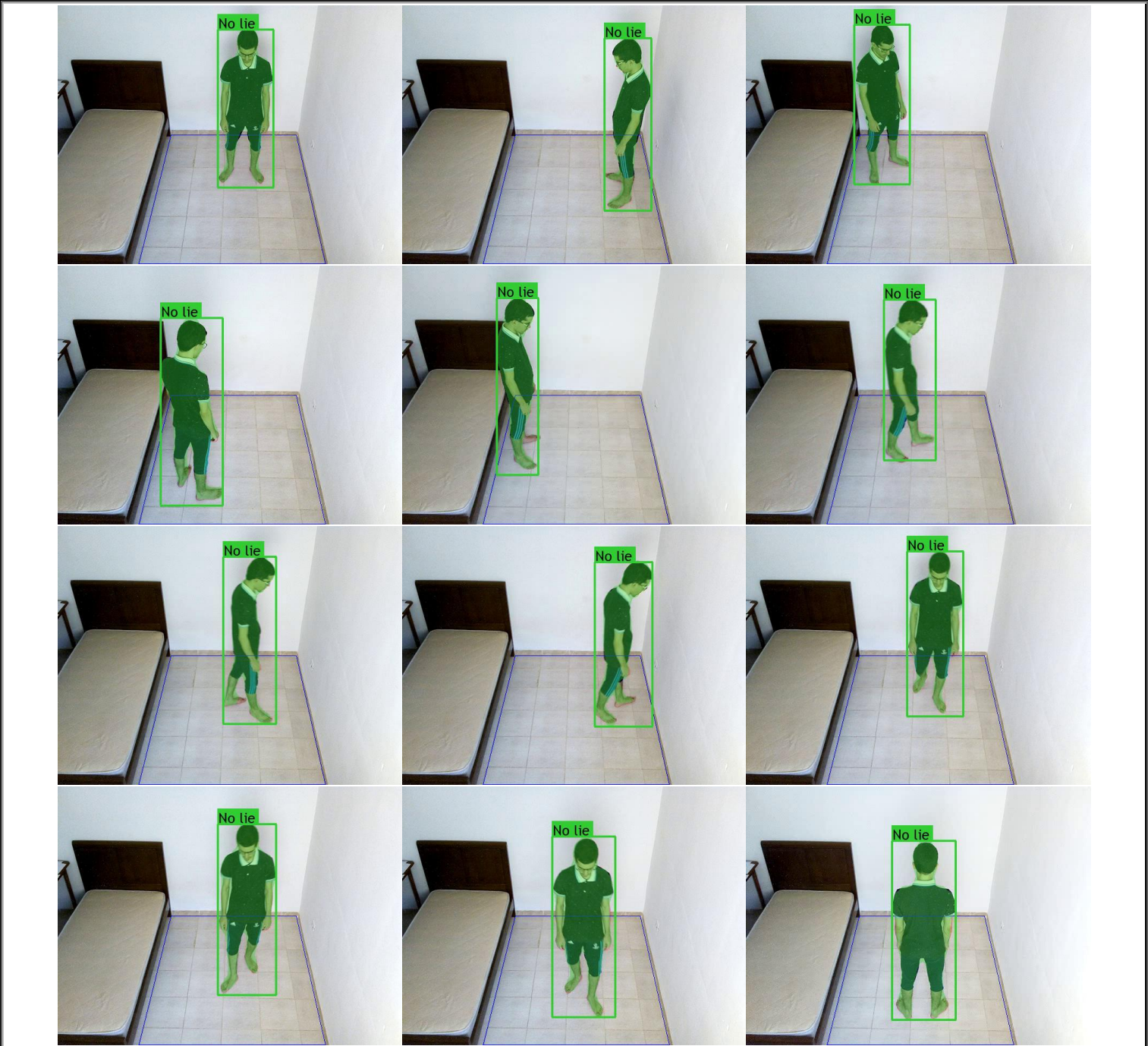


Figure 108 Graphes représentant l'évolution des métriques pendant 15 epochs

### 3.2. Tests et évaluation de notre système de détection des chutes

#### 3.2.1. Tests de reconnaissance de postures

Afin d'évaluer notre système notamment le classificateur obtenu, nous avons enregistré un ensemble de vidéos où la personne exerce différentes postures et mouvements : lorsque la personne est debout, lorsque la personne marche, lorsque la personne se penche pour ramasser un objet puis elle se relève, lorsque la personne est assise et lorsque la personne est allongée.





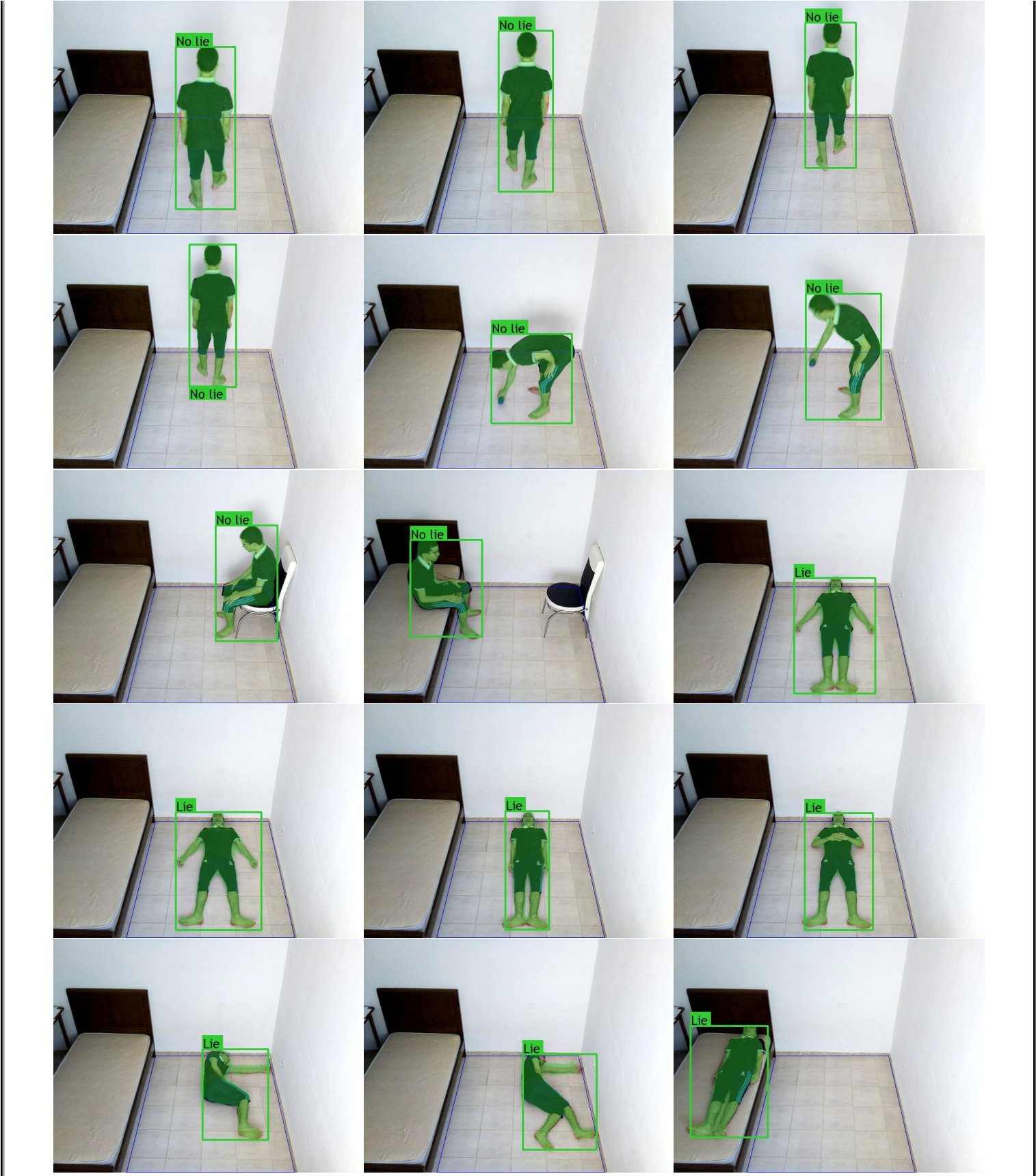


Figure 109 Les tests de reconnaissance de postures effectués

### 3.2.2. Tests de détection des chutes

Un évènement de chute est détecté, si et seulement si, la personne est allongée à l'intérieur de la zone représentant le sol pendant une durée de 30 secondes (pendant 30 secondes, les images de résolution 30x30 pixels (contenant la silhouette de la personne) obtenues à la fin de l'étape 'Extraction de la silhouette du corps humain' ont été toutes classées par le modèle dans la classe *lie*). Lors de la détection d'une chute, une alarme sonore et visuelle sera déclenchée au niveau du domicile de la personne ayant chuté et une alerte via un Appel/SMS est envoyée aux contacts introduits préalablement dans le système ainsi qu'aux services de la protection civile, de plus, une notification est envoyée à l'application Web/Mobile (utilisée par un administrateur distant appartenant aux services de la protection civile et les occupants du domicile) et le temps exact de la chute est sauvegardé dans l'historique des chutes.

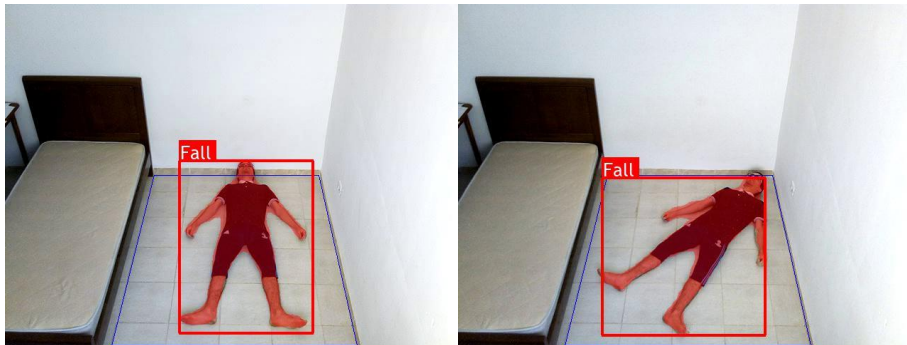


Figure 110 La personne été allongé dans la zone représentant le sol pendant une durée de 30 secondes

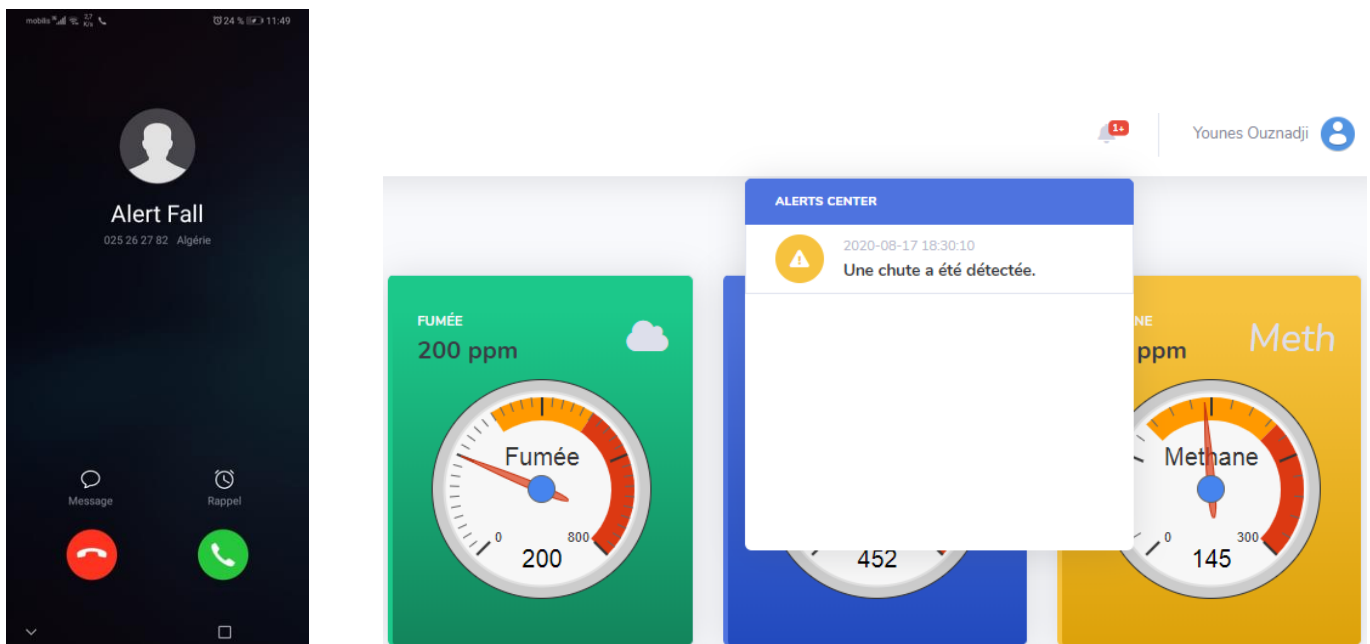


Figure 111 Une alerte via un appel et une notification envoyée à l'application Web/Mobile indiquant qu'une personne a chuté

### 3.2.2.1. Historique des chutes

Pour consulter toutes les chutes de personnes qui se sont produites au cours du temps.

ID	Time
43	2020-08-17 18:30:10
44	2020-08-17 20:07:11
45	2020-08-17 20:18:48
46	2020-08-17 20:24:18

Figure 112 Historique des chutes

### 3.2.3. Tests de notre système de surveillance

Le système que nous avons élaboré, offre à l'utilisateur la possibilité de choisir entre deux modes : mode détection des chutes qu'on vient d'illustrer précédemment et le mode surveillance qui permet la détection des intrusions de personnes malveillantes au domicile de l'utilisateur. Si l'utilisateur ou l'un de ces proches est présent au domicile, il opte pour le mode détection des chutes, sinon, s'il n'y a aucune présence, l'utilisateur peut déployer le mode surveillance. Si une intrusion est détectée, une alerte via un Appel/SMS est envoyée aux contacts introduits préalablement dans le système et une notification est envoyée à l'application Web/Mobile (utilisée par un administrateur distant et les occupants du domicile), de plus, le système sauvegarde le temps exact de l'intrusion dans l'historique des intrusions.



Figure 113 Tests de notre système de surveillance



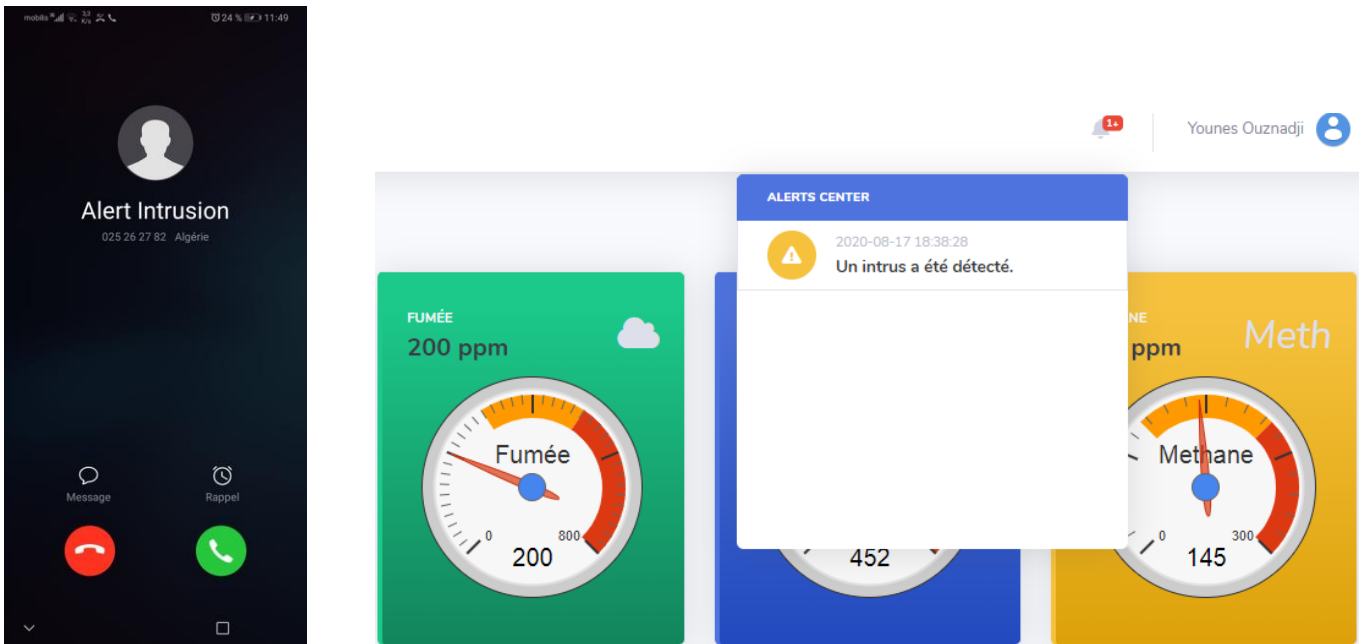


Figure 114 Une alerte via un appel et une notification envoyée à l'application Web/Mobile indiquant qu'une personne maleveillante c'est introduit au domicile

### 3.2.3.1. Historique des intrusions

Pour consulter toutes les intrusions de personnes malveillantes survenues dans le temps.

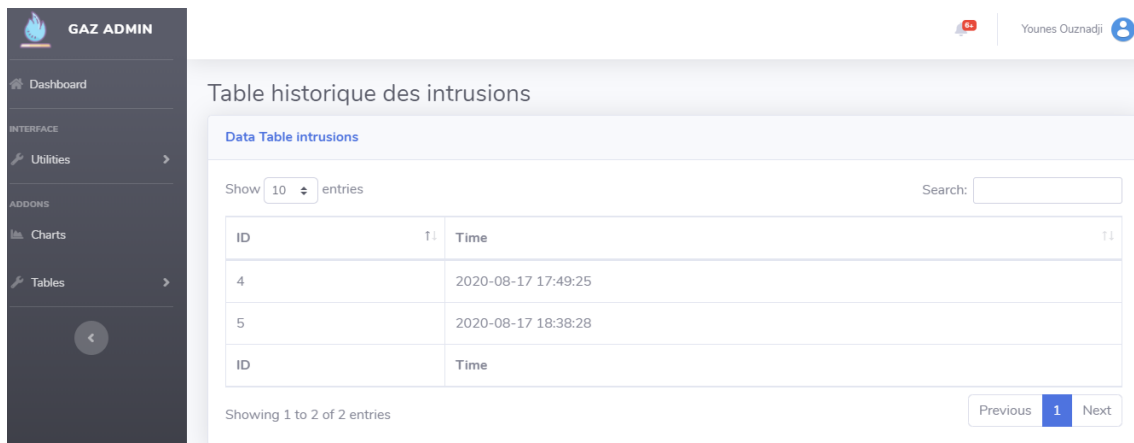


Figure 115 Historique des intrusions

## 3.3. Comparaison et validation de notre travail

Les auteurs de l'article [45] ont élaboré un système de détection des chutes basé sur la reconnaissance de postures, capable d'identifier 04 postures différentes qui sont : se tenir debout (*stand*), se pencher (*bend*), s'asseoir (*sit*) et s'allonger (*lie*). La première étape du système consiste à extraire la silhouette du corps humain depuis une image ou *frame* d'une vidéo à l'aide d'une méthode de soustraction d'arrière-plan appelée *codebook background subtraction*, où l'arrière-plan de la pièce où se trouve la personne est supprimé, obtenant ainsi une image contenant uniquement la silhouette de l'individu. Ensuite, l'image est introduite dans un réseau de neurones convolutif qui a la tâche de la classer dans l'une des classes suivantes : se tenir debout (*stand*), se pencher (*bend*), s'asseoir (*sit*) et s'allonger (*lie*). Un évènement de chute est détecté lorsque la personne est allongée (le modèle classe toutes les images



obtenues (par l'étape d'extraction de la silhouette) pendant une durée donnée dans la classe *lie*) sur le sol (la personne est allongée à l'intérieur de la zone représentant le sol) pendant une durée donnée. La zone représentant le sol est détectée à l'aide d'une méthode automatique d'apprentissage non supervisé basée sur la position des pieds de la personne.

L'idée de notre système de détection des chutes s'inspire de celle élaborée par les auteurs de l'article [45], cependant, on peut clairement constater que les deux systèmes diffèrent dans :

- La méthode utilisée pour extraire la silhouette d'une personne à partir d'une image ou *frame* d'une vidéo. Selon nos recherches effectuées (étude de l'existant), nous sommes les premiers au niveau internationale à élaborer un système de détection des chutes qui se base principalement sur le *framework Mask R-CNN* pour extraire la silhouette représentant l'individu depuis une image d'une vidéo.
- La méthode utilisée pour supprimer l'arrière-plan de la pièce où se trouve la personne.
- Le jeu de données utilisé pour entraîner et évalué le réseau de neurones convolutif.
- Le nombre de classe en sortie du modèle (contrairement aux auteurs de l'article [45], nous avons élaboré un classificateur binaire capable de classer l'image (contenant la silhouette de la personne) obtenu à la fin de l'étape 'Extraction de la silhouette du corps humain' dans l'une des deux classes suivantes : allongée (*lie*) ou non allongée (*no lie*).
- La méthode utilisée pour sélectionner la zone représentant le sol.

Les différences énoncées précédemment, nous ont permis de valider notre travail en effectuant :

- Une comparaison entre notre méthode et celle utilisée par les auteurs de l'article [45] pour extraire la silhouette d'un individu à partir d'une image ou *frame* d'une vidéo.
- Une comparaison entre le jeu de données que nous avons créé et celui réalisé par les auteurs de l'article [45] pour construire un classificateur capable de reconnaître si une personne est allongée ou non.
- Une comparaison entre les performances obtenues par deux modèles, l'un lorsque notre réseau neuronal convolutif est entraîné et évalué sur notre jeu de données et l'autre sur celui des auteurs de l'article [45]. Pour pouvoir effectuer cette comparaison, on doit d'abord modifier leur jeu de données de telle façon à ce qu'il comporte seulement deux classes (*lie* et *no lie*) comme dans notre *dataset*. Une fois que le classificateur est obtenu (en entraînant et évaluant le réseau de neurones convolutif sur le jeu de données des auteurs de l'article [45] que nous avons modifié par la suite), on doit ressortir la matrice de confusion, puis, calculer les métriques *accuracy*, *precision*, *recall*, *f1-score* ou *f-measure* et *loss*. En se basant sur la métrique *f-measure* nous pouvons savoir qui des deux modèles est le plus performant et par conséquent, nous pouvons connaître qui des deux jeux de données est le meilleur.
- Des tests de reconnaissance de postures (*no lie* et *lie*) en utilisant le classificateur obtenu.

### 3.3.1. La méthode d'extraction de la silhouette du corps humain

Pour extraire la silhouette d'une personne à partir d'une image ou *frame* d'une vidéo, les auteurs de l'article [45] ont conçu une méthode de soustraction d'arrière-plan nommée *codebook background subtraction*, qui consiste en une soustraction de deux images, la première est appelée image de référence, elle représente seulement l'arrière-plan de la pièce (le sujet d'intérêt (la personne) que nous souhaitons extraire par la suite n'est pas présent dedans) où on y trouve seulement des objets, par exemple : un lit, un tapis, une chaise, une télévision, etc... Tandis que la deuxième image appelée image courante, c'est la même image que la première (le même arrière-plan de la pièce) sauf que le sujet d'intérêt qui est la personne y figure dedans. Donc, le résultat de la soustraction de ces deux images est la personne.

Inconvénients	Avantages
<ul style="list-style-type: none"> <li>• Contrairement à notre méthode, celle présentée par les auteurs de l'article [45] peut être inefficace lors de changements de lumière dans la pièce, par exemple : si l'image de référence a été prise le matin et l'image courante le soir, la soustraction des deux images fait apparaître d'autres pixels indésirables, qui peuvent être rajoutés à la silhouette de la personne, ce qui va certainement affecter par la suite la classification.</li> <li>• Leur méthode n'est pas efficace lorsqu'un changement survient dans l'arrière-plan de la pièce (déplacement ou ajout d'objets qui ne figurent pas dans l'image de référence), la soustraction des deux images nous donne d'autres objets qui n'ont aucun intérêt, vu que ce qui nous importe pour reconnaître la posture exercée par la personne est sa silhouette.</li> </ul>	<ul style="list-style-type: none"> <li>• Leur méthode consomme moins de ressource, par rapport à notre méthode qui nécessite l'utilisation d'un serveur plus ou moins puissant.</li> </ul>

**Tableau 16 Avantages et inconvénients de la méthode d'extraction de la silhouette du corps humain conçu par les auteurs de l'article [45]**

### 3.3.2. Le jeu de données

Dans le but d'obtenir un classificateur ou un modèle capable de reconnaître différentes postures humaine (*bend, lie, sit, stand*), les auteurs de l'article [45] ont élaboré un *dataset* constitué de 3216 postures (dont 804 images qui représentent la posture se tenir debout ou *stand*, 769 images représentant la posture s'asseoir ou *sit*, 810 images concernant la posture s'allonger ou *lie* et 833 images de la posture se pencher

ou *bend*). Selon les auteurs de l'article, 10 personnes (femmes et hommes) ont été sollicités pour la construction de ce jeu de données, ils ont simulé des postures dans différentes directions afin que le classificateur construit soit robuste pour reconnaître une posture capturée depuis n'importe quel angle de vue. Quelques échantillons de ce jeu de données sont illustrés dans la Tableau 17.

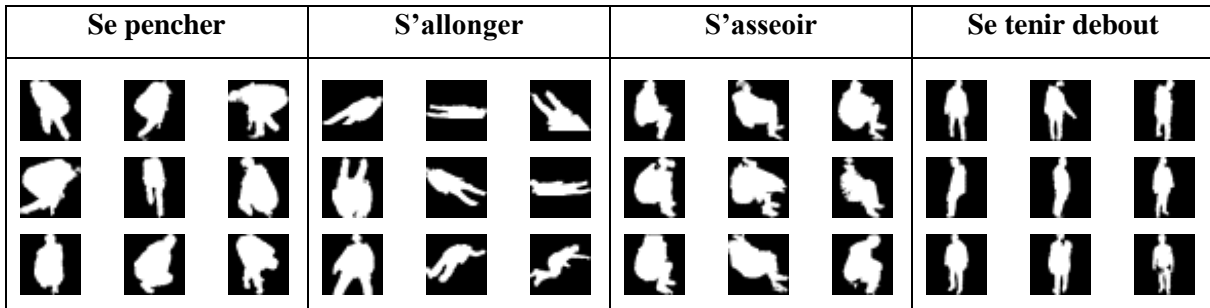


Tableau 17 Quelques images de postures contenues dans le jeu de données [45]

Inconvénients	Avantages
<ul style="list-style-type: none"> <li>• Contrairement à notre jeu de données, celui conçu par les auteurs de l'article [45] n'est pas consistant, car, on remarque l'absence d'images représentant quelques mouvements, par exemple : une personne qui marche, de plus, l'absence des mouvements transitoires entre les différentes postures, par exemple : lorsque la personne est debout puis elle s'assoie, lorsque la personne est assise puis elle se lève, lorsque la personne est debout ensuite elle se penche pour ramasser un objet, lorsque la personne est penchée ensuite elle se met debout.</li> <li>• Quelques images contenues dans leur <i>dataset</i> sont affectées par les points faibles (changements de lumière dans l'image courante ou la présence d'une ombre qui n'apparaît pas dans l'image de référence) de la méthode <i>codebook background subtraction</i> pour extraire la silhouette du corps humain.</li> </ul>	<ul style="list-style-type: none"> <li>• Pour construire le jeu de données, les auteurs de l'article [45] ont fait appel à 10 personnes de différents âges, sexes et corpulences, ce qui leur a permis d'obtenir un classificateur plus général et qui s'adapte le mieux à un milieu réel.</li> </ul>

Tableau 18 Avantages et inconvénients du jeu de données conçu par les auteurs de l'article [45]

Pour savoir qui des deux jeux de données est le meilleur pour construire un classificateur fiable en matière de reconnaissance de postures, nous avons entraîné et évalué notre réseau de neurones convolutif

sur le jeu de données conçu par les auteurs de l'article [45], puis, on a comparé entre les performances obtenues par ce classificateur et le nôtre qui est entraîné et évalué sur notre jeu de données.

Pour entraîner et évaluer notre réseau de neurones convolutif sur le jeu de données réalisé par les auteurs de l'article [45], on doit d'abord le modifier de tel sorte à ce qu'il comporte seulement deux classes qui sont : allongée (*lie*) et non allongée (*no lie*) pour l'adapter à notre réseau neuronal convolutif qui contient uniquement deux classes en sortie. Pour cela on déplace simplement les images contenues dans les classes *stand*, *bend* et *sit* dans la classe *no lie*, tandis que la classe *lie* n'est pas modifiée. La figure suivante illustre la nouvelle structure du jeu de données des auteurs de l'article [45] que nous avons modifié.

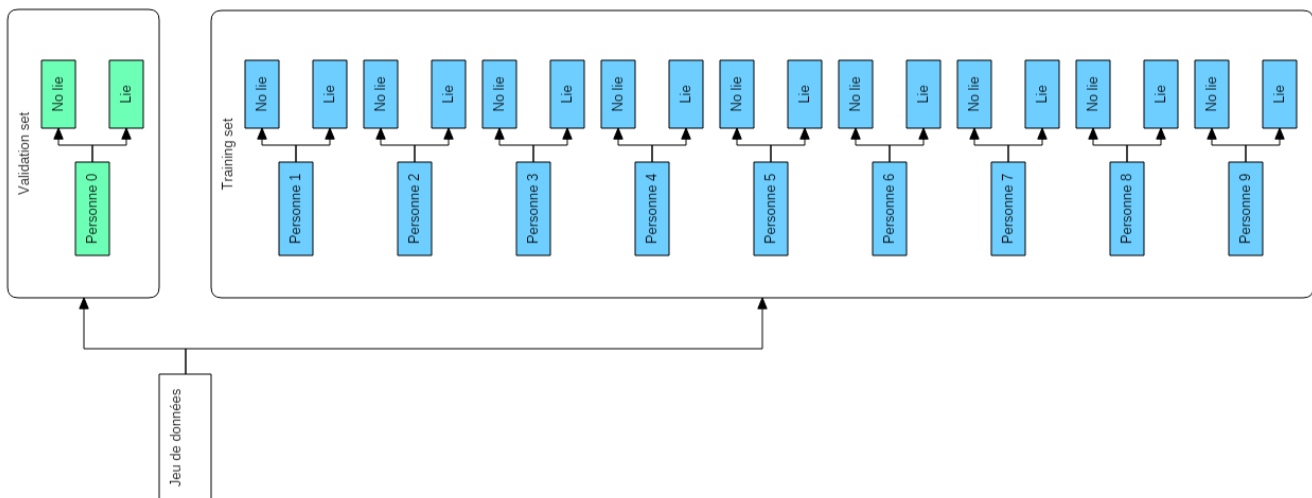


Figure 116 La nouvelle structure du jeu de données des auteurs de l'article [45] que nous avons modifié

On remarque que le jeu de données contient 10 dossiers où l'on retrouve deux sous-dossiers *lie* et *no lie*. En effet, chaque dossier représente une personne ayant contribué à la création du *dataset* et qui exerce des postures qui sont ensuite classées dans le sous-dossier *lie* ou *non lie*. Cette structure permet l'utilisation de la technique *cross-validation* qui consiste à sélectionner à tour de rôle une personne (qui représente l'ensemble de données de validation ou *validation set*) sur les dix existantes avec laquelle on évaluera notre réseau de neurones convolutif. Tandis que les personnes restantes seront utilisées pour entraîner notre réseau (elles représentent l'ensemble de données d'entraînement ou *training set*). Donc, on va obtenir 10 versions différentes du jeu de données, où nous allons entraîner et évaluer notre réseau neuronal convolutif sur chaque version, obtenant ainsi 10 classificateurs différents. En se basant sur la métrique *f-measure* on sélectionnera le meilleur modèle, qui va être ensuite comparé avec notre classificateur.

### Model 0

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.79	0.83	0.92	0.87	0.78

Tableau 19 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 0)

**Model 1**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.84	0.88	0.91	0.90	0.50

Tableau 20 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 1)

**Model 2**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.81	0.82	0.95	0.87	0.45

Tableau 21 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 2)

**Model 3**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.78	0.96	0.73	0.82	0.56

Tableau 22 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 3)

**Model 4**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.87	0.93	0.91	0.92	0.32

Tableau 23 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 4)

**Model 5**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.83	0.93	0.84	0.88	0.63

Tableau 24 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 5)

**Model 6**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.74	0.83	0.79	0.81	1.11

Tableau 25 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 6)

**Model 7**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.82	0.97	0.79	0.86	0.55

Tableau 26 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 7)

**Model 8**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.86	0.92	0.89	0.90	0.25

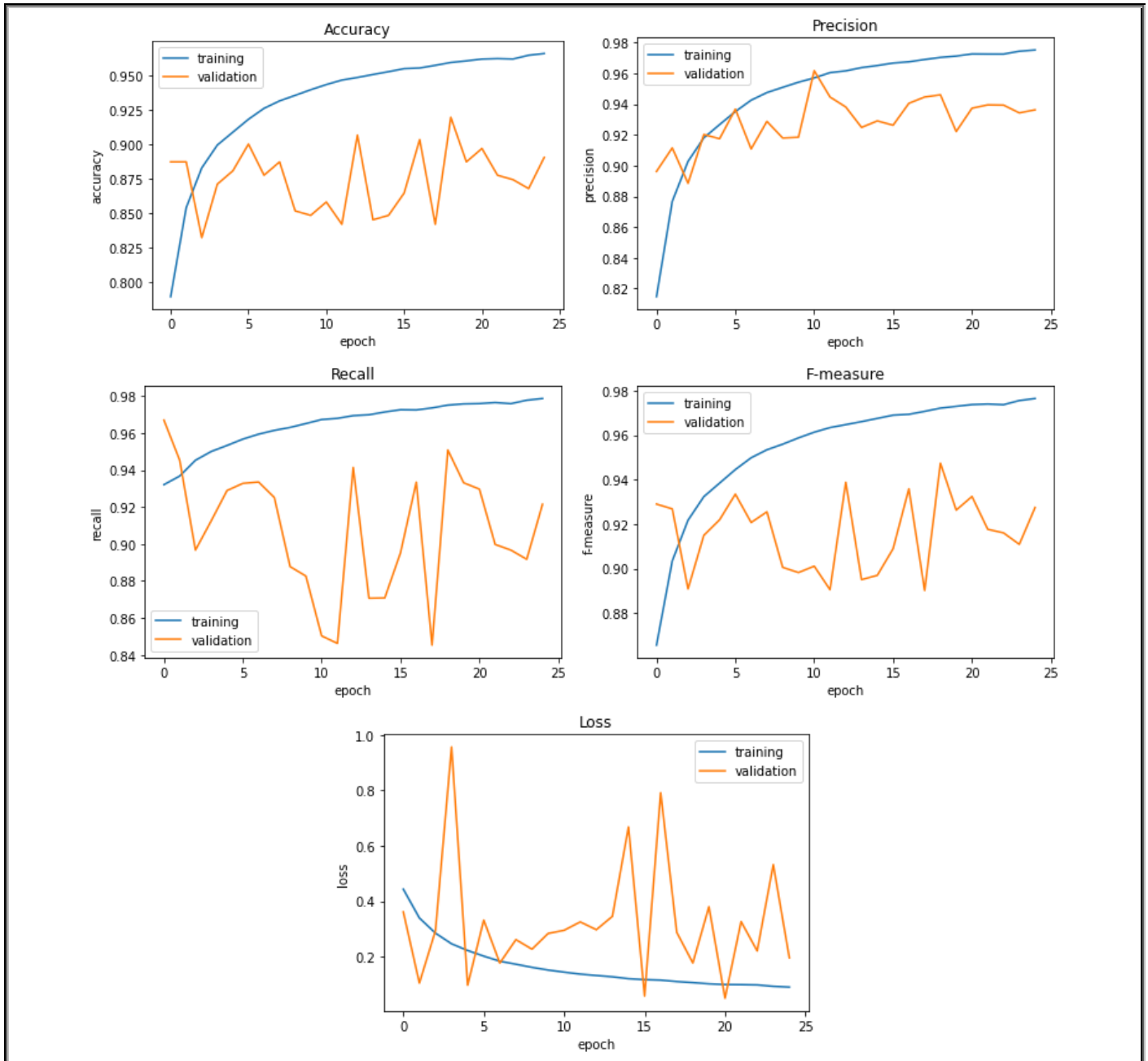
Tableau 27 Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 8)

**Model 9**

Métrique	Accuracy	Precision	Recall	F-measure	Loss
Valeur	0.77	0.94	0.75	0.83	0.68

**Tableau 28** Les valeurs des métriques obtenues après l'entraînement et l'évaluation de notre modèle sur le jeu de données des auteurs de l'article [45] pendant 25 epochs (Model 9)

**3.3.2.1. Graphes représentant l'évolution des métriques du modèle 4**



**Figure 117** Graphes représentant l'évolution des métriques du modèle 4 pendant 25 epochs

On peut constater que sur les 10 modèles, le modèle 4 est le plus performant, car, il obtient la valeur la plus élevée de la métrique *f-measure* qui est 0.92. Cependant, il reste bien en de ça par rapport à notre classificateur qui affiche une valeur plus élevée de la métrique *f-measure* (1.0), ce qui prouve que notre modèle est plus performant et par conséquent, notre jeu de données est bien meilleur que celui conçu par les auteurs de l'article [45].



### 3.3.3. Tests de reconnaissance de postures en utilisant le modèle 4

Afin de savoir qui des deux modèles (notre classificateur ou le modèle 4) est le plus fiable en matière de reconnaissance de postures, nous avons effectué des tests de notre système en remplaçant notre classificateur par le modèle 4.

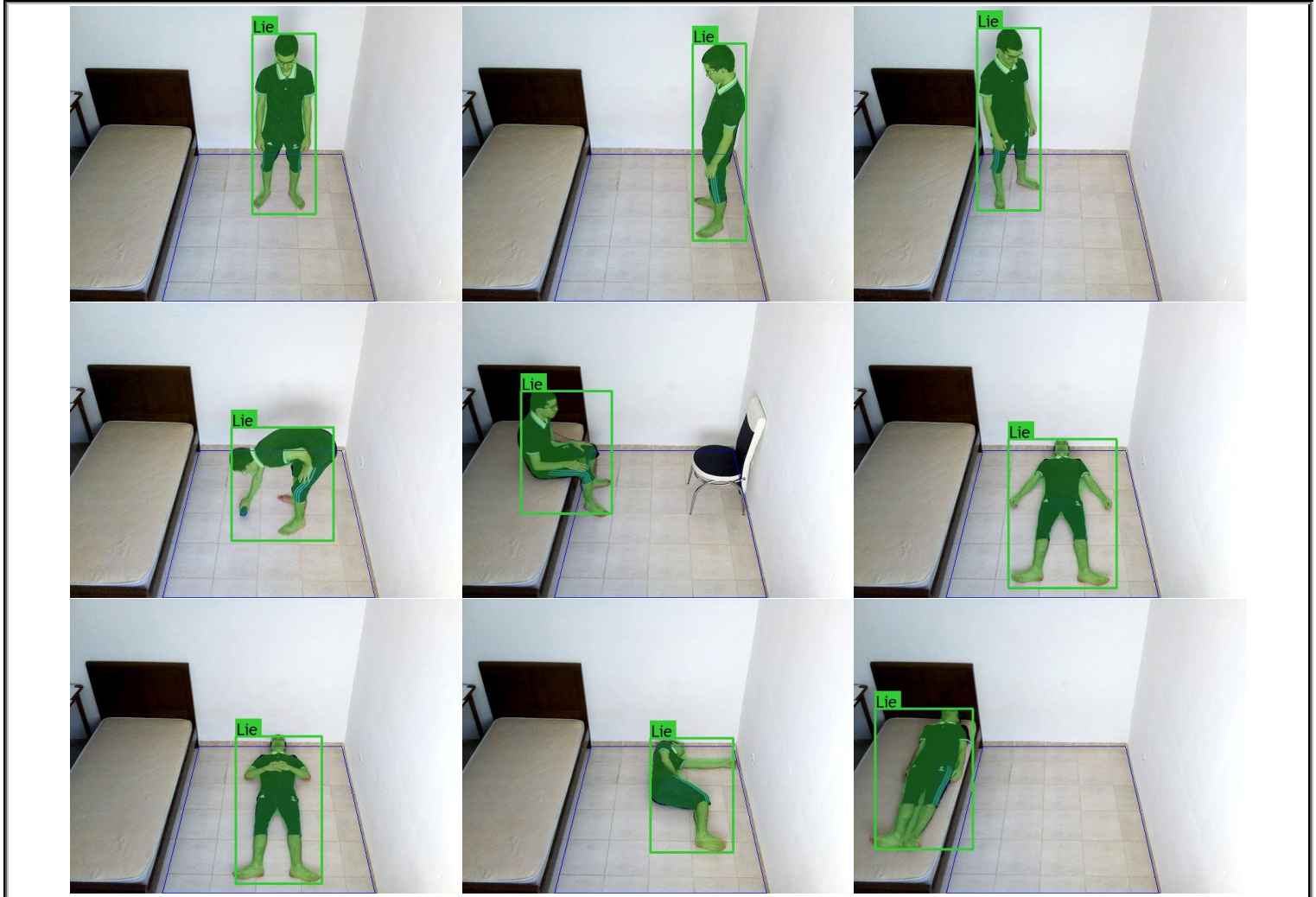


Figure 118 Tests de reconnaissance de postures en utilisant le modèle 4

Sur les mêmes tests que nous avons réalisé en utilisant notre classificateur, on peut clairement constater que le modèle 4 se trompe souvent, alors que notre classificateur ne se trompe jamais du fait qu'il a été entraîné sur notre jeu de données qui est bien plus consistant que celui des auteurs de l'article [45] sur lequel est entraîné le modèle 4. Donc, notre jeu de données est bien meilleur que celui des auteurs de l'article [45] pour concevoir un classificateur fiable en termes de reconnaissance de postures.

## 4. Environnements et Outils de développement

Dans cette partie, nous allons présenter l'ensemble des logiciels et langages de programmation utilisés pour le développement de notre système de détection des fuites de gaz et des chutes.

## ARDUINO IDE



Le logiciel de programmation des modules Arduino et NodeMCU, est une application Java, libre et multiplateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme à travers la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande. Le langage de programmation utilisé est le C++, lié à la bibliothèque de développement Arduino pour permettre l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plateformes Arduino à toute personne maîtrisant le C ou le C++. [46]

## XAMPP



XAMPP est une distribution d'Apache populaire dans le milieu de la programmation PHP. Sa facilité d'installation et d'utilisation ainsi que sa gratuité permet à n'importe quel utilisateur de se mettre à la programmation PHP. Le XAMPP contient la plupart des outils nécessaires pour reproduire le comportement d'un serveur web, à savoir :

- Apache : il s'agit du serveur web, c'est lui qui va réceptionner les requêtes HTTP et les étudier afin de présenter au visiteur la page demandée.
- MySQL : il s'agit du Système de Gestion de Bases de Données (le SGBD), il va permettre de sauvegarder les données de manière organisée sur le serveur
- PHP : ce module d'Apache va lui permettre d'interpréter les pages PHP.
- PhpMyAdmin : est une interface entre vous et vos données, il est fait pour simplifier l'administration de MySQL grâce à des pages web. [47]

## Visual Studio Code



Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et Mac OS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. [48]

## MySQL



MySQL, le plus populaire des serveurs de bases de données SQL Open Source, c'est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public



(applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. [49]

## PHP



PHP (officiellement, ce sigle est un acronyme récursif pour PHP Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML. Ce qui distingue PHP des langages de script comme le JavaScript, est que le code est exécuté sur le serveur, générant ainsi le HTML, qui sera ensuite envoyé au client. Le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques. [50]

## JavaScript



Le Javascript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. Ainsi le langage JavaScript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu. [51]

## Bootstrap 4

Bootstrap 4



Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option, il nous permet de créer des application Web responsive qui s'adaptent aux différentes tailles d'écran. [52]

## FRIZING



Fritzing est un logiciel libre, destiné aux non professionnels de l'électronique. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits. Le logiciel est disponible dans 16 langues dont le français. Il est adapté au débutant ou confirmés en électronique pour faire rapidement des circuits simples, et est également un bon outil didactique pour apprendre l'électronique par la pratique. [53]

## Google colab



*Google Colab* ou *Colaboratory* est un service cloud, offert par *Google* (gratuit), basé sur *Jupyter Notebook* et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud, sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. [54]

## Python



*Python* est un langage de programmation open source créé par le programmeur 'Guido van Rossum' en 1991. Il s'agit d'un langage de programmation interprété, qui ne nécessite donc pas d'être compilé pour fonctionner. Un programme 'interpréteur' permet d'exécuter le code *Python* sur n'importe quel ordinateur. Ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, ceci rend ce langage plus lent qu'un langage compilé comme le *C*. [55]

## 5. Conclusion

Dans ce chapitre, nous avons présenté les détails techniques liés à la mise en œuvre de notre système de détection des fuites de gaz et des chutes. Nous avons commencé par présenter la plateforme développée avec une description des différents modules et fonctionnalités qu'elle fournit. Ensuite, nous avons décrit l'ensemble des tests de fonctionnement que nous avons fait dans des conditions réelles avec une description des résultats obtenus qui ont été très satisfaisants, dans l'espoir que nous aurons l'occasion de faire l'expérience du système dans de véritables conditions à l'avenir. Ensuite, nous avons testé la capacité de notre système de détection des chutes, et plus précisément le classificateur de reconnaître différentes postures et de même les chutes, puis, nous avons comparé entre notre classificateur et celui des auteurs de l'article [45], les résultats obtenus ont été concluants, car sur la totalité des tests que nous avons effectués, notre modèle a prédit la bonne classe pour la posture exercée par la personne, contrairement à leur classificateur qui s'est trompé dans la plupart des tests, ce qui rend notre système plus fiable pour reconnaître si une personne a chuté ou non (moins de fausses alarmes). Enfin, nous avons conclu le chapitre par la présentation des technologies software que nous avons utilisées pour la réalisation de notre projet.

# **Conclusion générale**

Afin d'éradiquer les risques d'accidents domestiques liés à l'inhalation des gaz nocifs et les chutes, nous avons développé un prototype de système qui permet :

- Le contrôle et la surveillance en temps réel de l'état des différents gaz nocifs : monoxyde de carbone CO, gaz de ville Méthane, GPL et la Fumée) présents dans l'air ambiant et qui peuvent s'échapper des différents appareils du quotidien (cuisinière, chauffage, chauffe bain, etc..). Lors d'une perception d'une anomalie (un des gaz nocifs dépasse le seuil de tolérance qui lui est associé), le système intervient d'une manière automatique (à l'aide des actionneurs et des commutateurs) au niveau du domicile en effectuons les actions suivantes : couper l'alimentation électrique, fermer la vanne principale de gaz, ouvrir les fenêtres, actionner la ventilation et déclencher une alarme sonore et visuelle. Il envoi aussi une alerte via un appel aux services de secours et aux numéros de téléphones des occupants du domicile introduits préalablement dans le système. De plus, une notification est envoyée à une application *responsive* (web et mobile) utilisée par un administrateur distant appartenant aux services de la protection civile et les occupants du domicile,
- Aux occupants du domicile d'intervenir manuellement sur les actionneurs et les commutateurs qui contrôlent : l'alimentation électrique, la vanne principale de gaz, les fenêtres, la ventilation et l'alarme sonore,
- Aux occupants du domicile d'exploiter l'historique des changements des taux des gaz toxiques dans le temps (enregistrés sur la base de données) et le visualiser sous forme de graphe,
- À un administrateur distant de contrôler l'état de fonctionnement du matériel (carte ESP32, capteurs, actionneurs et commutateurs) installé au niveau du domicile des différents clients,
- À un administrateur distant de gérer les comptes des clients ayant installé ce système au niveau de leur domicile,
- À un administrateur distant de configurer les seuils à ne pas dépasser des différents gaz toxiques,
- De surveiller en temps réel les comportements des personnes à domicile avisant les membres de leur famille se trouvant en dehors de la maison ainsi que les services de secours en cas d'une situation de détresse,
- À l'aide d'une approche qui repose sur le *framework Mask R-CNN* (basée sur la segmentation d'instances dans une image) qui n'est pas affectée par les points faibles des méthodes de soustraction d'arrière-plan (changements de lumière dans la pièce, ajout ou déplacement ou suppression des objets dans la pièce), le système peut extraire uniquement la forme exacte de la silhouette d'une personne et donc d'assurer sa confidentialité. De plus, cette méthode permet au système d'assurer ces fonctions même durant la nuit (en utilisant une caméra assurant la vision nocturne),
- À l'aide du jeu de données que nous avons construit, le système peut reconnaître d'une manière fiable et robuste si la personne est allongée ou non,

- La surveillance en temps réel du domicile d'éventuelles intrusions de personnes malveillantes.

En perspective nous prévoyons :

- Ajouter d'autres capteurs comme : capteur de qualité d'air, capteur de gaz du butane (bouteille de gaz), etc.
- L'ajout d'un autre actionneur permettant au système d'intervenir d'une manière automatique ou manuelle sur la bouteille de gaz,
- En absence d'électricité le système doit pouvoir fonctionner également sur batterie,
- L'utilisation des capteurs industriels pour chaque gaz nocif pour améliorer la précision de captage,
- Étendre l'application en plusieurs langues (notamment en langue arabe) pour devenir approprié à tous les utilisateurs,
- Intégrer un système de géolocalisation (GPS) en vue de permettre aux services de secours d'intervenir le plus rapidement possible (prendre l'itinéraire le plus court et le moins encombré),
- La détection de la zone représentant le sol doit se faire de manière automatique,
- Étendre le jeu de données de telle sorte à ce qu'il comporte plus de postures et de mouvements, de plus, d'autres personnes seront sollicitées pour participer à sa création ce qui nous permettra d'obtenir un classificateur plus général qui s'adapte le mieux à un milieu réel,
- Intégrer un système permettant depuis une caméra de reconnaître lorsqu'une personne souffre de problèmes respiratoires.

## Bibliographie

- [1] <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> [Online] [Accessed: 11-March-2020]
- [2] <https://www.jumia.dz/detecteur-de-gaz-blanc-sans-marque-mpg953.html> [Online] [Accessed: 11-March-2020]
- [3] <https://www.jumia.dz/otio-detecteur-monoxyde-de-carbone-lm-201a-otio.-148813.html> [Online] [Accessed: 11-March-2020]
- [4] <https://www.jumia.dz/sans-marque-detecteur-de-monoxyde-de-carbone-avec-affichage-lcd-blanc-155210.html> [Online] [Accessed: 11-March-2020]
- [5] SHRIVASTAVA, Ashish, PRABHAKER, Ratnesh, KUMAR, Rajeev, *et al.* GSM based gas leakage detection system. *International Journal of Emerging Trends in Electrical and Electronics (IJETEE-ISSN : 2320-9569)*, 2013, vol. 3, no 2.
- [6] VARMA, Asmita, PRABHAKAR, S., et JAYAVEL, Kayalvizhi. Gas leakage detection and smart alerting and prediction using IoT. In : *2017 2nd International Conference on Computing and Communications Technologies (ICCT)*. IEEE, 2017. p. 327-333.
- [7] SHARMA, Manaswi, TRIPATHI, Diksha, YADAV, Narendra Pratap, *et al.* Gas leakage detection and prevention kit provision with IoT. *Gas*, 2018, vol. 5, no 02.
- [8] BHOIR, Shruti, GOREGAONKAR, Snehanjali, et SATRE, Shilpa. IoT based gas detection system. *Int J Eng Sci*, 2017, vol. 10251.
- [9] ANANDHAKRISHNAN, S., NAIR, Deepesh, RAKESH, K., *et al.* IOT Based Smart Gas Monitoring System. *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, 2017, p. 82-87.
- [10] LI, Yanjun, CHEN, Gan, SHEN, Yueyun, *et al.* Accelerometer-based fall detection sensor system for the elderly. In : *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*. IEEE, 2012. p. 1216-1220.
- [11] ALWAN, Majd, RAJENDRAN, Prabhu Jude, KELL, Steve, *et al.* A smart and passive floor-vibration based fall detector for elderly. In : *2006 2nd International Conference on Information & Communication Technologies*. IEEE, 2006. p. 1003-1007.
- [12] ROUGIER, Caroline, MEUNIER, Jean, ST-ARNAUD, Alain, *et al.* Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on circuits and systems for video Technology*, 2011, vol. 21, no 5, p. 611-622.
- [13] ANDERSON, Derek, KELLER, James M., SKUBIC, Marjorie, *et al.* Recognizing falls from silhouettes. In : *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006. p. 6388-6391.
- [14] ROUGIER, Caroline, MEUNIER, Jean, ST-ARNAUD, Alain, *et al.* Monocular 3D head tracking to detect falls of elderly people. In : *2006 international conference of the IEEE engineering in medicine and biology society*. IEEE, 2006. p. 6384-6387.
- [15] SIXSMITH, Andrew et JOHNSON, Neil. A smart sensor to detect the falls of the elderly. *IEEE Pervasive computing*, 2004, vol. 3, no 2, p. 42-47.

- [16] <https://www.anxiete.fr/trouble-panique/attaque-de-panique/> [Online] [Accessed: 12-March-2020]
- [17] BALAKRISHNAN, Guha, DURAND, Fredo, et GUTTAG, John. Detecting pulse from head motions in video. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013. p. 3430-3437.
- [18] <https://www.topsante.com/medecine/accidents/premiers-secours/tout-savoir-sur-la-detresse-respiratoire-244445> [Online] [Accessed: 12-March-2020]
- [19] <https://becominghuman.ai/an-introduction-to-machine-learning-33a1b5d3a560> [Online] [Accessed: 15-March-2020]
- [20] RASHID, Tariq. Make your own neural network: a gentle journey through the mathematics of neural networks, and making your own using the Python computer language. CreateSpace Independent Publ., 2016.
- [21] Vidéos du cours Deep Learning A-Z™ : Hands-On Artificial Neural Networks (<https://www.udemy.com/course/deeplearning/>) [Online] [Accessed: 15-March-2020]
- [22] <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> [Online] [Accessed: 16-March-2020]
- [23] <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-4-full-connection> [Online] [Accessed: 16-March-2020]
- [24] <https://mc.ai/deeplearning-series-convolutional-neural-networks/> [Online] [Accessed: 16-March-2020]
- [25] <https://www.jeremyjordan.me/convolutional-neural-networks/> [Online] [Accessed: 16-March-2020]
- [26] GIRSHICK, Ross, DONAHUE, Jeff, DARRELL, Trevor, *et al.* Rich feature hierarchies for accurate object detection and semantic segmentation. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014. p. 580-587.
- [27] GIRSHICK, Ross. Fast r-cnn. In : *Proceedings of the IEEE international conference on computer vision*. 2015. p. 1440-1448.
- [28] REN, Shaoqing, HE, Kaiming, GIRSHICK, Ross, *et al.* Faster r-cnn: Towards real-time object detection with region proposal networks. In : *Advances in neural information processing systems*. 2015. p. 91-99.
- [29] <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>
- [30] CORTES, Corinna et VAPNIK, Vladimir. Support-vector networks. *Machine learning*, 1995, vol. 20, no 3, p. 273-297.
- [31] HE, Kaiming, GKIOXARI, Georgia, DOLLÁR, Piotr, *et al.* Mask r-cnn. In : *Proceedings of the IEEE international conference on computer vision*. 2017. p. 2961-2969.
- [32] LIN, Tsung-Yi, MAIRE, Michael, BELONGIE, Serge, *et al.* Microsoft coco: Common objects in context. In : *European conference on computer vision*. Springer, Cham, 2014. p. 740-755.

- [33] <https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46> [Online] [Accessed: 23-March-2020]
- [34] <https://www.arduino-france.com/review/arduino-uno/> [Online] [Accessed: 26-March-2020]
- [35] <https://www.amazon.fr/Raspberry-Pi-Carte-M%C3%A8re-Model/dp/B01CD5VC92> [Online] [Accessed: 26-March-2020]
- [36] <https://fr.wikipedia.org/wiki/NodeMCU> [Online] [Accessed: 26-March-2020]
- [37] <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/> [Online] [Accessed: 27-March-2020]
- [38] <https://components101.com/motors/mg996r-servo-motor-datasheet> [Online] [Accessed: 28-March-2020]
- [39] <https://ardubotics.eu/en/relays-controllers/1100-dc-12v-1-channel-high-level-trigger-relay-module.html> [Online] [Accessed: 28-March-2020]
- [40] <https://www.instructables.com/id/How-to-use-a-Buzzer-Arduino-Tutorial/> [Online] [Accessed: 28-March-2020]
- [41] <https://binarytech-dz.com/produit/arduino-raspberry/arduino/module-gsm-gprs-sim800l-avec-antenne-pour-arduino/> [Online] [Accessed: 28-March-2020]
- [42] <https://arduino.blaisepascal.fr/pont-en-h-l298n/> [Online] [Accessed: 29-March-2020]
- [43] <https://euro-makers.com/fr/refroidissement/1300-ventilateur-50x50x20mm-12v-0-15a-25-m3-h-3701172902525.html> [Online] [Accessed: 02-April-2020]
- [44] KINGMA, Diederik P. et BA, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [45] YU, Miao, GONG, Liyun, et KOLLIAS, Stefanos. Computer vision based fall detection by a convolutional neural network. In : *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 2017. p. 416-420.
- [46] <https://www.arduino.cc/en/main/software> [Online] [Accessed: 10-August-2020]
- [47] <https://www.apachefriends.org/fr/index.html> [Online] [Accessed: 10-August-2020]
- [48] <https://code.visualstudio.com/> [Online] [Accessed: 10-August-2020]
- [49] <https://www.mysql.com/fr/> [Online] [Accessed: 10-August-2020]
- [50] <https://www.php.net/> [Online] [Accessed: 10-August-2020]
- [51] <https://www.w3schools.com/js/> [Online] [Accessed: 10-August-2020]
- [52] <https://getbootstrap.com/> [Online] [Accessed: 11-August-2020]
- [53] <https://fritzing.org/> [Online] [Accessed: 11-August-2020]
- [54] <https://ledatascientist.com/google-colab-le-guide-ultime/> [Online] [Accessed: 11-August-2020]
- [55] <https://www.lebigdata.fr/python-langage-definition> [Online] [Accessed: 11-August-2020]



# **Annexe : Fonctionnement d'un réseau de neurones**

Il est important de savoir comment les signaux progressent des entrées à travers les couches pour obtenir les sorties, pour que nous sachions ce qui se passe réellement à l'intérieur d'un réseau de neurones. Nous allons donc essayer de travailler avec un réseau de neurones plus petit avec seulement 2 couches, chacune avec 2 neurones, comme indiqué ci-dessous.

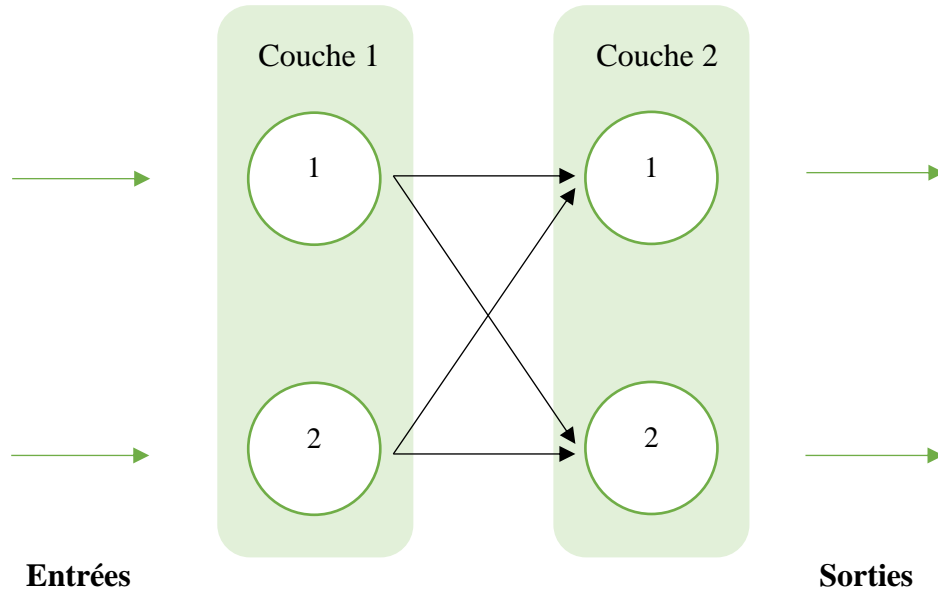


Figure 119 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds

Supposons que les deux entrées soient 1.0 et 0.5. Ce qui suit montre ces entrées introduites dans le réseau neuronal.

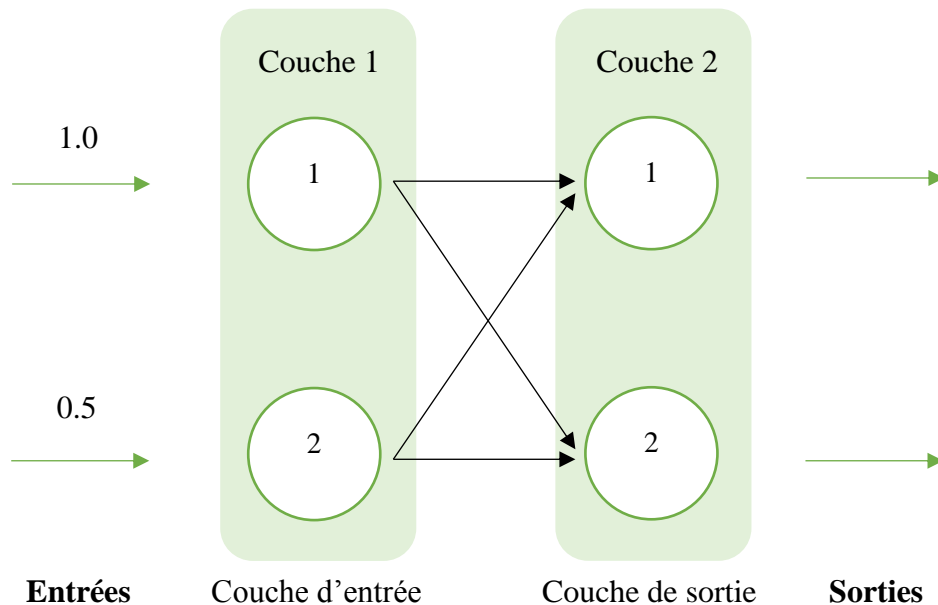


Figure 120 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds (avec les entrées 1.0 et 0.5)

Comme précédemment, chaque nœud transforme la somme des entrées en sortie à l'aide d'une fonction d'activation. Nous utiliserons également la fonction sigmoïde  $y = \frac{1}{1+e^{-x}}$  que nous avons vu précédemment, où  $x$  est la somme des signaux entrants vers un neurone, et  $y$  est la sortie de ce neurone.

Et les poids ? Par quelle valeur devraient-ils commencer ? Allons-y avec quelques poids aléatoires :  $W_{1,1} = 0.9$ ,  $W_{1,2} = 0.2$ ,  $W_{2,1} = 0.3$ ,  $W_{2,2} = 0.8$ .

Les valeurs de départ aléatoires ne sont pas une si mauvaise idée, ils vont être améliorés avec chaque exemple que le réseau de neurone va apprendre.

La première couche de nœuds dans un réseau de neurones est la couche d'entrée (*input layer*), elle ne fait rien d'autre que représenter les signaux d'entrée (aucun calcul ne se fait au niveau de cette couche). Autrement dit, les nœuds d'entrée n'appliquent pas de fonction d'activation à l'entrée. Vient ensuite la deuxième couche où nous devons faire des calculs. Pour chaque nœud de cette couche, nous devons déterminer l'entrée combinée  $x$  qui constituera l'entrée de notre fonction d'activation (fonction sigmoïde)  $y = \frac{1}{1+e^{-x}}$ .

Le  $x$  dans cette fonction est l'entrée combinée dans un nœud. Cette combinaison n'est que la somme des sorties des nœuds connectés dans la couche précédente, mais multipliées par les poids de liaison. Le diagramme suivant est comme celui que nous avons vu précédemment mais inclut maintenant la nécessité de multiplier les signaux entrants avec les poids de liaison.

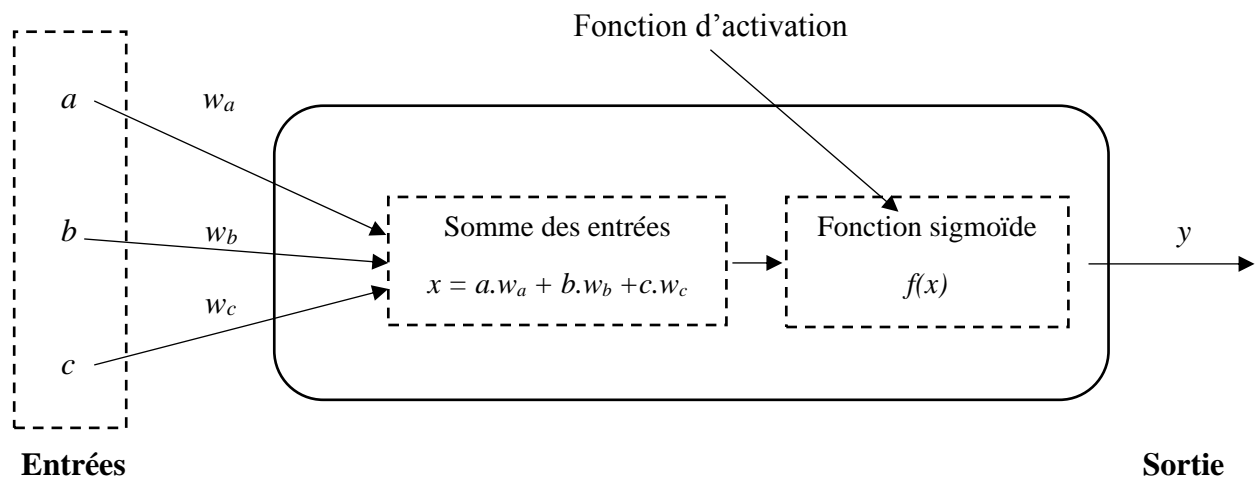


Figure 121 Le fonctionnement d'un neurone artificiel (avec les poids des liens reliés à ce neurone)

Concentrons-nous d'abord sur le nœud 1 de la couche 2. Les deux nœuds de la première couche d'entrée  $y$  sont connectés. Ces nœuds d'entrée ont des valeurs de 1.0 et 0.5. Le lien du premier nœud a un poids de 0.9 qui lui est associé. Le lien du second a un poids de 0.3. L'entrée combinée  $x$  est donc :

$$x = (\text{sortie du premier nœud} * \text{poids du lien}) + (\text{sortie du deuxième nœud} * \text{poids du lien})$$

$$x = (1.0 * 0.9) + (0.5 * 0.3) = 0.9 + 0.15 = 1.05$$

Si nous n'avons pas modéré le signal, nous aurions une addition très simple des signaux  $1.0 + 0.5$ , mais nous ne le voulons pas. Ce sont les poids qui font l'apprentissage dans un réseau de neurones car ils sont affinés de manière itérative pour donner de meilleurs résultats. Nous avons donc maintenant  $x = 1.05$  pour l'entrée modérée combinée dans le premier nœud de la deuxième couche. Nous pouvons maintenant

calculer la sortie de ce nœud en utilisant la fonction d'activation (fonction sigmoïde)  $y = \frac{1}{1+e^{-x}}$ , on obtient donc :  $y = 0.7408$ .

Nous avons maintenant une sortie d'un des deux nœuds de sortie du réseau. Reprenons le calcul avec le nœud restant qui est le nœud 2 dans la deuxième couche. L'entrée modérée combinée  $x$  est :

$$x = (\text{sortie du premier nœud} * \text{poids du lien}) + (\text{sortie du deuxième nœud} * \text{poids du lien})$$

$$x = (1.0 * 0.2) + (0.5 * 0.8) = 0.2 + 0.4 = 0.6$$

Alors maintenant, nous avons  $x$ , nous pouvons calculer la sortie du nœud en utilisant la fonction d'activation sigmoïde,  $y = 1 / (1 + 0.5488) = 1 / (1.5488)$ . Donc  $y = 0.6457$ . Le diagramme suivant montre les sorties du réseau que nous venons de calculer.

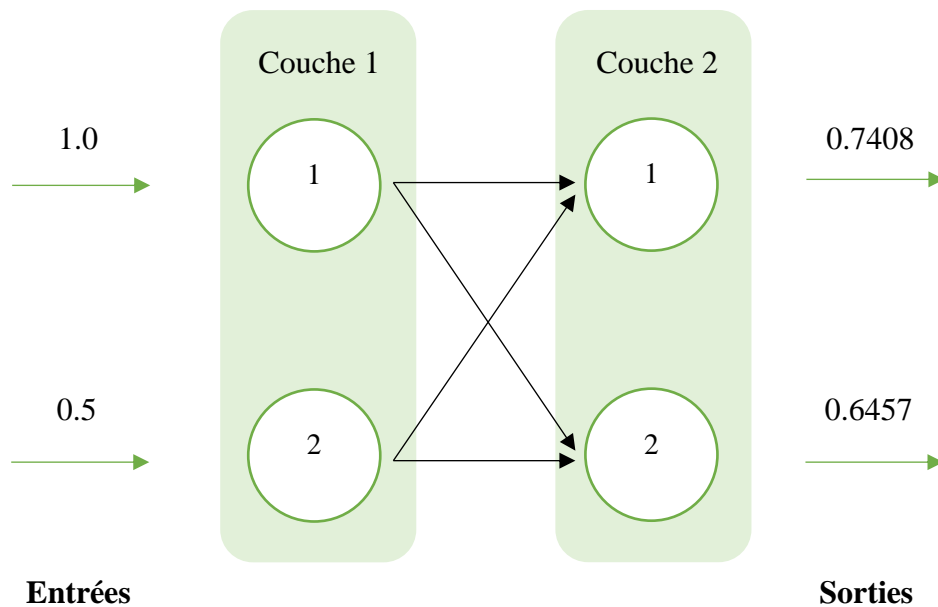


Figure 122 Un réseau de neurones artificiels contenant deux couches chacune avec deux nœuds (avec les sorties calculées)