

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingénierat
Département : Electronique
Domaine : Sciences et Techniques
Filière : Automatique
Spécialité : Automatique et informatique industriel

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**Detection des Masques Facial à L'aide de L'intelligence
Artificielle**

Présenté par : *Riade Boughaba*

Encadrant : *Bekaik Mounir*

Dr

Université de badji Mokhtar annaba

Jury de Soutenance :

Saidi Med Larbi	Pr	Université de badji Mokhtar annaba	Président
Bekaik Mounir	Dr	Université de badji Mokhtar annaba	Encadrant
Ramdani Messouad	Pr	Université de badji Mokhtar annaba	Examineur

Année Universitaire : 2020/2021

INTRODUCTION :	2
CHAPITRE 1:	4
1 - Bête à cornes :	5
2 - Diagramme de Pieuvre :	6
3 - Diagramme de Fast :	7
4 - Les Normes:	9
5 - L'approche de "Face Mask Detection Algorithm":	10
6 - Diagramme d'étude des cas :	11
7 - Diagramme de séquence :	12
CHAPITRE 2:	13
1 - Les réseaux neurones:	14
2 - Algorithm CNN:	14
2 - 1 - 1 Fonction d'activation linéaire:	15
2 - 1 - 2 Fonction d'activation non linéaire:	16
2 - 1 - 2 - 1 Sigmoid or Logistic Activation Function:	17
2 - 1 - 2 - 2 Tanh or hyperbolic tangent Activation Function:	18
2 - 1 - 2 - 3 Fonction d'activation ReLU:	19
3 - YOLO Algorithm (You Only Look Once):	26
4 - Les Techniques de site web:	30
5 - Base Des Données SQL:	37
CHAPITRE 3:	39
1 - Les captures:	40
2 - La Base Des Données:	43
3 - Le site de paiement:	45
4 - Le processus de paiement:	49
Conclusion:	52

Liste des Figures:

Figure 1 : Présentation bête à corne	9
Figure 2 : Présentation Diagramme de Pieuvre	10
Figure 3: Diagramme étude de cas	16
Figure 4 : Diagramme de séquence	17
Figure 5 - Representation des Réseaux neurones	19
Figure 6 - Fonction d'activation Linéaire	21
Figure 7 - Fonction d'activation non-linéaire	22
Figure 7.1 Fonction d'activation sigmoid	23

Figure 7.2 Fonction d'activation Tanh	23
Figure 7-3 Fonction d'activation sigmoid et ReLU	24
Figure 8 - Fonction de configuration pour le CNN	25
Figure 9 - Boucle "While" principale	26
Figure 10 - Fonction du processus	27
Figure 11 - Conditions des décision	27
Figure 12 - Fonction de reconnaissance du visage	28
Figure 13-1 Décision à visage unique	29
Figure 13-2 Décisions multi-visages	29
Figure 14 - Processus d'encodage de YOLO	31
Figure 15 - Phase de segmentation de YOLO	31
Figure 16 - YOLO avant et après la suppression non-maximale	32
Figure 17 - Phase de l'intersection YOLO	32
Figure 18 - Architecture interne de YOLO	33
Figure 19 - Principaux liens du site web	35
Figure 20 - Fonction de l'index principal du site web	36
Figure 21 - Code HTML de la page de connexion du site web	37
Figure 22 - Code HTML pour la page principale	39
Figure 23 - La fonction de paiement	40
Figure 24 - Table SQL	41
Figure 25 - Code de détection principal	44
Figure 26 - Cas d'absence de masque	45
Figure 27 - Cas de détection de masque	45
Figure 28 : Interface utilisateur graphique de l'éditeur SQL Phpmyadmin	47
Figure 29 - Page de connexion	48
Figure 30 - Code HTML de la page de connexion	50
Figure 31 - Page de paiement	50
Figure 32 - Code HTML de la page de paiement	51
Figure 33 - Notification par e-mail	52

Liste des Tableaux:

Tableau 1: Diagramme de Fast	41
Tableau 2 : Principales requêtes SQL	46
Tableau 3 - Architecture de la table principale des citoyens	46
Tableau 4 - Table de liste noire	55

Résumé:

Ce projet consiste à présenter une approche de détection des masques en se basant sur l'intelligence artificielle où l'algorithme CNN de reconnaissance faciale paraît le mieux placé pour répondre aux exigences attendues. Afin de pouvoir interagir entre les données enregistrées et les informations en temps réelles, nous avons créé deux bases de données dédiées pour les renseignements des citoyens et l'autre concerne les citoyens non porteurs du masque. Un site a été créé pour payer la fraude ainsi qu'un accusé de réception est envoyé aux concernés.

Mots clés : intelligence artificielle ; CNN ; Masque ; Base de données SQL ; détection.

Abstract :

This project consists of presenting an approach of mask detection based on artificial intelligence where the CNN facial recognition algorithm seems best placed to meet the expected requirements. In order to be able to interact between the recorded data and the information in real time, we have created two databases, one dedicated for citizen information and the other for non-masked citizens. A site has been created to pay for the fraud and an acknowledgment of receipt is sent to those concerned.

Keywords: artificial intelligence, CNN, Mask, SQL database, detection.

ملخص:

في الختام ، يهدف هذا المشروع إلى الكشف عن الأشخاص الذين ليس لديهم قناع للوجه في الأماكن العامة. تكتشف الخوارزمية أولاً وجه هذا الشخص ، ثم تتحقق مما إذا كان قد تم وضع القناع أم لا ، وإذا كان هناك قناع ، فلن تتخذ الخوارزمية أي إجراء آخر ، وإذا لم يكن هناك قناع ، تتعرف الخوارزمية على الوجه وتحفظ هذا القومي معرف القسيس. إلى القائمة السوداء ، يمكن للشخص دفع غرامة عدم ارتداء القناع مع موقع الويب حيث يجب عليه أولاً المصادقة ثم تقديم الخاصة به CIB معلومات بطاقة ائتمان

Introduction :

Covid-19 la pandémie qui menace le monde et a forcé de nombreux peuples à renoncer à leur vie normale, ce virus a menacé la vie de nombreuses personnes et en a tué beaucoup d'autres en raison de sa nature et de son fonctionnement en attaquant le système respiratoire chez l'homme causant des problèmes respiratoires et donc la mort.

À partir de là, les gens et les gouvernements réalisent le rôle important de la protection et décident donc de prendre des mesures en forçant les gens à faire certains actes qui peuvent réduire l'infection et éventuellement sauver des vies, comme le port de masques.

La problématique provient du fait que vous ne pouvez pas contrôler tout le monde et vous assurer du port des masques au moins pour les humains, de ce fait ; des solutions apparaissent intéressantes aux gouvernements qui porte sur la mise en œuvre de l'intelligence artificielle[1] à l'aide d'algorithmes[7] spécifiques pouvant répondre parfaitement au besoin attendu par les gouvernements.

Ces algorithmes[7] sont différents et chacun présente des avantages et des inconvénients, à partir de ces algorithmes[7], nous pouvons citer les suivants :

1. Algorithmes basés sur la géométrie / basés sur des modèles
2. Algorithmes basés sur l'apparence / basés sur un modèle
3. Algorithmes basés sur des modèles / statistiques / réseaux de neurones
4. Algorithmes de correspondance de modèles
5. Algorithmes d'analyse des composants indépendants [ICA]
6. Réseaux de neurones avec algorithmes de filtres Gabor
7. Réseaux de neurones et algorithmes de modèles de Markov cachés
8. Algorithmes de réseaux de neurones flous
9. Algorithmes de reconnaissance FisherFaces
10. Algorithmes CNN (Conventional Neural Network)

Pour ce projet, l'algorithme[7] CNN[19] a été sélectionné et paraît le meilleur placé surtout dans la détection d'objets. Cette technologie de suivi CNN[19] appelée YOLO (vous ne regardez qu'une seule fois) qui implémente certaines fonctions chez CNN[19] et qui augmente les performances en obtenant un pourcentage plus élevé de la prédiction.

De plus, cet algorithme[7] représente un avantage intéressant en termes de la capacité de gérer des ensembles de données volumineux et complexes sans aucun problème ou ce que l'on appelle Big-Data, cet algorithme[7] offre un moyen d'analyse d'énormes données sans avoir besoin de créer une structure de données complexe à partir de zéro et sans le recours aux anciennes méthodes pour gérer ce type de méga données. De plus, le CNN[19] est très flexible lorsqu'il s'agit de détecter des objets, donc si des fonctionnalités supplémentaires veulent être ajoutées, aucun problème ne sera détecté en raison de sa structure et de sa hiérarchie qui ajoutent des points positifs à la raison pour laquelle nous devrions utiliser l'algorithme[7] CNN[19]. Un autre point est que CNN[19] est capable de sélectionner les meilleures données et d'éliminer les mauvaises en raison de ses fonctionnalités. C'est le problème du Big Data[2] car il est énorme et il faudra probablement des années pour naviguer, nettoyer et structurer les données par les humains, donc CNN[19] résout également ce problème facilement.

Le flux de travail de cet algorithme[7] est basique, l'algorithme[7] doit d'abord collecter certaines informations sur les personnes telles que le nom, le nom de famille, la date de naissance, l'adresse, l'e-mail et le numéro d'identification national.

Deuxièmement, ces informations doivent être archivées dans la base de données SQL, chacune avec son nom de colonne pertinent, cela est normalement fait par quelqu'un qui est capable de gérer les bases de données SQL, le serveur et les requêtes.

Troisièmement, l'algorithme[7] doit établir une connexion avec la base de données SQL avec certaines bibliothèques python et à partir de ce point l'algorithme[7] fera tout automatiquement, de la détection de la personne, à l'analyse complète de l'image pour détecter et sélectionner le visage, faire un traitement pour éliminer les détails inutiles et ne garder que le visage.

Ensuite, ces images seront stockées dans une certaine base de données en étiquetant chaque image avec son identifiant correspondant pour la transmettre plus tard au processus de reconnaissance qui gèrera le reste, la reconnaissance se fait en déterminant quel visage appartient à qui en validant l'image avec la base de données, si le visage est trouvé, l'algorithme[7] obtiendra l'identifiant et un traitement supplémentaire sera effectué. Si la personne porte un masque, aucune autre opération ne sera exécutée, mais s'il n'y a pas de masque, l'identifiant sera sauvegardé dans une table spéciale qui contient tous ceux qui enfreignent la loi et ils doivent payer des frais pour cet acte. L'algorithme[7] analysera et affichera le nom complet de cette personne. Plus tard, la personne doit accéder au site Web afin de payer le montant et cela se fait en fournissant son numéro d'identification national et sa date de naissance comme mot de passe. Ensuite, il lui suffit de payer avec sa carte CIB en fournissant le numéro de cette carte, le CVV et la date d'expiration ; puis il recevra un accusé par e-mail si le paiement est réussi, cette étape consiste également à assurer une protection supplémentaire en facilitant le paiement des frais sans avoir besoin de quitter la maison.

De cette façon, nous pouvons assurer la sécurité des personnes sans avoir besoin de faire fonctionner des humains mais plutôt un système intelligent 24/7 et ne manquez personne. Cette approche est actuellement utilisée partout sur terre mais surtout en Chine. La Chine surveille chaque citoyen partout et ils le font avec cette approche, un algorithme[7] de renseignement qui peut détecter et reconnaître les personnes, lancer des caméras publiques et opérer certains résultats. De cette manière, veillez à l'application de la loi et chacun respectera la loi.

CHAPITRE 1: Analyse des besoins et Analyse fonctionnelle.

1 - Bête à cornes :

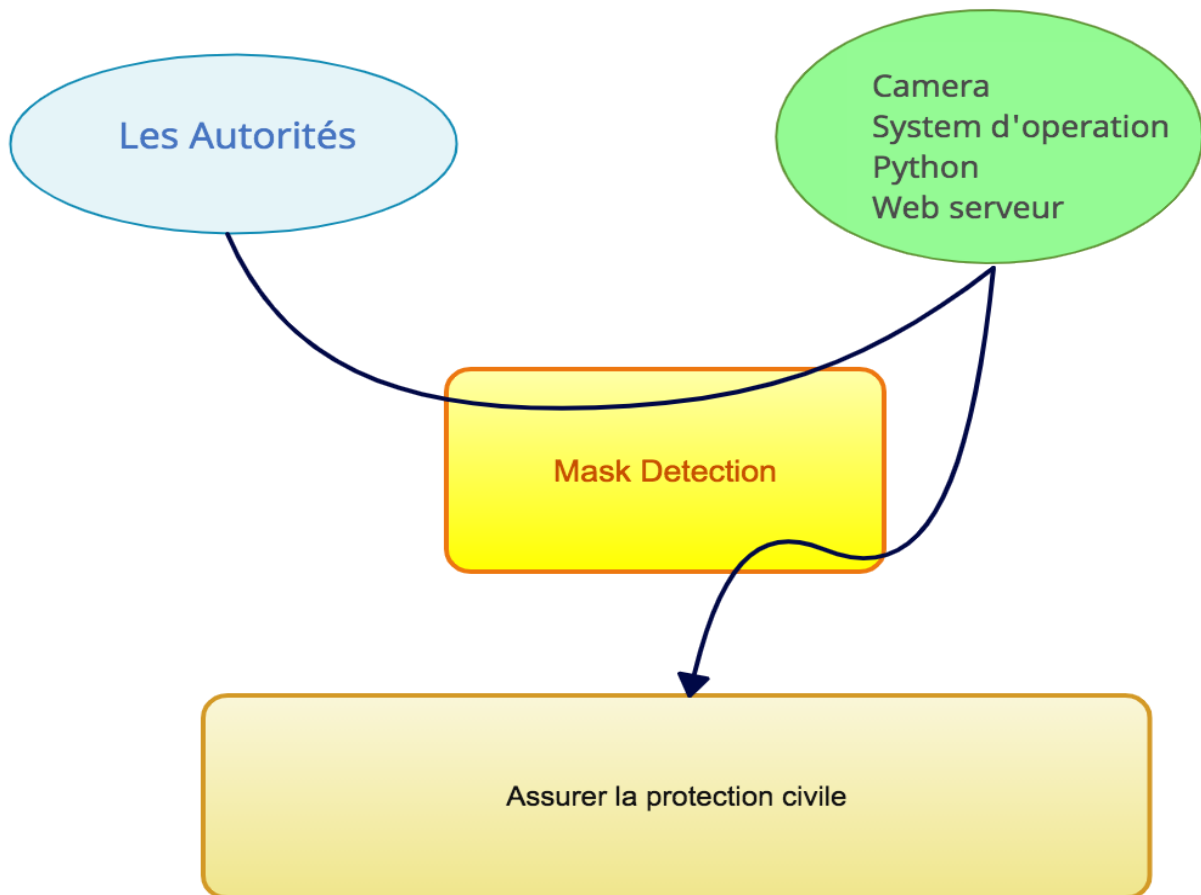


Figure 1 : Présentation bête à corne

2 - Diagramme de Pieuvre :

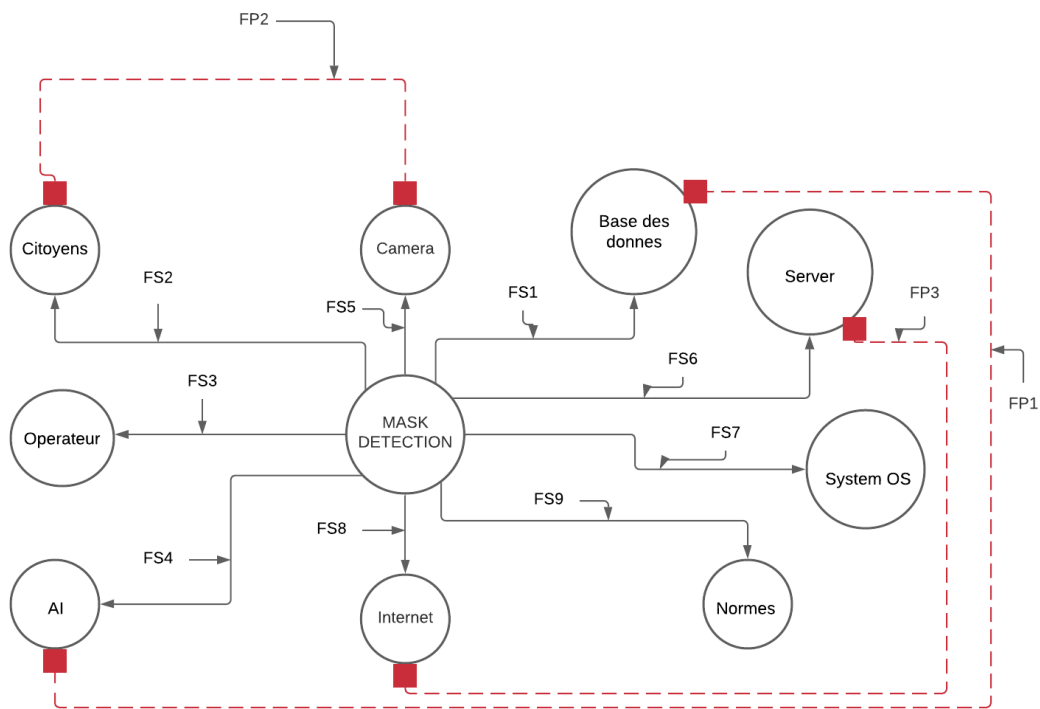


Figure 2 : Présentation Diagramme de Pieuvre

FS : Fonction de service

FP : Fonction principale

FS1 : Sauvegarde des informations des citoyens

FS2 : Protéger les citoyens

FS3 : Programmer et commander

FS4 : Reconnaître et sauvegarder dans la base des données

FS5 : Détecter les personnes

FS6 : Permettre l'accès au site

FS7 : Utiliser les systèmes OS

FS8 : Se connecter au site pour payer les amendes

FS9 : Respecter les normes

FP1 : Accéder et sauvegarder les informations des citoyens

FP2 : Permettre d'avoir les images des personnes

FP3 : Permettre aux citoyens d'accéder au site web

3 - Diagramme de Fast :

	Critère	Solution technologique
FP1	<p>FP11: Commencer le traitement. FP12: Exécuter le programme. FP13: Afficher le résultat. FP 131: Si rouge, afficher nom et prénom du personne. FP 132: Si vert afficher "Mask" dans le cadre. FP14: Donner la liste des citoyens non porteurs du Mask. FP 141: Donner les noms et les prénoms. FP 142: Donner l'adresse et l'heure de la fraude.</p>	<ul style="list-style-type: none"> ● Python ● SQL Server ● Bibliothèques
FP2	<p>FP21: Détecter les personnes via la caméra. FP22: Faire une numérisation pour détecter les visages des personnes. FP23: Traiter les résultats de la détection. FP24: Éliminer les perturbations dans les images. FP25: Faire une reconnaissance via intelligence artificielle. FP251: Renvoyer le numéro ID si cette image est trouvée dans la base des données. FP 252: Renvoyer "Unknown" si cette image n'est pas dans la base des données. FP26: Exécuter des opérations SQL pour lire le nom complet associé avec cette ID. FP27: Renvoyer le nom et le prénom de la personne.</p>	<ul style="list-style-type: none"> ● SQL ● SQL Query ● SQL Database ● AI ● Python
FP3	<p>FP31: Déterminer le navigateur utilisé par l'utilisateur. FP32: Se Connecter au port web 8080 pour accepter les demandes http, https. FP33: Convertir le nom du domaine à l'adresse IP pour établir une connexion avec le serveur. FP34: Envoyer une demande HTTP/HTTPS au serveur pour accéder au site web. FP35: Envoyer une réponse HTTP/HTTPS pour le client contient les informations de site web. FP36: Livrer la page web à la personne.</p>	<ul style="list-style-type: none"> ● Web Server ● HTTP/HTTPS port ● Navigateur ● HTML5
FS1	<p>FS11 : Acquérir les informations des citoyens. FS12 : Créer une base de données. FS13 : Intégrer la base de données dans le programme de l'IA. FS14 : Afficher le résultat de la détection.</p>	<ul style="list-style-type: none"> ● SQL ● SQL Server ● SQL Query
FS2	<p>FS21: Obtenir les informations nécessaires des citoyens (Nom, Prénom, Adresse, Date de naissance, email et numéro de carte national). FS22: Se connecter avec le serveur de la base des données. FS23: Créer une base de données. FS24: Ajouter un tableau dans cette base de données.</p>	<ul style="list-style-type: none"> ● SQL Server ● SQL Query ● SQL Connection ● SQL Database

	FS25: Exécuter des opérations SQL pour créer le tableau.	
FS3	<p>FS31: Inclure les bibliothèques nécessaires.</p> <p>FS32: Créer la classe principale.</p> <p>FS33: Déclarer les variables nécessaires.</p> <p>FS34: Définir les fonctions de l'algorithme.</p> <p>FS35: Créer la boucle principale.</p> <p>FS36: Charger les modules de l'algorithme à partir des bibliothèques.</p> <p>FS37: Créer CNN.</p> <p>FS38: Définir le modèle de prédiction.</p> <p>FS 39: Créer les conditions utilisées par le modèle de prédiction.</p> <p>FS310: Créer fonction de connexion avec la base des données.</p> <p>FS311: Définir les opérations SQL.</p>	<ul style="list-style-type: none"> ● System opérations ● Editor de code ● Les bibliothèques ● Python ● SQL ● SQL Server
FS4	<p>FS41: Faire une numérisation pour détecter le visage de cette personne.</p> <p>FS42: Capturer l'image.</p> <p>FS43: Supprimer les perturbations dans l'image.</p> <p>FS 44: Appliquer un filtre spécial à l'image.</p> <p>FS 45: Faire un recadrement à l'image.</p> <p>FS 46: Minimiser la taille d'image pour afficher le visage scellement.</p> <p>FS 47: Attribuer un numéro ID pour chaque image.</p> <p>FS48: Faire une connexion avec la base des données.</p> <p>FS49: Sauvegarde l'image finale dans la base des données.</p>	<ul style="list-style-type: none"> ● AI ● Base des données spéciale ● Python
FS5	<p>FS51: Reconnaître les personnes via la caméra.</p> <p>FS52: Détecter les visages des personnes.</p> <p>FS53: Analyser les images.</p> <p>FS54: Capturer les images.</p> <p>FS55: Traiter les images avec l'intelligence artificielle.</p> <p>FS56: Comparer les résultats avec la base de données des visages.</p> <p>FS 57: Renvoyer le résultat de la détection.</p>	<ul style="list-style-type: none"> ● Caméra ● Intelligence artificielle ● Système D'opération ● Base Des données
FS6	<p>FS 61: Lancer le navigateur web.</p> <p>FS 62: Établir une connexion avec le web serveur.</p> <p>FS 63: Connecter au port Web HTTP/HTTPS.</p> <p>FS 64: Envoyer une requête Web au serveur.</p> <p>FS 65: Demander la page Web.</p> <p>FS 66: Rendre la demande de page Web par navigateur.</p> <p>FS 67: afficher la page Web à la personne.</p>	<ul style="list-style-type: none"> ● Server ● Internet ● Web Navigateur
FS7	<p>FS71: Lancer les fonctions de base de système d'opération.</p> <p>FS72: Lire les informations nécessaires.</p> <p>FS 73: Passer les informations à l'algorithme.</p>	<ul style="list-style-type: none"> ● Système d'opération
FS8	<p>FS81: Ouvrir le site web.</p> <p>FS82: Établir une connexion avec la base des données SQL.</p> <p>FS83: Demander à l'utilisateur de saisir les informations nécessaires pour l'authentification.</p> <p>FS 84: lire les informations de l'utilisateur.</p> <p>FS85: Envoyer ces informations au serveur.</p>	<ul style="list-style-type: none"> ● Web Server ● SQL Server ● Web Protocole ● Navigateur ● API ● SQL Query

	FS86: Exécuter les opérations SQL pour valider les informations. FS 861: Envoyer une réponse « erreur » dans le cas échéant. FS 862: Envoyer une réponse de réussite dans le cas succès. FS 87: Traiter la réponse du serveur. FS 88: Authentifier l'utilisateur avec la plateforme. FS 89: Lire les informations des paiements. FS 810: Décrypter ces informations par le protocole. FS 811: Envoyer les informations pour la validation. FS 8111: Annuler l'opération dans le cas échéant FS 8112: Se connecter avec API de paiement dans le cas succès. FS 812: Retourner une réponse positive à l'utilisateur. FS 813: Afficher un message de succès. FS 814: Envoyer un email à l'utilisateur pour confirmation.	
FS9	FS91: Déterminer les normes à respecter. FS92: Appliquer les normes adéquates pour chaque étape du projet.	<ul style="list-style-type: none"> • Normes

Tableau 1: Diagramme de Fast

4 - Les Normes:

Lors du développement de ce projet, un ensemble des normes a été appliqué pour garantir une haute qualité professionnelle du projet.

1- Normes de programmation de AI :

1. Tous les variables suivent les déclarations de cas de chameau (CCD)
2. Application de la méthode "Set and forget"
3. DRY (Don't repeat yourself)
4. Pas de duplication

2- Normes d'écrire le code Python :

1. Principe de responsabilité unique (SRP)
2. Principe ouvert / fermé (OCP)
3. Principe de substitution de Liskov (LSP)
4. Principe d'inversion de dépendance (DIP)

3- Normes de programmation de site web :

1. Suivre le model ES6 pour JavaScript
2. Protection contre SQL Injection
3. Protection contre CSRF
4. Protection contre Phishing
5. Protection contre XSS
6. DRY (Don't repeat yourself)
7. Protection contre DDOs attacks

4- Normes de création des SQL databases :

1. SHA256 encryption
2. HTML tags/syntax escape
3. NNV (No null values)
4. No SQL raw queries
5. No weak authentication
6. Deni de service (DOs)
7. Secure system architecture

5- Normes de Backend (Server side) :

1. Protection contre les défauts d'injection (PI)
2. Protection contre Authentification cassée (BA)
3. Protection contre Exposition des données (DB)
4. Protection contre Bruteforce

5 - L'approche de "Face Mask Detection Algorithm":

Lors du développement de cet algorithme de détection de masque, certaines technologies ont été utilisées pour identifier les visages des personnes et analyser leur visage, puis décider si elles portent un masque ou non, l'une de ces technologies est YOLO (vous ne regardez qu'une seule fois) et est fondamentalement la détection d'objets en temps réel. Cet algorithme, qui est l'un des algorithmes de détection d'objets les plus efficaces qui englobe également plusieurs des idées les plus innovantes issues de la communauté de recherche en vision par ordinateur. La détection d'objets est une capacité critique de la technologie des véhicules autonomes. C'est un domaine de la vision par ordinateur qui explose et fonctionne tellement mieux qu'il y a à peine quelques années. À la fin de cet article, nous verrons quelques mises à jour récentes de YOLO par les chercheurs originaux.

Cet algorithme rend la détection de masque possible en implémentant un CNN spécial capable de traiter les données extrêmement rapidement en fonction de fonctions de coût qui tentent de minimiser l'erreur autant que possible et d'obtenir la meilleure prédiction possible.

6 - Diagramme d'étude des cas :



Figure 3: Diagramme étude de cas

7 - Diagramme de séquence :

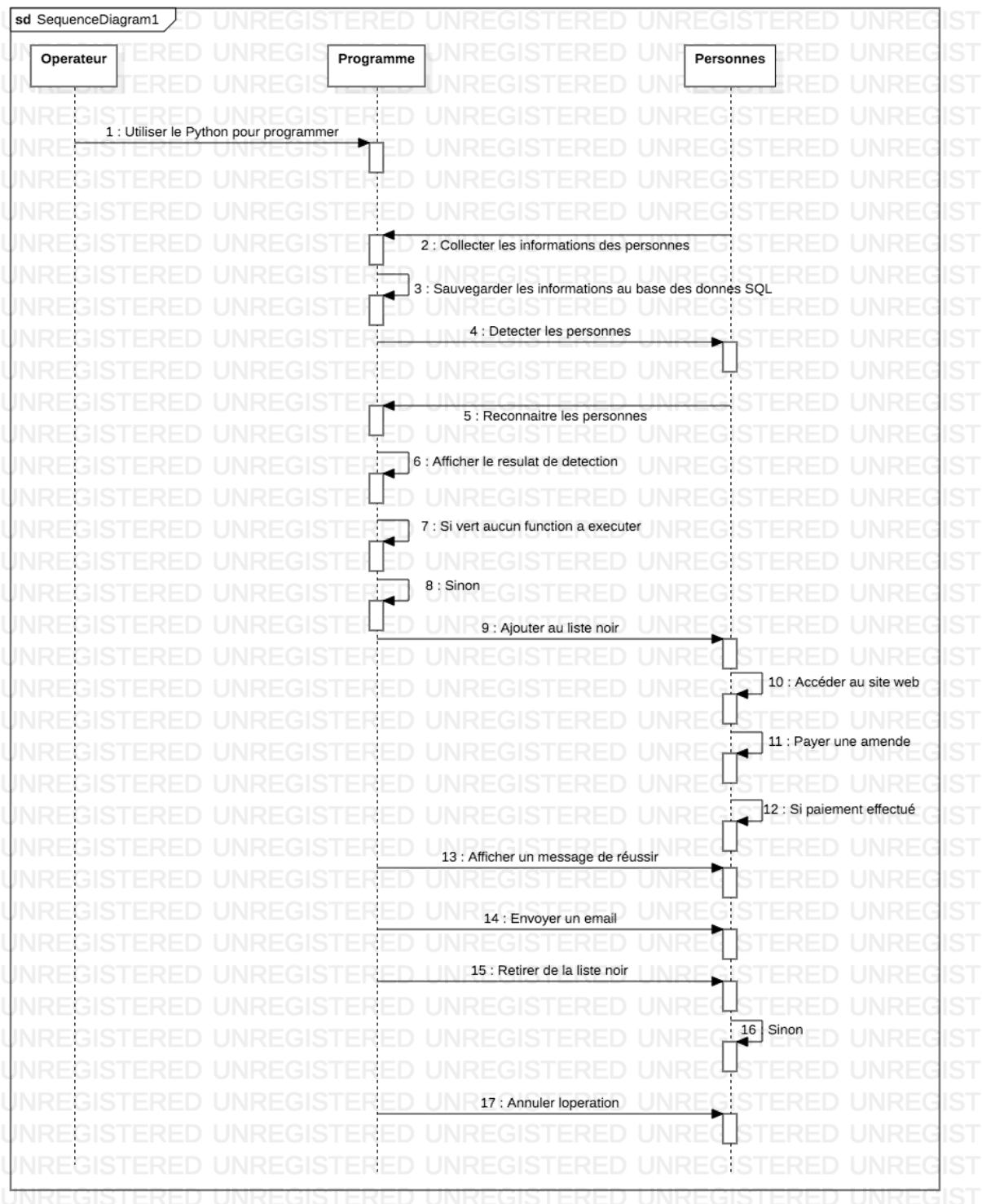


Figure 4 : Diagramme de séquence

CHAPITRE 2: Conception et réalisation

1 - Les réseaux neurones :

Les réseaux neurones sont un ensemble d'algorithmes, modélisés sur le cerveau humain et conçus pour reconnaître des modèles. Ils interprètent les données sensorielles à travers une sorte de machine, d'étiquetage ou de regroupement des entrées brutes. Les modèles qu'ils reconnaissent sont numériques, contenus dans des vecteurs, dans lesquels toutes les données du monde réel, qu'il s'agisse d'images, de sons, de textes ou de séries temporelles, doivent être traduites.

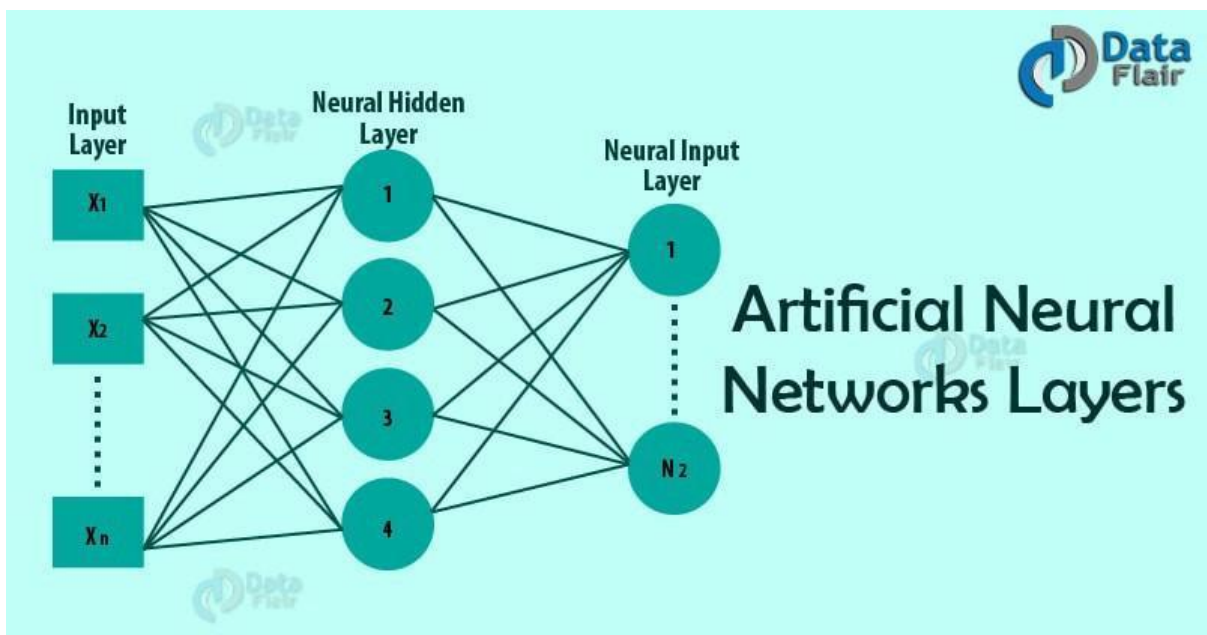


Figure 5 - Representation des Réseaux neurones

2 - Algorithm CNN:

CNN ou Conventional Neural Network est une classe de réseaux d'apprentissage en profondeur principalement utilisés pour la reconnaissance et le traitement d'images.

Il s'agit également d'un puissant outil de traitement d'image, qui utilise l'apprentissage en profondeur pour effectuer certaines tâches, souvent en utilisant la vision par ordinateur qui comprend la reconnaissance d'images et de vidéos.

Les réseaux neurones conventionnels suivent la structure du cerveau en termes de fonctionnalité, il est composé de 3 parties : couche d'entrée, couches cachées et couche de

sortie. Ces couches sont composées de neurones reliés entre eux par des liens spéciaux, ces liens sont capables de recevoir et d'envoyer des informations entre neurones.

Les réseaux neurones sont composés par la première couche qui est la couche d'entrée qui est responsable de la réception des données d'entrées sous forme de tableau, cette entrée pourrait être une image ou une vidéo qui doit être traitée via CNN pour certains processus, cette image / vidéo est décomposées en plusieurs sous images qui constituent un tableau d'image après son traitement avec des fonctions externes.

Après cela, la couche d'entrée analysera et traitera les données, puis elles seront transmises à la première couche cachée qui compose le réseau neuronal.

La couche cachée dans le réseau de neurones convolutifs est située entre l'entrée et la sortie de l'algorithme, alors que les couches cachées effectuent des transformations non linéaires des entrées dans le réseau ou en d'autres termes la couche cachée applique des pondérations aux entrées et les dirige via une fonction d'activation comme sortie.

Les couches cachées sont des couches de fonctions mathématiques conçues chacune pour produire une sortie spécifique à un résultat attendu. Par exemple, certaines formes de couches cachées sont appelées fonctions d'écrasement. Ces fonctions sont particulièrement utiles lorsque la sortie prévue de l'algorithme est une probabilité car elles prennent une entrée et produisent une valeur de sortie comprise entre 0 et 1, la plage pour définir la probabilité.

Les fonctions d'activation sont une autre partie principale de la couche cachée, elles sont utilisées pour déterminer la sortie du réseau neuronal comme oui ou non. Il attribut les valeurs résultant entre 0 à 1 ou -1 à 1 etc. (selon la fonction).

Les fonctions d'activation peuvent être fondamentalement divisées en 2 types :

- Fonction d'activation linéaire
- Fonctions d'activation non linéaires

2 - 1 - Fonction d'activation linéaire :

La fonction est une ligne ou linéaire. Par conséquent, la sortie des fonctions ne sera confinée entre aucune plage.

Équation: $f(x) = x$

Plage: (-infini à l'infini)

Le degré de la complexité des divers paramètres des données habituelles qui sont transmises aux réseaux de neurones rend le système fortement non linéaire.

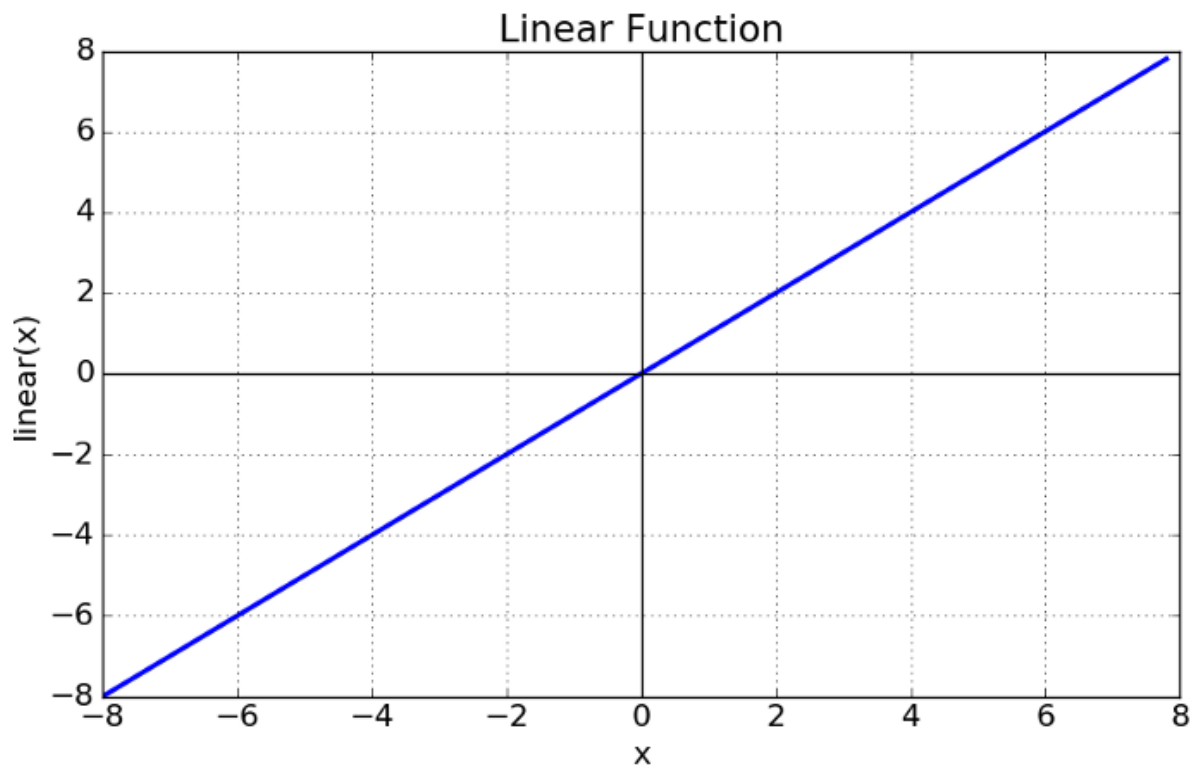


Figure 6 - Fonction d'activation Linéaire

2 - 1 -2 Fonction d'activation non linéaire :

Les fonctions d'activation non linéaires sont les fonctions d'activation les plus utilisées. La non-linéarité aide à donner au graphique un aspect similaire :

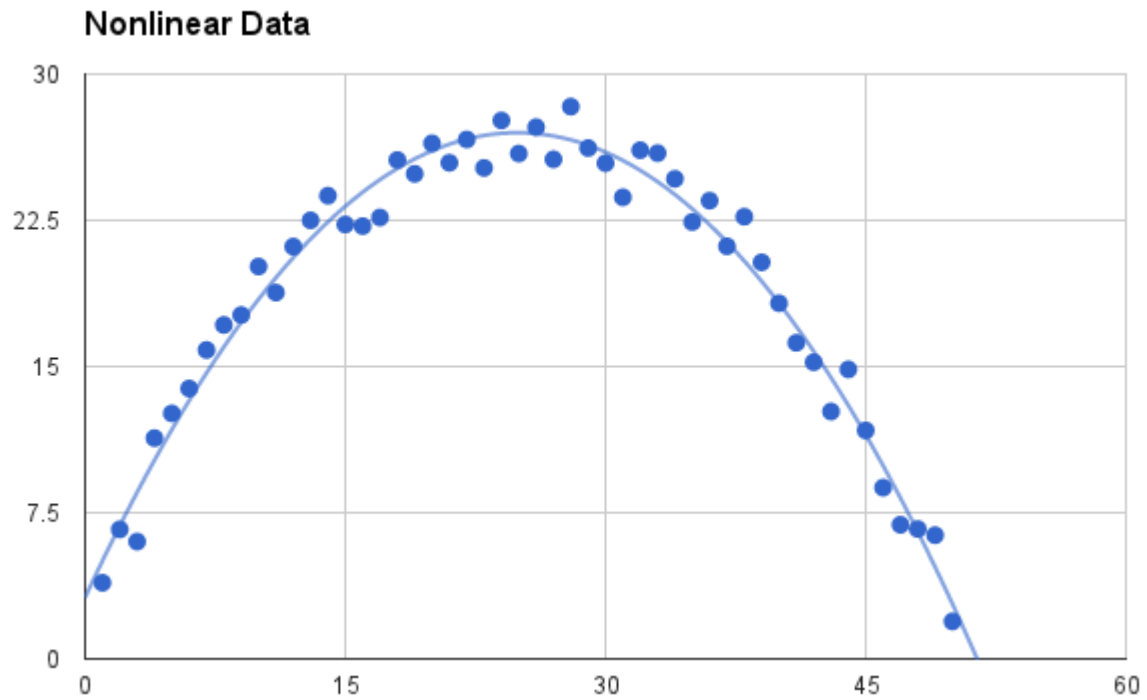


Figure 7 - Fonction d'activation non-linéaire

Cela permet au modèle de se généraliser ou de s'adapter facilement à une variété de données et de différencier les résultats.

Les principales terminologies nécessaires pour comprendre les fonctions non linéaires sont:

Dérivée ou différentielle : changement de l'axe des y par rapport à t. changement de l'axe des x. Il est également connu sous le nom de pente.

Fonction monotone : une fonction qui est entièrement non croissante ou non décroissante.

Les fonctions d'activation non linéaires sont principalement divisées en fonction de leur plage ou de leurs courbes :

1. Sigmoid or Logistic Activation Function:

The Sigmoid Function curve looks like a S-shape

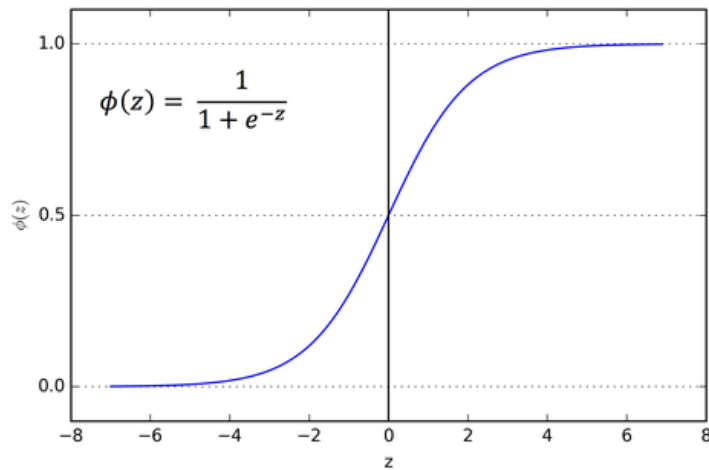


Figure 7.1 Fonction d'activation sigmoïde

La principale raison pour laquelle nous utilisons la fonction sigmoïde est qu'elle existe entre (0 et 1). Par conséquent, il est particulièrement utilisé pour les modèles où nous devons prédire la probabilité en tant que sortie, car la probabilité de quoi que ce soit n'existe qu'entre 0 et 1, le sigmoïde est le bon choix.

2. Tanh or hyperbolic tangent Activation Function:

\tanh est aussi comme le sigmoïde logistique mais mieux. La plage de la fonction \tanh va de (-1 à 1). \tanh est également sigmoïde (en forme de S).

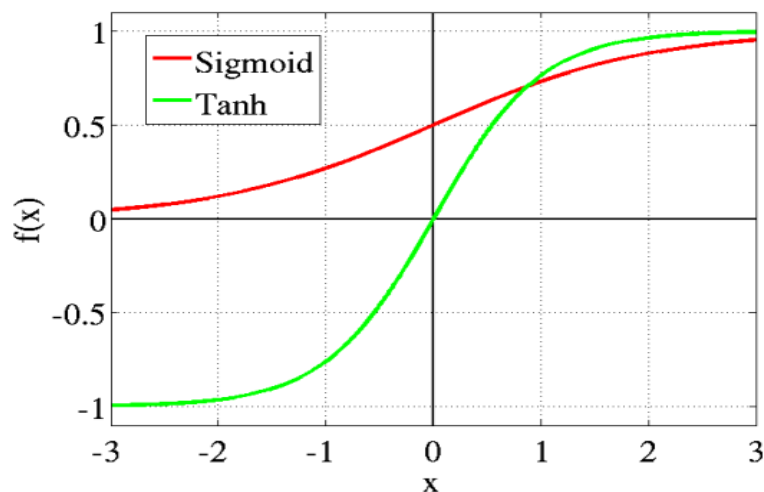


Figure 7.2 Fonction d'activation Tanh

L'avantage est que les entrées négatives seront mappées fortement négatives et les entrées nulles seront mappées près de zéro dans le graphique \tanh .

3. Fonction d'activation ReLU (Rectified Linear Unit):

Le ReLU est la fonction d'activation la plus utilisée au monde à l'heure actuelle, depuis, elle est utilisée dans presque tous les réseaux de neurones convolutifs ou le deep learning.

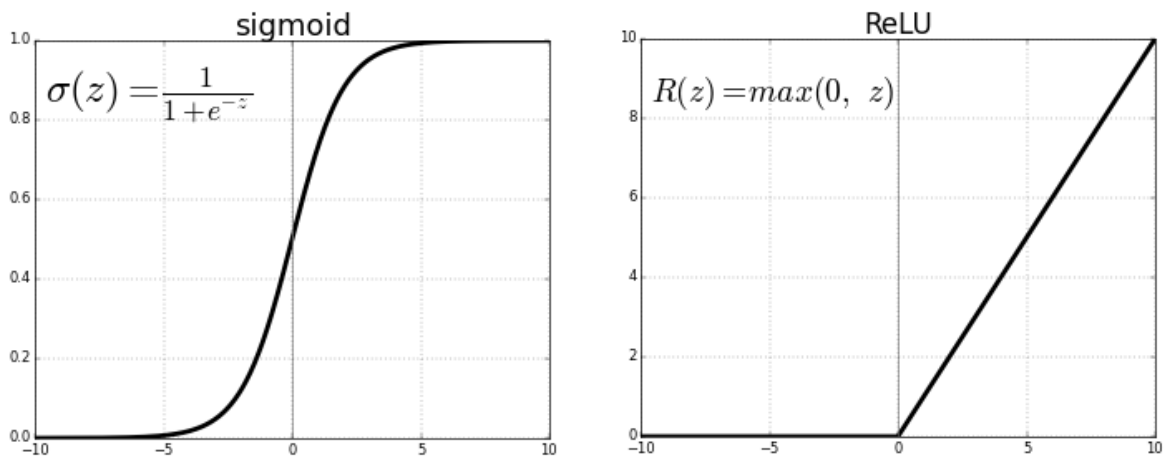


Figure 7-3 Fonction d'activation sigmoid et ReLU

Comme vous pouvez le voir, le ReLU est à moitié rectifié (par le bas). $f(z)$ est égal à zéro lorsque z est inférieur à zéro et $f(z)$ est égal à z lorsque z est supérieur ou égal à zéro. Plage: [0 à l'infini]

Enfin, la couche de sortie est chargée de produire une sortie ou plutôt une prédiction à partir du CNN.

La couche de sortie prend les entrées qui sont transmises par les couches avant elle, effectue les calculs via ses neurones, puis la sortie est calculée.

Dans ce projet, la fonction d'activation ReLU peut être sélectionnée.

Pour mettre en œuvre ce CNN dans le projet, les étapes suivantes ont été suivies :

```

def maskSetup([self]):
    if self.setup:
        checkpoint = torch.load(self.modelPath, map_location='cpu')
        model = checkpoint['model']
        model.load_state_dict(checkpoint['state_dict'])
        for parameter in model.parameters():
            parameter.requires_grad = False

        self.model = model.eval()

        self.faceDetector = MTCNN()

        self.transforms = transforms.Compose([
            transforms.Resize((224,224)),
            transforms.ToTensor(),
            transforms.Normalize((0.5,0.5,0.5), (0.5,0.5,0.5))
        ])

    self.setup = False

```

Figure 8 - Fonction de configuration pour le CNN

La première étape est effectuée par cette fonction, d'abord cette fonction charge le fichier de prédiction qui est généré à partir de millions de données de test. Ensuite, certains modèles sont déclarés qui seront utilisés plus tard pour construire le CNN comme le mode FaceDetector et les variables de transformation qui seront utilisées pour capturer, traiter et analyser les données.

La variable "setup" est chargée de faire exécuter la fonction une seule fois lorsque la classe est utilisée pour éviter les conflits.

```

while(True):
    ret, frame = cap.read()
    if not ret:
        break

    currentImg = frame.copy()
    currentImg = resize(currentImg, width=980)
    imageShape = currentImg.shape
    fno += 1

    if(fno%1 == 0 or fno == 1):
        detectorObj.maskSetup()
        detectorObj.maskProcess(currentImg, minH, minW)
        outputFrame = detectorObj.maskDisplay()

        if writer is None:
            print("Writing the output file to: ", opname)
            fourcc = cv2.VideoWriter_fourcc(*"MJPG")
            writer = cv2.VideoWriter(opname, fourcc, 30,
                (outputFrame.shape[1], outputFrame.shape[0]), True)

        cv2.imshow('Mask Detection Dashboard', outputFrame)

    if videoMode:
        writer.write(outputFrame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()

```

Figure 9 - Boucle "While" principale

Après la configuration ci-dessus, nous pouvons maintenant commencer à créer le CNN. La première chose est la couche d'entrée du CNN qui est chargée de lire les données de l'utilisateur et de les transmettre aux couches cachées pour traitement.

Pour que cela soit implémenté, nous devons ouvrir une fenêtre et démarrer la caméra pour capturer les données, pour cela, nous avons défini une boucle while infinie qui bouclera sur chaque image jusqu'à ce que l'utilisateur appuie sur la lettre "q" dans le clavier, ce qui provoquera l'arrêt de la boucle while et la sortie du programme.

Pour démarrer la caméra, nous utilisons cette commande "ret, frame = cap.read ()" qui amènera toute caméra Web correctement installée à démarrer le streaming vidéo, après quoi la boucle capturera chaque image et l'enverra aux couches cachées de le CNN. La fenêtre ouverte aura des dimensions carrées 980 x 980 px.


```

def maskProcess(self, frame, minH, minW):
    image = frame.copy()
    displayImg = frame.copy()
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = self.faceDetector.detect_faces(image)
    faces = []
    font_scale=0.5
    thickness = 2
    font=cv2.FONT_HERSHEY_SIMPLEX

    for i in results:
        faces.append(i['box'])

    for (x, y, w, h) in faces:

        x, y = max(0,x), max(0,y)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255,0,0), 2)

        cropped_img = frame[y:y+h, x:x+w, :]
        temp_image = Image.fromarray(cropped_img, mode = "RGB")
        temp_image = self.transforms(temp_image)
        image = temp_image.unsqueeze(0)

        result = self.model(image)
        _, maximum = torch.max(result.data, 1)
        prediction = maximum.item()

```

Figure 10 - Fonction du processus

À ce stade, la couche d'entrée du CNN est terminée, nous pouvons maintenant commencer à créer la couche cachée qui est responsable du traitement des entrées de la couche d'entrée. La fonction "mask Process" qui est responsable du traitement reçoit la trame de la couche d'entrée et la hauteur minimale, largeur de la trame. Ensuite, ce cadre sera analysé pour détecter s'il contient un visage, si tel est le cas, le visage sera coupé du cadre et ajouté au tableau en tant que "boîte", plus tard ce tableau de boîtes sera analysé pour voir s'il s'agit d'un visage avec un masque ou non. Une fois l'analyse terminée, nous serons prêts à transmettre les données à la couche de sortie, puis à l'utilisateur en conséquence.

```

if prediction == 0:
    cv2.putText(displayImg, "Masked", (x,y - 10), font, font_scale, (0,255,0), thickness)
    cv2.rectangle(displayImg, (x, y), (x+w, y+h), (0,255,0), 2)
elif prediction == 1:
    if len(faces) >= 2:
        for f in faces:
            label = self.recognizer(frame, minH, minW, faces)
            cv2.putText(displayImg, label, (x,y - 10), font, font_scale, (0,0,255), thickness)
            cv2.rectangle(displayImg, (x, y), (x+w, y+h), (0,0,255), 2)
    else:
        label = self.recognizer(frame, minH, minW, faces)
        cv2.putText(displayImg, label, (x,y - 10), font, font_scale, (0,0,255), thickness)
        cv2.rectangle(displayImg, (x, y), (x+w, y+h), (0,0,255), 2)

self.outputFrame = displayImg.copy()

```

Figure 11 - Conditions des décision

La fonction précédente générera une valeur comprise entre 0 et 1. Si cette variable est égale à 0, le visage porte un masque, la fonction de traitement transmettra le texte "Masque" qui sera affiché avec le visage.

Sinon si la valeur est égale à 1, la trame sera passée à une autre fonction responsable de la reconnaissance faciale.

```
def recognizer(self, img, minH, minW, fc):
    id = 2
    names = ['', 'Riade', 'elon']

    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer.yml') #load trained model
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath);

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
    )

    if len(faces) == 1:
        for(x,y,w,h) in faces:
            id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

            # Check if confidence is less them 100 ==> "0" is perfect match
            if (confidence < 50):
                connection = pymysql.connect(host="localhost", user="root",passwd="",database="Citizens" )
                cursor = connection.cursor()
```

Figure 12 - Fonction de reconnaissance du visage

```

cursor.execute("SELECT ID FROM NO_MASK_CITIZENS WHERE ID = {};".format(id) )
data = cursor.fetchall()
if len(data) == 0:
    Q = "INSERT INTO NO_MASK_CITIZENS(ID) VALUES({});".format(id)
    cursor.execute(Q)

cursor.execute("Select NAME from main_1_citizens_data WHERE NATID={};".format(id))
NAME = cursor.fetchall()

cursor.execute("Select ADDRESS from main_1_citizens_data WHERE NATID={};".format(id))
ADD = cursor.fetchall()

# info = N + F
connection.commit()
connection.close()
N = ''.join(str(NAME))
return(N[N.find('(')+2 : N.find(')')])
else:
    id = "unknown,"
    return(str(id))

elif len(fc) >= 2:

    Fname = []

    for(x,y,w,h) in faces:
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 50):
            connection = pymysql.connect(host="localhost", user="root",passwd="",database="Citizens" )
            cursor = connection.cursor()

```

Figure 13-1 Décision à visage unique

```

cursor.execute("SELECT ID FROM NO_MASK_CITIZENS WHERE ID = {};".format(id) )
data = cursor.fetchall()
if len(data) == 0:
    Q = "INSERT INTO NO_MASK_CITIZENS(ID) VALUES({});".format(id)
    cursor.execute(Q)

cursor.execute("Select NAME from main_1_citizens_data WHERE NATID={};".format(id))
NAME = cursor.fetchall()

# cursor.execute("Select FAMILY_NAME from main_1_citizens_data WHERE NATID={};".format(id))
# FNAME = cursor.fetchall()

cursor.execute("Select ADDRESS from main_1_citizens_data WHERE NATID={};".format(id))
ADD = cursor.fetchall()

# info = NAME + FNAME
connection.commit()
connection.close()
N = ''.join(str(NAME))
Fname.append(N[N.find("(")+2 : N.find(")")])
else:
    id = "unknown"
    Fname.append(id)

import re
str_fname = " ".join(re.findall("[a-zA-Z]+", str(Fname)))

return str(str_fname)

```

Figure 13-2 Décisions multi-visages

Cette fonction est responsable de la reconnaissance faciale pour les visages non masqués, d'abord cette fonction recevra le cadre, la hauteur et la largeur min du cadre et la liste des visages. Ceci est également considéré comme CNN car la fonction précédente "Processus de masque" agit comme une couche d'entrée et transmet les données à cette fonction et agit comme couche cachée de la reconnaissance de visage CNN, la fonction reçoit le tableau des images et analysera chaque image du tableau. Ensuite, si la longueur du tableau des images est égale à 1, l'algorithme exécutera un test de confiance pour voir si le visage est stocké dans la base de données, si c'est le cas, l'algorithme exécutera des requêtes SQL pour obtenir le nom du visage et renvoyer le résultat au " mask process "sinon il retournera" Unknown ". Ce processus est répété si le tableau est plus grand que 1 en longueur et il crée un nouveau tableau des images pour stocker toutes les données de cette table et ensuite renvoyer le résultat à la fonction principale.

À ce stade, le processus est terminé et l'algorithme, bien sûr, si le visage n'a pas de masque et que la prédiction est de 1, il capturera l'identifiant associé à ce visage et l'enregistrera dans la base de données avant de récupérer le nom du visage pour une utilisation ultérieure.

3 - YOLO Algorithm (You Only Look Once):

YOLO ou (vous ne regardez qu'une seule fois) est l'un des algorithmes les plus utilisés en matière de reconnaissance visuelle. Cet algorithme est le meilleur choix lorsqu'il s'agit de la reconnaissance visuelle en raison de sa capacité à détecter des objets en temps réel et peut assurer une prédiction de haute qualité, ce qui est difficile pour les algorithmes de reconnaissance visuelle standard.

L'algorithme YOLO est un algorithme basé sur la régression, au lieu de sélectionner la partie intéressante d'une image, il prédit des classes et des boîtes englobantes pour l'image entière en une seule exécution de l'algorithme.

YOLO ne recherche pas les régions intéressées dans l'image d'entrée qui pourraient contenir un objet, mais divise l'image en cellules, généralement une grille 19x19. Chaque cellule est alors responsable de la prédiction de "K" boîtes englobantes.

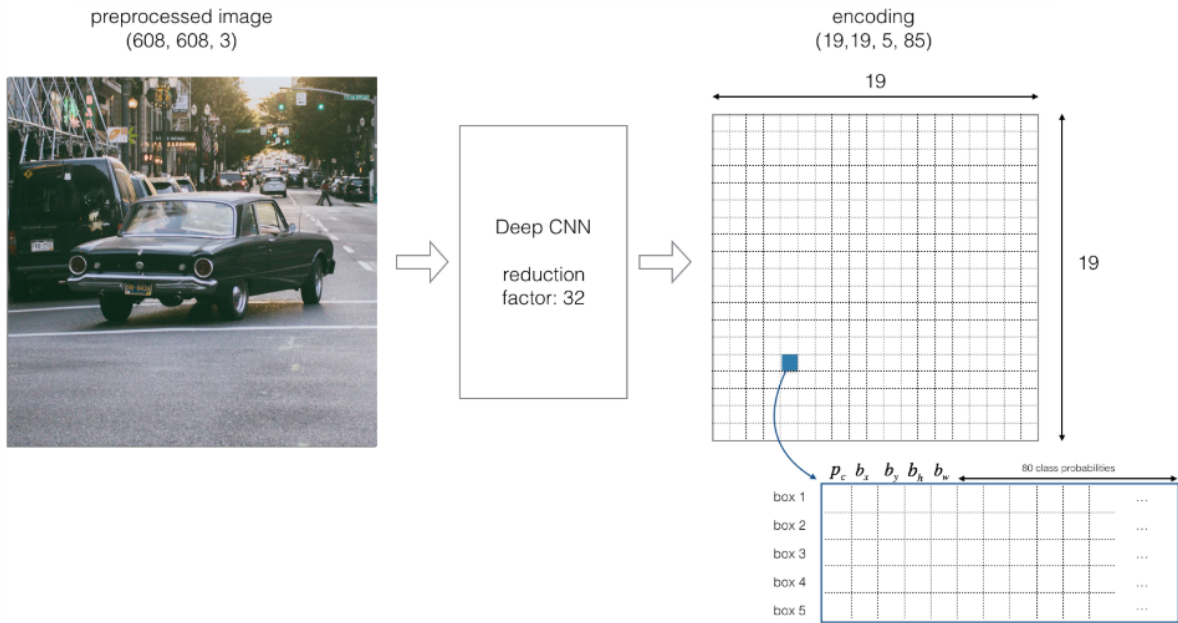


Figure 14 - Processus d'encodage de YOLO

Un objet est considéré comme se trouvant dans une cellule spécifique de la décomposition précédente uniquement si les coordonnées du centre de la boîte d'ancrage se trouvent dans cette cellule.

Pendant la propagation rapide, YOLO détermine la probabilité que la cellule contienne une certaine classe.

La classe avec la probabilité maximale est choisie et affectée à cette cellule de grille particulière. Un processus similaire se produit pour toutes les cellules de la grille présentes dans l'image.

Une fois ce processus terminé, nous devrions obtenir quelque chose comme ceci:

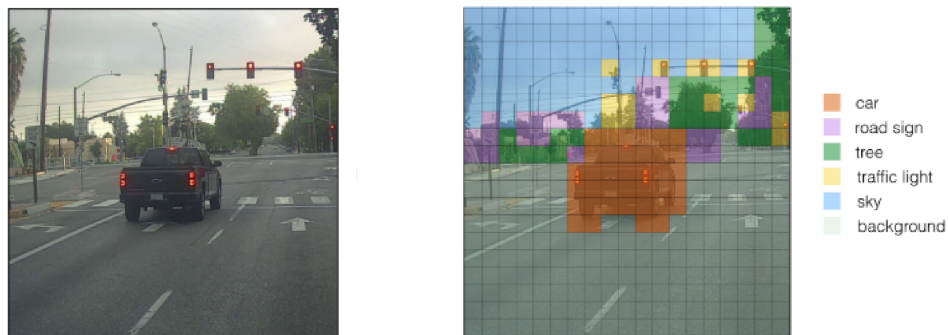


Figure 15 - Phase de segmentation de YOLO

L'image ci-dessus montre l'avant et l'après pour chaque cellule de la grille de l'image après avoir prédit les probabilités de classe, l'étape suivante est la suppression Non-max, cela aide l'algorithme à se débarrasser des boîtes d'ancrage inutiles, comme vous pouvez le voir dans la figure ci-dessous, il existe de nombreuses boîtes d'ancrage calculées en fonction des probabilités de classe.



Figure 16 - YOLO avant et après la suppression non-maximale

La suppression non-max élimine les boîtes englobantes qui sont très proches en effectuant l'IoU (Intersection over Union). En gardant les boîtes englobantes ayant la probabilité de classe la plus élevée, il rejette alors les boîtes englobantes dont la valeur de «IoU» est supérieure à un seuil. Cela signifie que ces deux boîtes englobantes couvrent le même objet mais que l'autre a une faible probabilité, donc il est éliminé

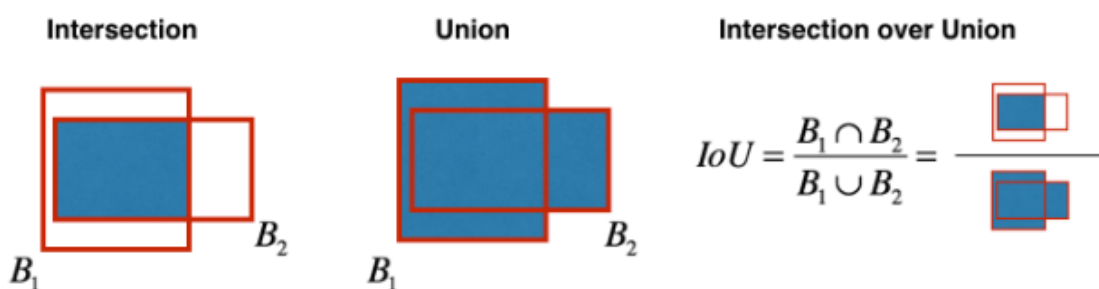


Figure 17 - Phase de l'intersection YOLO

A la fin, l'algorithme est prêt à sortir la prédiction, l'architecture de l'algorithme peut être représentée dans la figure ci-dessous :

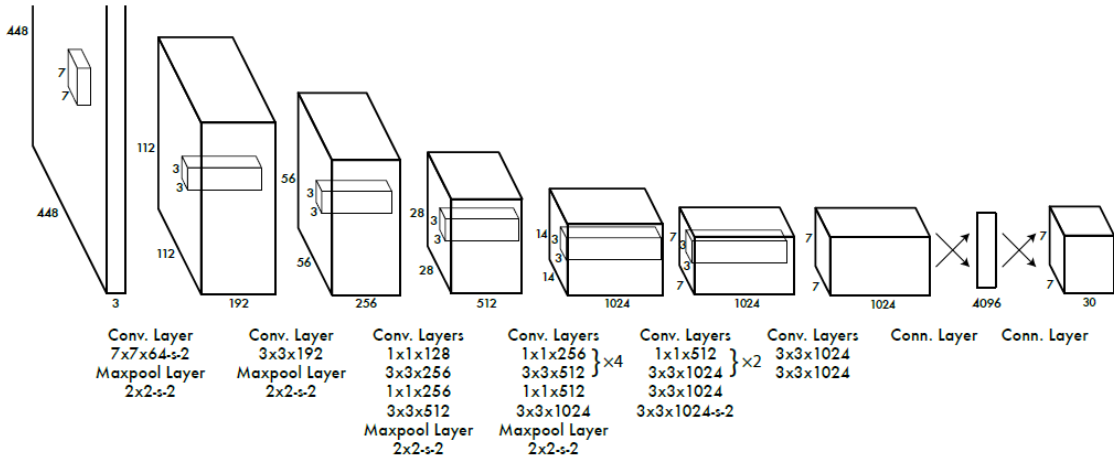


Figure 18 - Architecture interne de YOLO

Il y a aussi la fonction de coût (une fonction qui minimiser l'erreur) qui est le paramètre le plus important et peut être représentée avec l'équation mathématique suivante :

$$\begin{aligned}
 \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} & \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} & \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} & (C_i - \hat{C}_i)^2 \\
 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} & (C_i - \hat{C}_i)^2 \\
 + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} & (p_i(c) - \hat{p}_i(c))^2
 \end{aligned}$$

Représentation mathématique de la fonction de coût utilisée dans YOLO

4 - Les Techniques de site web:

Lors du développement des sites Web, diverses techniques ont été mises en œuvre en utilisant des langages de programmation de développement Web standard tels que javascript et HTML5 / CSS3.

HTML5 signifie «langage de balisage hypertexte» qui est au cœur de tout site Web, html n'est pas un langage de programmation mais plutôt un langage basé sur des balises qui se concentre sur la création de composants Web à afficher pour les utilisateurs via le navigateur Web.

C'est aussi un système qui permet de modifier l'apparence des pages Web, ainsi que de faire des ajustements à leur apparence. Il permet également de structurer et de présenter du contenu pour le Web.

En utilisant la fonction html seul, les pages Web ne peuvent pas avoir de style pour ses composants tels que les couleurs, les polices et d'autres propriétés. Par conséquent, CSS3 est utilisé pour styliser les composants HTML et ajouter des propriétés différentes à chaque balise html individuelle pour fournir des éléments faciles à comprendre et une meilleure expérience UX pour les utilisateurs plutôt qu'une simple page Web.

CSS3 est responsable de créer les classes de nom et id qui sont déclarés à l'intérieur de la balise html, de cette façon, le css3 peut identifier certaines balises html et appliquer des modifications sans affecter les autres composants.

Ce qu'on vient d'évoquer représente le « frontend » ; dans ce qui il s'en suit nous nous intéresserons au côté « backend » ou plutôt le côté serveur.

Pour gérer cette tâche, un framework backend a été implémenté et ce framework est de type « django ».

Django est un framework backend qui utilise python comme langage de programmation et il suit le modèle MVC (model view controller), Django est très flexible et il offre beaucoup de fonctionnalités prédéfinies pour gérer différentes tâches. De cette façon, les développeurs n'auront pas à tout créer à partir de zéro mais plutôt à se concentrer sur le développement du site Web, il offre également de nombreuses bibliothèques qui peuvent être utilisées facilement pour fournir certaines fonctionnalités qui ne peuvent pas être faites avec des langages frontaux tels que les e-mails, les notifications push, les interactions de base de données et d'autres fonctionnalités.

Ensuite, concernant la base de données, il existe de nombreuses façons différentes pour les utiliser dans le site Web, pour ce projet, l'interface Phpmysql a été utilisée pour être le serveur SQL et l'interface principale des interactions dans les tables et la base de données.

Le site Web a été créé en suivant ces étapes :

Tout d'abord, Django a été installé afin de gérer toutes les opérations côté serveur telles que les pages Web, les interactions avec la base de données et de gérer les demandes Web à l'intérieur du site Web.


```
core > core > urls.py
1  """core URL Configuration
2
3  The `urlpatterns` list routes URLs to views. For more information please see:
4      https://docs.djangoproject.com/en/3.0/topics/http/urls/
5  Examples:
6  Function views
7      1. Add an import:  from my_app import views
8      2. Add a URL to urlpatterns:  path('', views.home, name='home')
9  Class-based views
10     1. Add an import:  from other_app.views import Home
11     2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
12  Including another URLconf
13     1. Import the include() function: from django.urls import include, path
14     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
15  """
16  from django.contrib import admin
17  from django.urls import path, include
18  from main_1.views import index, pay
19  urlpatterns = [
20      path('', index),
21      path('submit/', pay),
22      path('admin/', admin.site.urls),
23  ]
24
```

Figure 19 - Principaux liens du site web

Donc, la première chose après l'installation de django est de créer les routes du site Web, cette tâche est responsable de la création des URL pour le site Web et chaque URL doit afficher une page Web différente. Si l'utilisateur saisit l'url "localhost: 8080" dans le navigateur web, django attribuera la fonction liée au chemin "/" ou plutôt le chemin vide "" et appellera l'index de la fonction, cela provoquera le retour de la fonction index page Web spécifique liée à ce chemin.

De même la fonction d'index a été déclarée dans le fichier de vue du projet:

```
core > main_1 > views.py
1  from django.shortcuts import render
2  from .models import Citizens_data
3  from django.core.mail import send_mail
4  def index(request):
5      if request.method == "POST":
6          ID = request.POST.get("ID")
7          PASS = request.POST.get("PASS")
8          if Citizens_data.objects.filter(NATID = ID).exists():
9              data = Citizens_data.objects.get(NATID = ID)
10             context = {
11                 'data': data
12             }
13
14             return render(request, 'main/dashboard.html', context)
15         else:
16             print("NOT FOUND")
17         return render(request, 'main/main.html')
18
19  from django.db import connection
20  from django.conf import settings
21  from django.core.mail import EmailMultiAlternatives
22  from django.template.loader import get_template
23  from django.template import Context
24
25  def pay(request):
26      if request.method == "POST":
27          ID = request.POST.get("ID")
28          cursor = connection.cursor()
```

Figure 20 - Fonction de l'index principal du site web

Ici, la fonction d'index renverra le modèle html "principal" dans tous les cas, le fichier html "principal" est notre page de connexion et si l'utilisateur saisit ses données et les soumet au serveur, la fonction recevra la requête "POST" et puis compare les données reçues avec celles de la base, si les données envoyées par l'utilisateur sont valides, le site Web authentifie l'utilisateur et renverra différentes pages Web liées à cet utilisateur spécifiquement, sinon il les redirigera via la page de connexion à nouveau.

```

<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<div
  class="min-h-screen flex items-center justify-center bg-gray-50 py-12 px-4 sm:px-6 lg:px-8"
>
  <div class="max-w-md w-full space-y-8">
    <div>
      
      <h2 class="my-6 text-center text-3xl font-extrabold text-gray-900">
        Sign in to your account
      </h2>
      <div
        class="flex items-center bg-blue-500 text-white text-sm font-bold px-4 py-3"
        role="alert"
      >
        <svg
          class="fill-current w-4 h-4 mr-2"
          xmlns="http://www.w3.org/2000/svg"
          viewBox="0 0 20 20"
        >
          <path
            d="M12.432 0c1.34 0 2.01 1.12 2.01 1.957 0 1.305-1.164 2.512-2.679 2.512-1.269 0-2.009-.75-1.974-1.99C9.789 1.436 10.67 0"
          />
        </svg>
      <p>

```

```

        In order to login you need to provide your ID number and your
        birthday date as password, no spaces or special characters are
        allowed.
      </p>
    </div>
  </div>
  <div>
    <form class="mt-8 space-y-6" action="" method="POST">
      {% csrf_token %}
      <input type="hidden" name="remember" value="true" />
      <div class="rounded-md shadow-sm -space-y-px">
        <div>
          <label for="email-address" class="sr-only">
            National ID Number</label>
          <input
            id="email-address"
            name="ID"
            required
            class="appearance-none rounded-none relative block w-full px-3 py-2 border border-gray-300 placeholder-gray-500 text-gray-900"
            placeholder="National ID Number"
          />
        </div>
        <div>
          <label for="password" class="sr-only">Birthday Date</label>
          <input
            id="password"
            name="PASS"
            type="password"
            autocomplete="current-password"
            required
            class="appearance-none rounded-none relative block w-full px-3 py-2 border border-gray-300 placeholder-gray-500 text-gray-900"
            placeholder="Birthday Date"
          />
        </div>
      </div>
    </form>
  </div>

```

```

</div>
</div>
<div>
  <button
    type="submit"
    class="group relative w-full flex justify-center py-2 px-4 border border-transparent text-sm font-medium rounded-md text-white
  >
    <span class="absolute left-0 inset-y-0 flex items-center pl-3">
      <!-- Heroicon name: lock-closed -->
      <svg
        class="h-5 w-5 text-indigo-500 group-hover:text-indigo-400"
        xmlns="http://www.w3.org/2000/svg"
        viewBox="0 0 20 20"
        fill="currentColor"
        aria-hidden="true"
      >
        <path
          fill-rule="evenodd"
          d="M5 9V7a5 5 0 10 0V2a2 2 0 01-2 2H5a2 2 0 01-2-2v-5a2 2 0 012-2zm8-2v2H7a3 3 0 016 0z"
          clip-rule="evenodd"
        />
      </svg>
    </span>
    Sign in
  </button>
</div>
</form>
</div>
</div>
</body>
</html>

```

Figure 21 - Code HTML de la page de connexion du site web

Le code html est simple : un en-tête, un corps et un style CSS. Ici, le framework CSS utilisé est tailwind css pour un style de pages Web rapide et meilleur, les styles dépendent des classes utilisées, par exemple si la balise html a reçu la classe "shadow", cette balise aura une ombre sans qu'il soit nécessaire pour créer le css brut car il a été précédemment déclaré par tailwind css.

Ce modèle html contient des formulaires qui nécessitent 2 entrées afin d'authentifier l'utilisateur, un numéro d'identification national et une date de naissance comme mot de passe pour assurer la sécurité et l'efficacité.

Lorsque l'utilisateur appuiera sur le bouton de connexion, ce formulaire enverra une requête POST au serveur contenant ces informations et plus particulièrement à la fonction d'index dont nous avons parlé précédemment.

Une fois l'utilisateur authentifié, Django rediriger les utilisateurs vers ce modèle HTML dynamique avec ces informations utilisateur.

```

<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<nav>
</nav>
<section
class="h-screen w-screen bg-gray-100 flex flex-col items-start justify-center p-10"
>
<h1 class="text-2xl mb-5 font-medium">Your Pending Payments:</h1>
<div class="w-full bg-white shadow-md rounded-md self-center p-10">
<h1 class="font-bold tracking-tight text-xl">{{data.NAME}}</h1>
<p class="text-gray-500 mt-3">
You have a ticket with 10000.00 DA for not wearing your mask and not
respecting the law, therefore you must pay this amount here with your
credit card or at the nearest police station, thank you for your
concerns.
</p>
<p class="font-bold text-lg mt-5">
AMOUNT:
<span class="text-lg font-bold text-blue-500 mt-4 font-mono"
>10000.00 DA</span>
</p>
<p class="font-bold text-lg mt-2">
FULL NAME:
<span class="text-lg font-bold text-blue-500 mt-4"
>{{data.NAME}}</span>
</p>
<p class="font-bold text-lg mt-2">
ADRESS:
<span class="text-lg font-bold text-blue-500 mt-4"
>{{data.ADDRESS}}</span>
</p>
<p class="font-bold text-sm uppercase underline mt-10">
Enter your credit card informations
</p>
<form action="submit/" method="POST" class="mt-1">
{csrf_token%}
<input type="text" name="ID" class="hidden" value="{{data.NATID}}" />
<input
type="number"
required
class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
placeholder="Credit Card Number"
maxlength="16"
/>
<div class="flex justify-center mt-3">
<input
type="text"
required
class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
placeholder="End Date"
/>
<input
type="text"
required
class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
placeholder="CCV"
/>
</div>
<button
type="submit"
class="w-full p-2 bg-blue-500 rounded my-3 text-white"

```

Figure 22 - Code HTML pour la page principale

Comme les modèles HTML précédents, les mêmes techniques ont été copiés et réutilisés dans ce modèle html, la différence est que ce modèle est dynamique et qu'il affichera le nom de l'utilisateur authentifié, par exemple "{{data.NAME}}" renverra le nom de cet utilisateur

authentifié et "{{data.ADDRESS}}" renverra l'adresse personnelle de cet utilisateur en fonction des enregistrements de la base de données.

Ce modèle contient un autre "formulaire" qui est responsable de l'envoi des demandes de paiement au serveur après que l'utilisateur a fourni les informations nécessaires, ce "formulaire" nécessite le numéro de carte bancaire, le numéro CVV et le numéro de date d'expiration et après que l'utilisateur a appuyé sur le bouton de paiement, le formulaire enverra la demande au serveur avec les informations soumises par l'utilisateur et effectue une validation avec l'API de paiement afin d'accepter la demande de paiement et de supprimer l'utilisateur du tableau "liste noire".

Une autre protection a été appliquée à ce formulaire, le "{% csrf_token%}" empêchera tout utilisateur externe de détourner la requête du serveur et compromettre les informations de la carte, ce qui représente une erreur critique lors des processus de paiement.

La route Web chargée d'accepter ces demandes est la route "/ submit" dans le fichier "urls.py", cette route est associée à la fonction "pay()" dans le fichier de vue qui gèrera le processus de paiement et supprimera l'utilisateur de la base de données si le paiement a réussi.

```
from django.db import connection
from django.conf import settings
from django.core.mail import EmailMultiAlternatives
from django.template.loader import get_template
from django.template import Context

def pay(request):
    if request.method == "POST":
        ID = request.POST.get("ID")
        cursor = connection.cursor()
        cursor.execute('DELETE FROM NO_MASK_CITIZENS WHERE ID={}'.format(ID))
        email = Citizens_data.objects.filter(NATID = ID).values('email', 'NAME', 'ADRESS')
        data = email[0]
        print(data['NAME'], data['ADRESS'])

        subject, from_email, to = 'Payment Notice', settings.EMAIL_HOST_USER, str(data['email'])
        text_content = "Payment Notice"
        html_content = "<h5>We are happy to inform you that your payment was success, below you will find informations about you</h5><hr /><p>"
        msg = EmailMultiAlternatives(subject, text_content, from_email, [to])
        msg.attach_alternative(html_content, "text/html")
        msg.send()

    return render(request, "main/message.html")
```

Figure 23 - La fonction de paiement

5 - Base Des Données SQL:

La base de données SQL a été utilisée pour gérer toutes les opérations SQL, notamment la mise à jour, l'enregistrement, l'insertion et la suppression des informations utilisateur. La base de données est utilisée à la fois par le site Web et par l'intelligence artificielle pour gérer

toutes les tâches précédentes, par conséquent, l'utilisation de requêtes SQL est essentielle pour manipuler la base de données.

Le nom de la base de données est "Citoyens"

et il contient deux tables importantes "main_1_citizens_data" qui contient toutes les informations relatives à l'utilisateur telles que l'adresse e-mail, le numéro d'identification national, la date de naissance, le nom et l'adresse et la seconde est "NO MASK CITIZENS" qui est essentiellement une liste noire qui contient tous les numéros d'identification nationaux de les peuples qui ont été capturés ne portant pas de masques par l'intelligence artificielle.

La table "main_1_citizens_data" est en mode lecture uniquement par l'algorithme et le site web, ce qui signifie que seul l'opérateur humain peut modifier les colonnes de cette table mais d'autre part la table "NO MASK CITIZENS" est disponible pour la manipulation à la fois par l'algorithme et le site web.

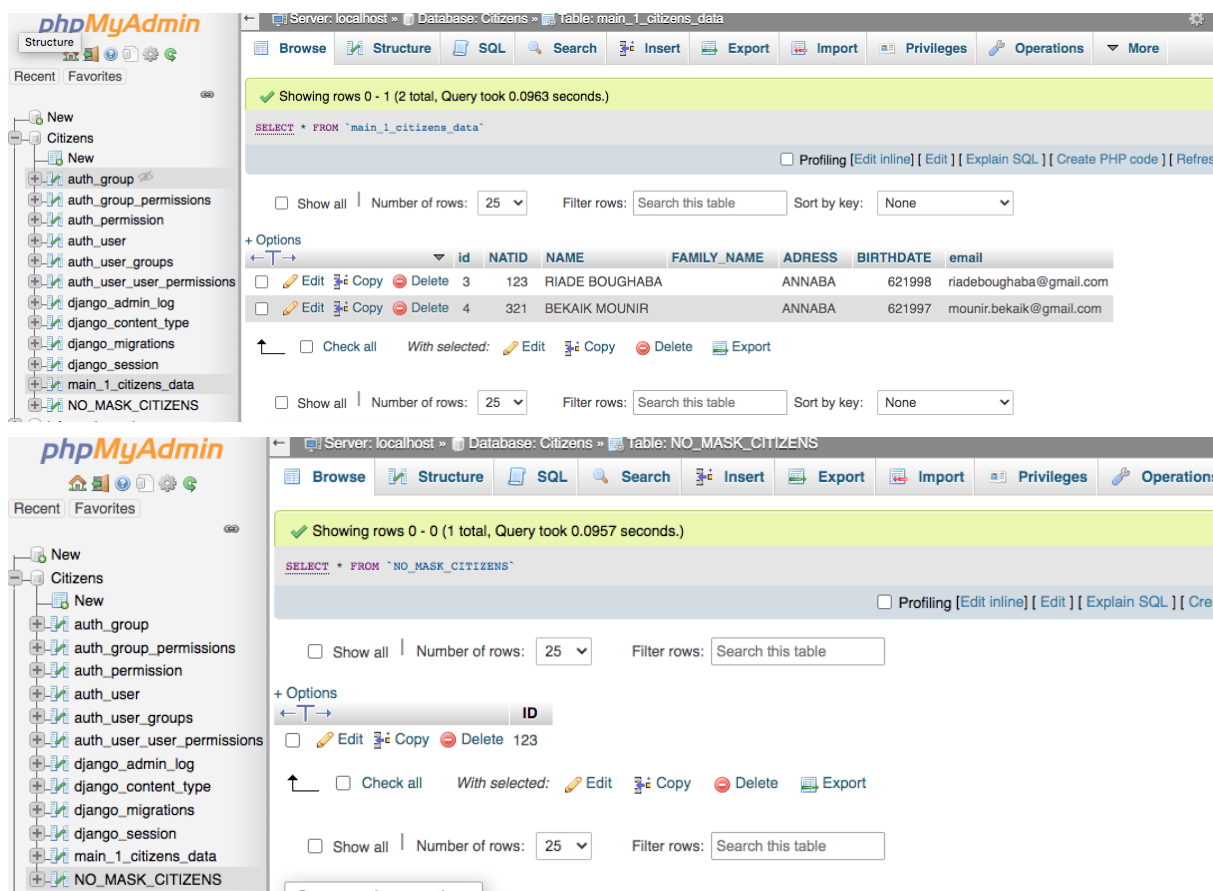


Figure 24 - Table SQL

Cette table peut accepter les opérations d'insertion de l'algorithme uniquement et supprimer les opérations du site Web uniquement, l'algorithme peut ajouter une ligne à cette table mais ne peut pas supprimer ou mettre à jour les colonnes et de même, le site Web ne peut supprimer les colonnes de cette table qu'une fois l'utilisateur effectué le paiement seulement.

Quant aux principales opérations SQL utilisées à la fois par l'algorithme et le site Web sont répertoriées dans le tableau ci-dessous:

	Opération SQL	Résultat
1	DELETE FROM NO_MAS_CITIZENS WHERE ID={}	Supprimer la colonne des citoyens sans masque où le numéro d'identification est égal à la valeur entre "{}"
2	INSERT INTO NO_MASK_CITIZENS(ID) VALUES({})	Insérer une nouvelle colonne dans la table "NO MASK CITIZENS" chère le numéro d'identification est égal à la variable "ID" et les informations sont passées entre "{}"
3	SELECT ADDRESS FROM main_1_citizens_data WHERE NATID={}	Sélectionnez la colonne d'adresse dans le tableau "main_1_citizens data" où le numéro d'identification national est égal à la valeur entre "{}"
4	SELECT NAME FROM main_1_citizens_data WHERE NATID={}	Sélectionnez la colonne de nom dans le tableau "main_1_citizens data" où le numéro d'identification national est égal à la valeur entre "{}"

Tableau 2 : Principales requêtes SQL

CHAPITRE 3: Testé et validation

1 – Prise d'image :

La webcam est connectée à un algorithme qui peut lire les données, les analyser en se basant sur la reconnaissance faciale.

La première étape consiste à démarrer la webcam

```
while(True):
    ret, frame = cap.read()
    if not ret:
        break

    currentImg = frame.copy()
    currentImg = resize(currentImg, width=980)
    imageShape = currentImg.shape
    fno += 1

    if(fno%1 == 0 or fno == 1):
        detectorObj.maskSetup()
        detectorObj.maskProcess(currentImg, minH, minW)
        outputFrame = detectorObj.maskDisplay()

        if writer is None:
            print("Writing the output file to: ", opname)
            fourcc = cv2.VideoWriter_fourcc(*"MJPG")
            writer = cv2.VideoWriter(opname, fourcc, 30,
                                     (outputFrame.shape[1], outputFrame.shape[0]), True)

        cv2.imshow('Mask Detection Dashboard', outputFrame)

    if videoMode:
        writer.write(outputFrame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Figure 25 - Code de détection principal

La ligne "cap.read ()" est responsable de l'ouverture de la webcam dès que python exécutera cette ligne de code, le système d'exploitation créera une fenêtre d'interface graphique python temporaire pour être le conteneur du flux vidéo afin que l'utilisateur puisse facilement interagir avec la fenêtre facilement.

La prochaine ligne importante est "resize (currentImg, width = 980)" qui gère les dimensions de la fenêtre, dans ce cas la dimension est carrée avec une longueur de 980 pixels.

À ce stade, l'étape finale consiste à transmettre les données à la fonction responsable du traitement de l'image, la ligne «mask Process (currentImg, minH, minW)» est chargée de transmettre les paramètres à la fonction de processus pour analyse.

Après avoir exécuté toutes les étapes nécessaires, le résultat ressemblera à ceci:

Si le visage est reconnu comme portant un masque facial, le résultat ressemblera à ceci:

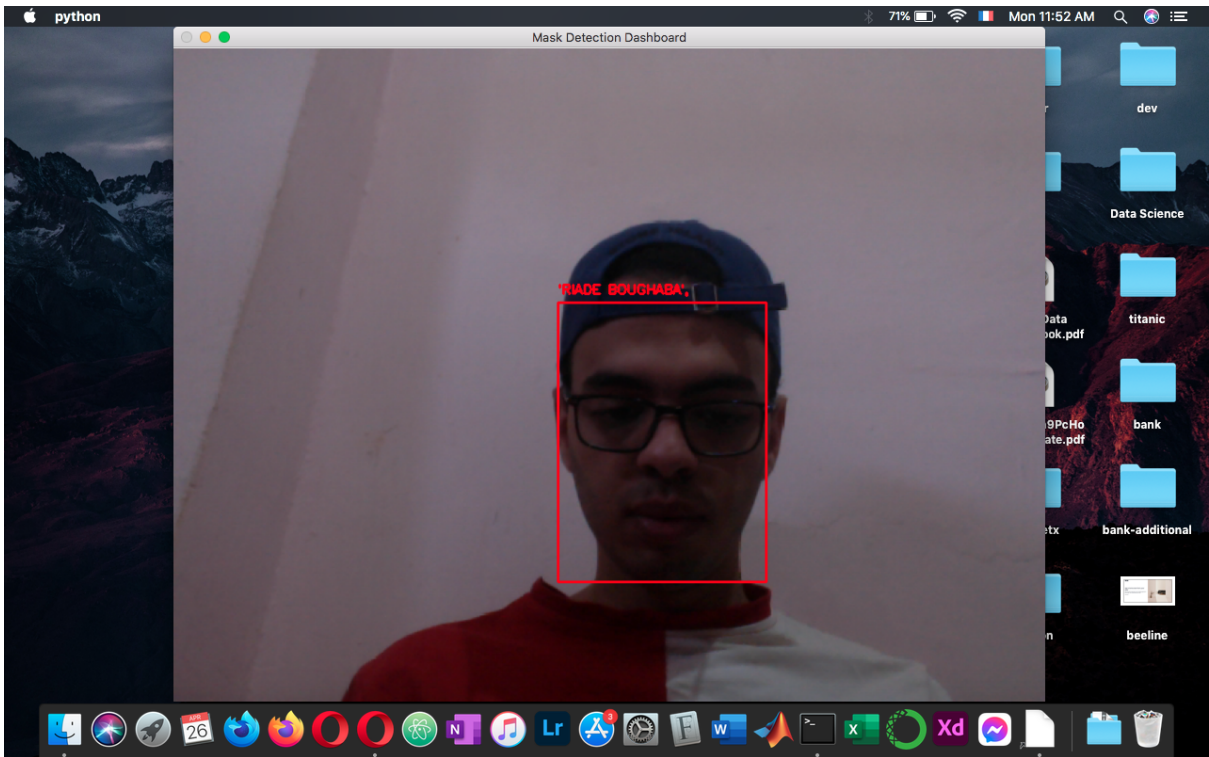


Figure 26 - Cas d'absence de masque

En revanche, si la personne ne porte pas de masque, le résultat sera le suivant:

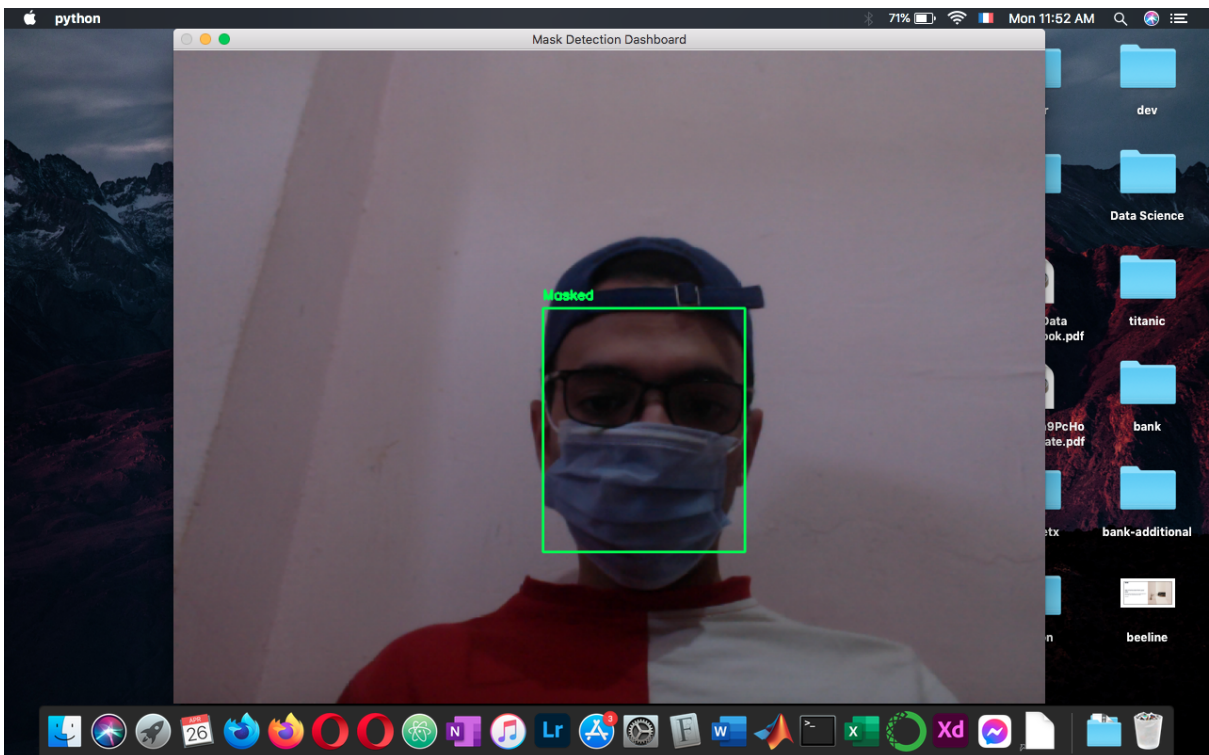
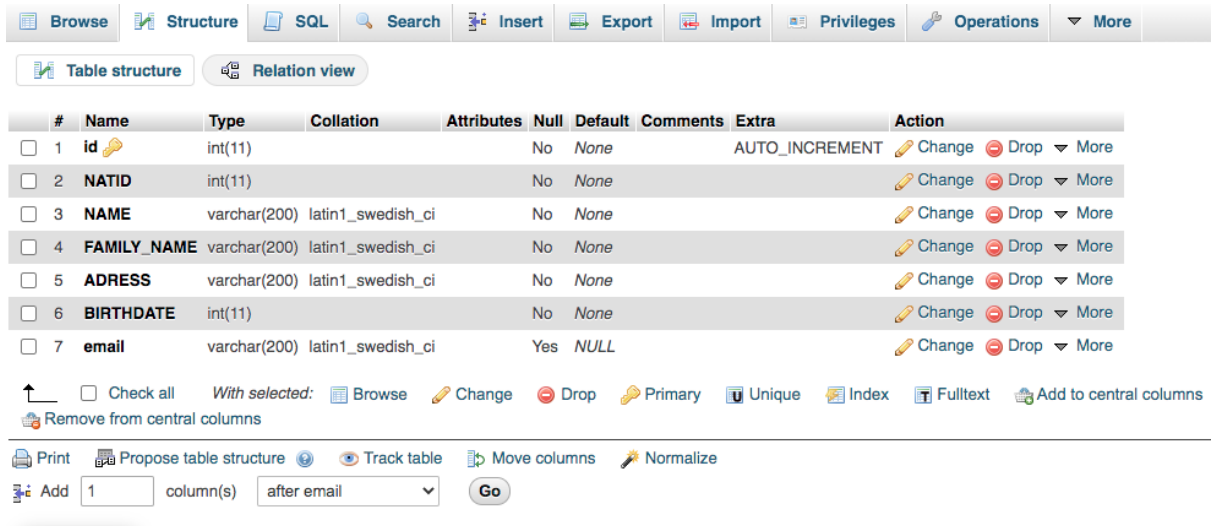


Figure 27 - cas de détection de masque

2 – Création de la Base Des Données :

Pour ce projet, nous avons créé deux tables de base de données principales "main_1_cirizens data" qui contient toutes les informations nécessaires sur les personnes qui seront utilisées plus tard par l'algorithme et le site Web et la liste noire des citoyens non porteur du masque "NO_MASK_CITIZENS".

Le premier tableau "main_1_cirizens data" qui contient des informations personnelles telles que la date de naissance, l'adresse e-mail, le nom complet et le numéro d'identification national est structuré par colonnes, chaque colonne a un ensemble de propriétés spécifiques.



The screenshot shows a database management interface with a table structure view. The table has 7 columns: id, NATID, NAME, FAMILY_NAME, ADRESS, BIRTHDATE, and email. Each column has specific properties like type, collation, nullability, and default values. The interface includes various toolbars for actions like 'Change', 'Drop', 'More', 'Primary', 'Unique', 'Index', 'Fulltext', 'Add to central columns', 'Remove from central columns', 'Print', 'Propose table structure', 'Track table', 'Move columns', and 'Normalize'. There is also a 'Go' button at the bottom.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	NATID			No	None			Change Drop More
<input type="checkbox"/>	3	NAME	varchar(200) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4	FAMILY_NAME	varchar(200) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	5	ADRESS	varchar(200) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	6	BIRTHDATE	int(11)		No	None			Change Drop More
<input type="checkbox"/>	7	email	varchar(200) latin1_swedish_ci		Yes	NULL			Change Drop More

Tableau 3 - Architecture de la table principale des citoyens

La longueur entière maximale autorisée est de 11 et le nombre maximal de caractères dans la colonne de type chaîne est de 255 comme indiqué ci-dessus dans l'image, la colonne "id" est une colonne spéciale car il s'agit d'une colonne à incrémentation automatique utilisée par la base de données pour attribuer chaque identifiant unique à chaque nouvelle création de colonne.

Le même processus a été suivi lors de la création de la structure des colonnes "NO_MASK_CITIZENS":

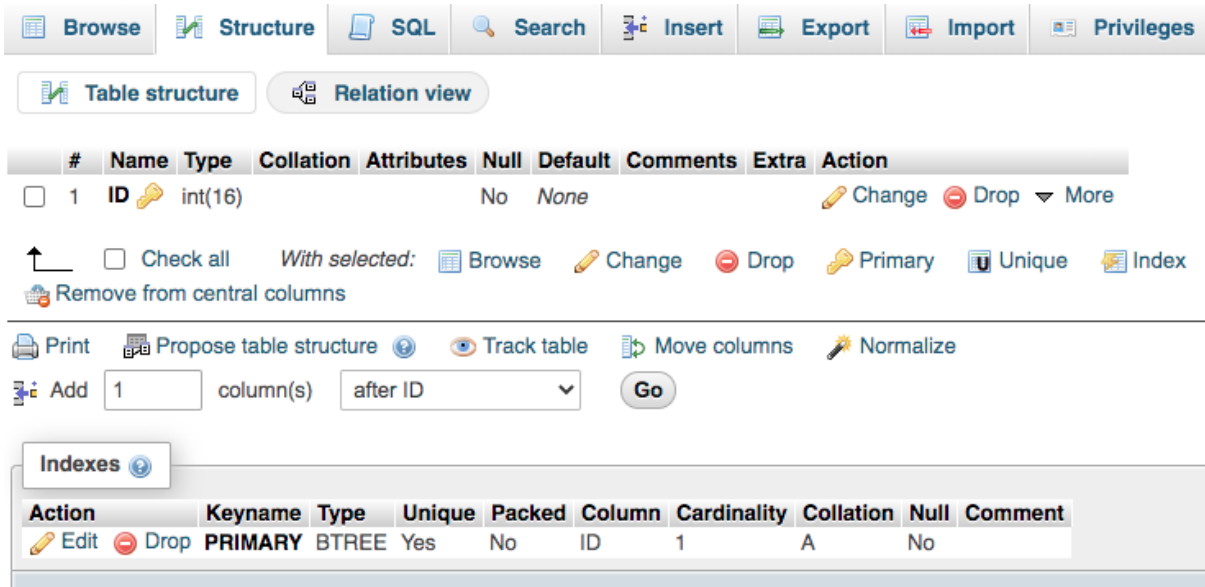


Tableau 4 - Table de liste noire

"NO_MASK_CITIZENS" contient une seule colonne "id" qui est utilisée par l'algorithme pour enregistrer le numéro d'identification national de ceux qui ne portent pas de masques qui est de type entier avec 16 nombres maximum pour chaque entrée.

L'interface graphique "phpmyadmin" permet aux développeurs d'exécuter facilement des requêtes SQL en cliquant simplement sur le bouton pour modifier, mettre à jour, supprimer ou insérer une nouvelle colonne de table:

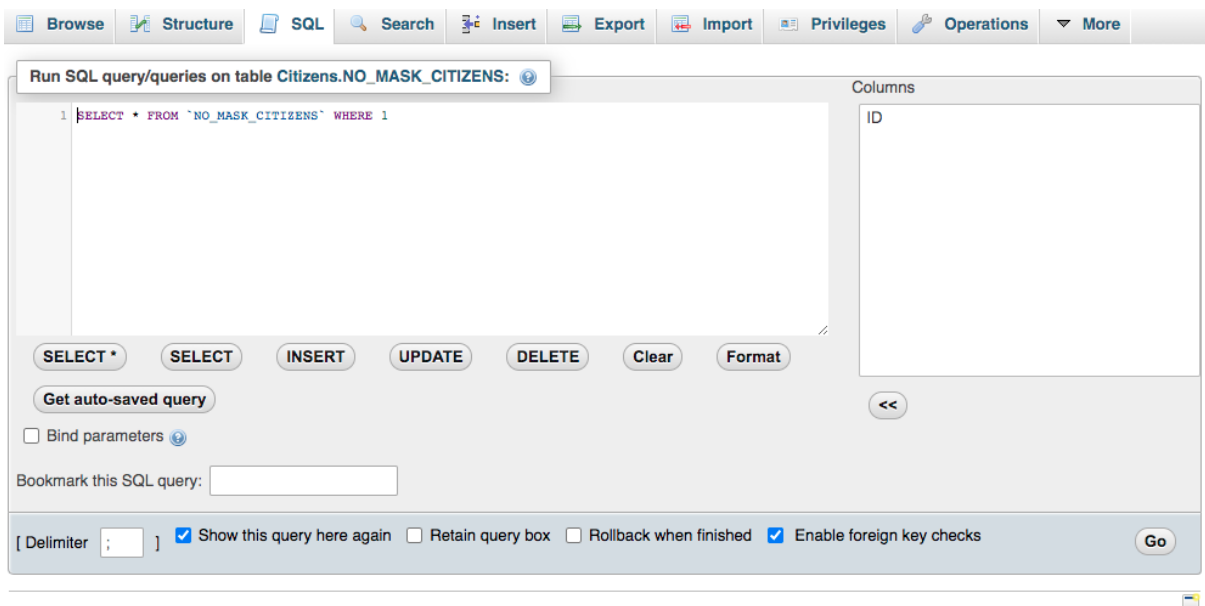
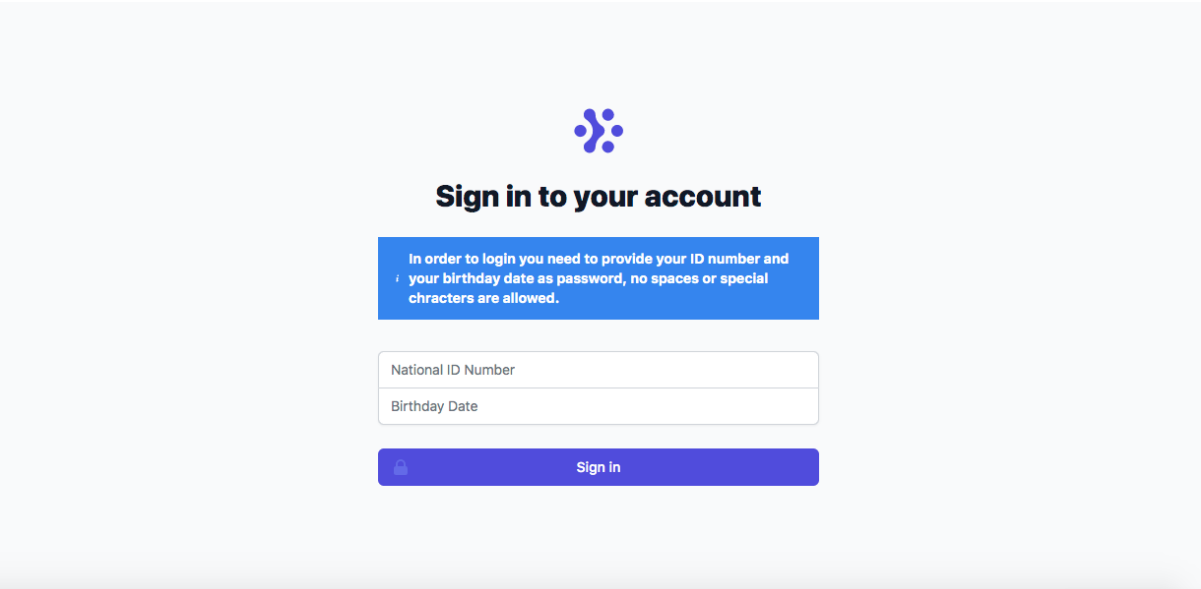


Figure 28 : Interface utilisateur graphique de l'éditeur SQL Phpmyadmin

3 - Le site de paiement :

Le site Web "COVID 19 EYE" a été conçu pour permettre aux personnes d'effacer leurs noms de la liste noire, le site Web a été construit en utilisant html, css (tailwind css framework) comme principales technologies frontend et django comme framework backend pour gérer toutes les tâches du serveur telles que traiter les données de paiement, manipuler la base de données, envoyer des e-mails et authentifier les utilisateurs avec la plate-forme.

En ce qui concerne le site Web, la page principale est la page de connexion où les utilisateurs saisissent les informations nécessaires pour être authentifiés auprès de la plate-forme telles que le numéro d'identification national et la date de naissance.



Sign in to your account

In order to login you need to provide your ID number and your birthday date as password, no spaces or special characters are allowed.

National ID Number

Birthday Date


 Sign in

Figure 29 - Page de connexion

Le code html complet de cette page Web est ci-dessous:

```

<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<div
  class="min-h-screen flex items-center justify-center bg-gray-50 py-12 px-4 sm:px-6 lg:px-8"
>
  <div class="max-w-md w-full space-y-8">
    <div>
      
      <h2 class="my-6 text-center text-3xl font-extrabold text-gray-900">
        Sign in to your account
      </h2>
      <div
        class="flex items-center bg-blue-500 text-white text-sm font-bold px-4 py-3"
        role="alert"
      >
        <svg
          class="fill-current w-4 h-4 mr-2"
          xmlns="http://www.w3.org/2000/svg"
          viewBox="0 0 20 20"
        >
          <path
            d="M12.432 0c1.34 0 2.01 1.12 2.01 1.957 0 1.305-1.164 2.512-2.679 2.512-1.269 0-2.009-.75-1.974-1.99C9.789 1.436 10.67 0"
          />
        </svg>
      <p>

```

```

        In order to login you need to provide your ID number and your
        birthday date as password, no spaces or special characters are
        allowed.
      </p>
    </div>
  </div>
  <div>
    <form class="mt-8 space-y-6" action="" method="POST">
      {% csrf_token %}
      <input type="hidden" name="remember" value="true" />
      <div class="rounded-md shadow-sm -space-y-px">
        <div>
          <label for="email-address" class="sr-only">
            National ID Number</label>
          <input
            id="email-address"
            name="ID"
            required
            class="appearance-none rounded-none relative block w-full px-3 py-2 border border-gray-300 placeholder-gray-500 text-gray-900"
            placeholder="National ID Number"
          />
        </div>
        <div>
          <label for="password" class="sr-only">Birthday Date</label>
          <input
            id="password"
            name="PASS"
            type="password"
            autocomplete="current-password"
            required
            class="appearance-none rounded-none relative block w-full px-3 py-2 border border-gray-300 placeholder-gray-500 text-gray-900"
            placeholder="Birthday Date"
          />
        </div>
      </div>
    </form>
  </div>

```

```

</div>
</div>
<div>
  <button
    type="submit"
    class="group relative w-full flex justify-center py-2 px-4 border border-transparent text-sm font-medium rounded-md text-white
  >
    <span class="absolute left-0 inset-y-0 flex items-center pl-3">
      <!-- Heroicon name: lock-closed -->
      <svg
        class="h-5 w-5 text-indigo-500 group-hover:text-indigo-400"
        xmlns="http://www.w3.org/2000/svg"
        viewBox="0 0 20 20"
        fill="currentColor"
        aria-hidden="true"
      >
        <path
          fill-rule="evenodd"
          d="M5 9V7a5 5 0 010 2v2a2 2 0 01-2 2H5a2 2 0 01-2-2v-5a2 2 0 012-2zm8-2v2a2 2 0 01-2 2H8a2 2 0 01-2-2v-5a2 2 0 012-2z"
          clip-rule="evenodd"
        />
      </svg>
    </span>
    Sign in
  </button>
</div>
</form>
</div>
</div>
</body>
</html>

```

Figure 30 - Code HTML de la page de connexion

Quant à la page où la personne verra ses informations personnelles et le formulaire de paiement est ci-dessous:

Your Pending Payments:

RIADE BOUGHABA

You have a ticket with 10000.00 DA for not wearing your mask and not respecting the law, therefore you must pay this amount here with your credit card or at the nearest police station, thank you for your concerns.

AMOUNT: 10000.00 DA

FULL NAME: RIADE BOUGHABA

ADRESS: ANNABA

ENTER YOUR CREDIT CARD INFORMATIONS

PAY 10000.00 DA

Figure 31 - Page de paiement

Le code HTML / CSS complet de cette page Web est ci-dessous:


```

<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<nav>
</nav>
<section
class="h-screen w-screen bg-gray-100 flex flex-col items-start justify-center p-10"
>
<h1 class="text-2xl mb-5 font-medium">Your Pending Payments:</h1>
<div class="w-full bg-white shadow-md rounded-md self-center p-10">
<h1 class="font-bold tracking-tight text-xl">{{data.NAME}}</h1>
<p class="text-gray-500 mt-3">
You have a ticket with 10000.00 DA for not wearing your mask and not
respecting the law, therefore you must pay this amount here with your
credit card or at the nearest police station, thank you for your
concerns.
</p>
<p class="font-bold text-lg mt-5">
AMOUNT:
<span class="text-lg font-bold text-blue-500 mt-4 font-mono"
>10000.00 DA</span>
</p>
<p class="font-bold text-lg mt-2">
FULL NAME:
<span class="text-lg font-bold text-blue-500 mt-4"
>{{data.NAME}}</span>
</p>
<p class="font-bold text-lg mt-2">
ADRESS:
<span class="text-lg font-bold text-blue-500 mt-4"
>{{data.ADDRESS}}</span>
</p>
<p class="font-bold text-sm uppercase underline mt-10">
Enter your credit card informations
</p>
<form action="submit/" method="POST" class="mt-1">
{csrf_token%}
<input type="text" name="ID" class="hidden" value="{{data.NATID}}" />
<input
type="number"
required
class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
placeholder="Credit Card Number"
maxlength="16"
/>
<div class="flex justify-center mt-3">
<input
type="text"
required
class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
placeholder="End Date"
/>
<input
type="text"
required
class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
placeholder="CCV"
/>
</div>
<button
type="submit"
class="w-full p-2 bg-blue-500 rounded my-3 text-white"

```

Figure 32 - Code HTML de la page de paiement

Les informations sont récupérées de la base de données lorsque l'utilisateur fournit ses informations personnelles sur la page de connexion, le serveur validera les informations et redirigera la personne authentifiée vers cette page afin de démarrer le processus de paiement.

Si le processus de paiement a été effectué avec succès, django redirigera l'utilisateur vers cette page de réussite indiquant que le processus a réussi :

The Process Was Successful, Thanks

Après l'affichage de cette page, une autre étape de vérification est faite en envoyant cet e-mail contenant toutes ses informations afin qu'il / elle puisse s'assurer que le paiement a été validé et que son identifiant a été retiré de la liste noire.

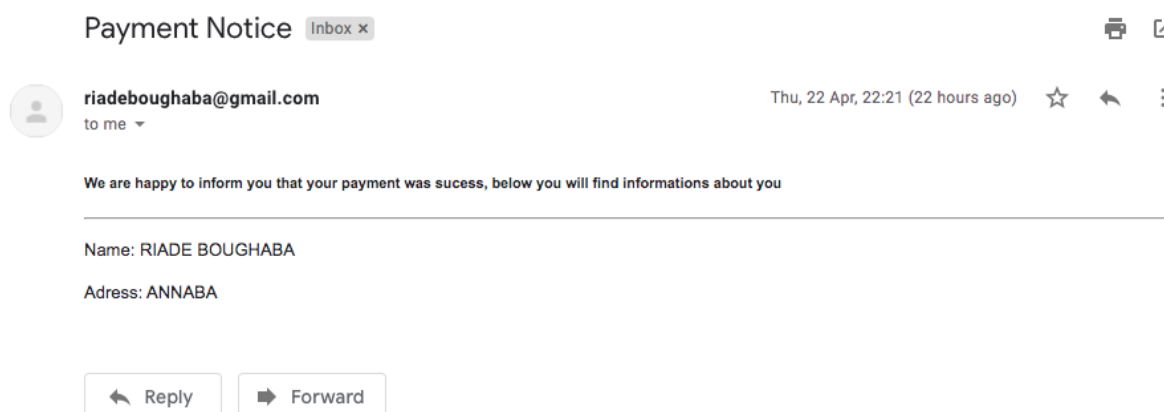


Figure 33 - Notification par e-mail

4 - Le processus de paiement :

Dans ce projet, le processus de paiement dépend strictement du paiement en ligne API qui est lié au ministère des communications et du courrier. Ce processus nécessite certains paramètres remplis par l'utilisateur pour réussir, d'abord la personne doit avoir une carte de crédit ou une carte CIB prenant en charge les clients en ligne, l'utilisateur pourra s'authentifier au le site Web, pour entrer les informations de la carte CIB telles comme le numéro de la

carte, la date de l'expiration et le numéro CVV qui figurent au dos de la carte. Après cela, l'utilisateur appuiera sur le bouton de paiement et le site Web enverra les informations à valider avec l'API et si la validation est réussie, l'API renverra un accusé de réception et le montant du paiement sera automatiquement prélevé sur la carte de la personne.

ENTER YOUR CREDIT CARD INFORMATION

Credit Card Number	
End Date	CCV
PAY 10000.00 DA	

Cette API doit être connectée aux serveurs du gouvernement afin que ce processus puisse être validé et détecte automatiquement les détails complets de la personne à partir des informations fournies, le site Web doit être connecté avec un certificat HTTPS valide qui crypte toutes les informations entre le client et le serveur afin de protéger les informations fournies.

Le processus pour obtenir les données utilisateur sur le serveur est assez simple, il doit d'abord y avoir un formulaire HTML qui envoie des informations sous forme de requête "POST" au serveur.

```
main > templates > main > dashboard.html > ...
<span class="text-lg font-bold text-blue-500 mt-4"
  >{{data.ADRRESS}}</span
>
</p>
<p class="font-bold text-sm uppercase underline mt-10">
  Enter your credit card informations
</p>
<form action="submit/" method="POST" class="mt-1">
  {csrf_token%}
  <input type="text" name="ID" class="hidden" value="{{data.NATID}}" />
  <input
    type="number"
    required
    class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
    placeholder="Credit Card Number"
    maxlength="16"
  />
</div class="flex justify-center mt-3">
  <input
    type="text"
    required
    class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
    placeholder="End Date"
  />
  <input
    type="text"
    required
    class="w-full p-2 focus border-2 border-gray-100 focus:border-blue-600 w-1/2"
    placeholder="CCV"
  />
</div>
<button
  type="submit"
  class="w-full p-2 bg-blue-500 rounded my-3 text-white"
```

Figure 34 - Code HTML du paiement

Le formulaire contient 3 entrées : numéro de carte de crédit, date d'expiration et le CVV. Ces 3 entrées contiennent la balise "OBLIGATOIRE" qui est un moyen de demander à la personne à saisir les informations avant d'envoyer les informations à l'API pour validation. Afin de sécuriser les informations du formulaire ; il consiste à inclure le "{% csrf_token%}" qui sécurise les informations contre les attaques "Cross-Site Request Forgery", cette attaque peut permettre à des personnes non autorisées d'obtenir des informations de paiement et de voler les informations de la carte de crédit, cette étape n'est pas requise par l'API mais elle est essentielle lorsqu'il s'agit de protéger les informations privées.

Conclusion :

Dans ce mémoire, nous avons présenté une approche de contrôle du port de masque très importante en se basant sur la programmation orientée objet qui se repose sur l'intelligence artificielle d'un côté et les bases de données. L'algorithme proposé détecte d'abord le visage de cette personne, puis vérifie si un masque est mis ou non, s'il y a un masque, l'algorithme n'effectuera aucune action supplémentaire ; en revanche ; en l'absence de celui-ci, l'algorithme reconnaît le visage et rajoute son identifiant national à la liste noire, la personne concerné reçoit les information de sa pénalité ainsi que le site de paiement afin de régler sa situation où il peut s'authentifier et payer facilement la fraude via la carte de crédit CIB.

En termes de perspective ; nous pourrons aussi proposer une approche de détection à distance de température afin de prévenir la contamination et céder contre la propagation de la maladie en utilisons notre approche.

Bibliographie:

- [1] **Intelligence Artificielle**, *L'intelligence artificielle (IA)*: https://fr.wikipedia.org/wiki/Intelligence_artificielle
- [2] **Big Data**, *Qu'est-ce que le Big Data?*: <https://www.lebigdata.fr/definition-big-data>
- [3] **Développement Web**, *Le Développement Web, C'est Quoi ?*:
<https://www.net-concept.fr/actualites/le-developpement-web-cest-quoi/>
- [4] **HTML**, *HTML (HyperText Markup Langage) : définition, traduction*:
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/>
- [5] **CSS**, *C'est quoi le CSS?*: <https://myhappyagency.com/blog/cest-quoi-le-css/>
- [6] **Javascript**, *Javascript : définition simple et applications pratiques*:
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/>
- [7] **algorithm**, *Algorithme*: <https://fr.wikipedia.org/wiki/Algorithme>
- [8] **Reconnaissance Faciale**, *Reconnaissance faciale : qu'est-ce que c'est et quels sont les dangers ?*:
<https://www.lebigdata.fr/reconnaissance-faciale-tout-savoir>
- [9] **Database**, *Base de données : qu'est-ce que c'est ? Définition et présentation*:
<https://www.lebigdata.fr/base-de-donnees>
- [10] **SQL**, *C'est quoi le SQL ?*: <https://www.culture-informatique.net/cest-quoi-sql/>
- [11] **Python**, *Python : définition et utilisation de ce langage informatique*:
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>
- [12] **API**, *Qu'est-ce qu'une API ?*:
<https://www.redhat.com/fr/topics/api/what-are-application-programming-interfaces>
- [13] **HTTP**, *Que signifie HTTP?*: https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [14] **HTTPS**, *HTTPS : ce que cela signifie et pourquoi c'est important*:
<https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/le-https-cest-quoi/>
- [15] **Server**, *Serveur informatique*: https://fr.wikipedia.org/wiki/Serveur_informatique
- [16] **Development Backend**, *Comment devenir développeur back-end*:
<https://blog.freelancerepublik.com/developpeur-back-end/>
- [17] **Django**, *Présentation de django*: <https://python.doctor/page-django-introduction-python>
- [18] **Apache SQL Server**, *Qu'est ce que Apache ? Une description complète du Serveur Web Apache*:
<https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/>
- [19] **CNN**, *Convolutional neural network*: https://en.wikipedia.org/wiki/Convolutional_neural_network
- [20] **Framework**, *Framework*: <https://fr.wikipedia.org/wiki/Framework>
- [21] **CSRF**, *CSRF : le Cross Site Request Forgery*:
<https://www.ionos.fr/digitalguide/serveur/securite/cross-site-request-forgery/>
- [22] **XSS**, *Se protéger des failles de sécurité XSS/cross-site scripting*:
<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/quest-ce-que-le-xss-cross-site-scripting/>