

Laura Romero Huete

Aplicació d'escriptori pel control de l'aforament amb C++

TREBALL DE FI DE GRAU

dirigit per Hatem Abdellatif Fatahallah Ibrahim Mahmoud

Doble titulació de Grau d'Enginyeria Informàtica i Biotecnologia



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2021

Resum.

Aquest projecte es basa en una aplicació d'escriptori que té com a objectiu poder controlar l'aforament en un local o establiment a través d'una càmera en directe. A causa de la situació social causada per la COVID-19 en la que ens trobem avui en dia, el control de l'aforament és un aspecte molt controlat avui en dia. Es tracta d'una aplicació que captura la imatge d'una càmera i analitza els frames. A partir d'aquests frames gràcies a la llibreria d'OpenCV es detecta si hi ha una persona dins del pla i si aquesta està entrant o sortint del local. El resultat es mostrarà per pantalla per tal que l'usuari pugui veure-ho fàcilment.

Per desenvolupar l'aplicació s'ha fet servir el Visual Studio 2019 i la llibreria cvui per fer la interfície gràfica, i la llibreria d'OpenCV per analitzar la imatge i el llenguatge de programació C++ .

Resumen.

Este proyecto se basa en una aplicación de escritorio que tiene como objetivo poder controlar el aforo en un local o establecimiento a través de una cámara en directo. Debido a la situación social causada por la COVID-19 en la que nos encontramos hoy en día, el control del aforo es un aspecto muy controlado hoy en día. Se trata de una aplicación que captura la imagen de una cámara y analiza los frames. A partir de estos frames gracias a la librería de OpenCV se detecta si hay una persona dentro del plan y si ésta está entrando o saliendo del local. El resultado se mostrará por pantalla para que el usuario pueda verlo fácilmente.

Para desarrollar la aplicación se ha utilizado el Visual Studio 2019 y la librería cvui para hacer la interfaz gráfica, y la librería de OpenCV para analizar la imagen y el lenguaje de programación C ++.

Abstract.

Due to the social situation caused by COVID-19 in which we find ourselves today, capacity control is a very controlled aspect today. Thus, in this project, we aim at developing a desktop application that is able to control the capacity in a shop or establishment through a live camera. This application can capture a live video of a surveillance camera and analyze the input frames. From these frames, thanks to the OpenCV library, the application can detect if there is a person inside the surveilled region and if he is entering or leaving the establishment. The result will be shown to the application's user on the screen so that he/she can easily monitor and count the number of people in the establishment. Besides, the system can automatically determine if the establishment has a full capacity or more people can enter.

Visual Studio 2019 and the CVUI library were used to develop the application to implement the graphical interface (GUI) of the application, and the OpenCV library is used to analyze the image to detect and count the people appearing in the scene.

Índex

1	INTRODUCCIÓ	8
1.1	MOTIVACIÓ	9
2	DESCRIPCIÓ GENERAL DEL PROJECTE	10
2.1	GLOSSARI	10
2.2	OBJECTIUS DEL PROJECTE	10
2.3	OBJECTIUS DEL PROGRAMA	10
3	REQUISITS.....	12
3.1	REGLES DEL NEGOCI.....	12
3.2	DIAGRAMA DE CASOS D'ÚS	13
3.3	ESPECIFICACIÓ TEXTUAL DE CASOS D'ÚS	14
3.3.1	<i>Cdu 01 Seleccionar Aforament.....</i>	<i>14</i>
3.3.2	<i>Cdu 02 Seleccionar Entrada</i>	<i>14</i>
3.3.3	<i>Cdu 03 Iniciar Càmera.....</i>	<i>15</i>
3.3.4	<i>Cdu 04 Detectar Persones.....</i>	<i>15</i>
3.3.5	<i>Cdu 05 Controlar Aforament.....</i>	<i>16</i>
3.3.6	<i>Cdu 06 Carregar Benvinguda</i>	<i>16</i>
3.4	REQUISITS NO FUNCIONALS	17
3.4.1	<i>Eficiència.....</i>	<i>17</i>
3.4.2	<i>Accessibilitat</i>	<i>17</i>
3.4.3	<i>Usabilitat.....</i>	<i>17</i>
3.4.4	<i>Fiabilitat.....</i>	<i>17</i>
3.4.5	<i>Escalabilitat.....</i>	<i>17</i>
3.4.6	<i>Claredat del text.....</i>	<i>17</i>
4	DISSENY	19
4.1	ARQUITECTURA DEL PROGRAMA	19
4.1.1	<i>Llenguatge.....</i>	<i>19</i>
4.1.2	<i>Altres tecnologies</i>	<i>20</i>

4.2	DISSENY DE LA INTERFÍCIE GRÀFICA	27
4.2.1	<i>Pantalles</i>	27
4.3	DISSENY DE L'ESTRUCTURA DE L'APLICACIÓ.....	28
5	IMPLEMENTACIÓ.....	29
5.1	INTERFÍCIE GRÀFICA	29
5.1.1	<i>Pantalla principal</i>	29
5.1.2	<i>Pantalla de càmera</i>	32
5.2	FUNCIONALITAT DE L'APLICACIÓ	34
5.2.1	<i>Main</i>	34
5.2.2	<i>Start</i>	35
5.2.3	<i>Update_mhi</i>	36
5.2.4	<i>InitImages</i>	36
5.2.5	<i>Silhoutte_reduction</i>	37
5.2.6	<i>Model1</i>	37
5.2.7	<i>Model2</i>	37
5.2.8	<i>Reject_small_and_large_object</i>	38
5.2.9	<i>Detect_object</i>	39
6	AVALUACIÓ.....	40
6.1	JOC DE PROVES	40
6.2.1	<i>Detecció d'una sola persona entrant en una entrada horitzontal</i>	42
6.2.2	<i>Detecció d'una sola persona sortint en una entrada horitzontal</i>	42
6.2.3	<i>Detecció de més d'una persona en una entrada horitzontal</i>	43
6.2.4	<i>Detecció d'aforament complet en una entrada horitzontal</i>	43
6.2.5	<i>Detecció d'una sola persona entrant en una entrada vertical</i>	44
	44
6.2.6	<i>Detecció d'una sola persona sortint en una entrada vertical</i>	44
6.2.7	<i>Detecció de més d'una persona entrant en una entrada vertical</i>	45
6.2.8	<i>Detecció d'aforament complet en una entrada vertical</i>	45

7 CONCLUSIONS	46
8 REFERÈNCIES	48
9 ANNEXES	50
GUIA D'INSTAL·LACIÓ	50

Índex de taules

TAULA 1 JOC DE PROVES	41
-----------------------------	----

Índex de figures

FIGURA 1 HUMIMIC CHIP3, "MULTI-ORGAN-CHIP" DE TISSUSE GMB	9
FIGURA 2 DIAGRAMA DE CASOS D'ÚS	13
FIGURA 3 LOGOTIP C++	20
FIGURA 4 LOGOTIP DE VISUAL STUDIO	21
FIGURA 5 LOGOTIP D'OPENCV	21
FIGURA 6 APLICACIÓ BÀSICA DE CVUI AMB UN BOTÓ I UN TEXT	22
FIGURA 7 WANSVIEW CÁMARA VIGILANCIA	23
FIGURA 8 CÀMERA NIKON D3020	24
FIGURA 9 CÀMERA LOGITECH BRIO STREAM	24
FIGURA 10 CABLE USB TIPO C A USB TIPO A	25
FIGURA 11 CABLE MINI HDMI A HDMI	25
FIGURA 12 CAPTURADORA DE VÍDEO HDMI	26
FIGURA 13 ESBÓS DE LA PANTALLA PRINCIPAL DE LA GUI	27
FIGURA 14 ESBÓS DE LA PANTALLA DE LA CÀMERA DE LA GUI	28
FIGURA 15 PANTALLA PRINCIPAL DE LA GUI	30
FIGURA 16 PANTALLA DE LA CÀMERA DE LA GUI, AMB ENTRADA VERTICAL AL LOCAL	33
FIGURA 17 PANTALLA DE LA CÀMERA DE LA GUI, AMB ENTRADA HORITZONTAL AL LOCAL	33
FIGURA 18 DETECCIÓ D'UNA SOLA PERSONA ABANS D'ENTRAR EN UNA ENTRADA HORITZONTAL	42
FIGURA 19 DETECCIÓ D'UNA SOLA PERSONA DESPRÉS D'ENTRAR EN UNA ENTRADA HORITZONTAL	42
FIGURA 20 DETECCIÓ D'UNA SOLA PERSONA ABANS DE SORTIR EN UNA ENTRADA HORITZONTAL	42
FIGURA 21 DETECCIÓ D'UNA SOLA PERSONA DESPRÉS DE SORTIR EN UNA ENTRADA HORITZONTAL	42
FIGURA 22 DETECCIÓ DE DUES PERSONES ENTRANT EN UNA ENTRADA HORITZONTAL	43
FIGURA 23 DETECCIÓ DE DUES PERSONES SORTINT EN UNA ENTRADA HORITZONTAL	43
FIGURA 24 DETECCIÓ D'UNA SOLA PERSONA ABANS D'ENTRAR EN UNA ENTRADA HORITZONTAL QUAN NOMÉS POT ENTRAR UNA PERSONA MÉS.	43
FIGURA 25 DETECCIÓ D'UNA SOLA PERSONA DESPRÉS D'ENTRAR I COMPLETAR L'AFORAMENT DEL LOCAL EN UNA ENTRADA HORITZONTAL	43
FIGURA 26 DETECCIÓ D'UNA SOLA PERSONA ABANS D'ENTRAR EN UNA ENTRADA VERTICAL	44
FIGURA 27 DETECCIÓ D'UNA SOLA PERSONA DESPRÉS D'ENTRAR EN UNA ENTRADA VERTICAL	44

FIGURA 28 DETECCIÓ D'UNA SOLA PERSONA ABANS DE SORTIR EN UNA ENTRADA VERTICAL. 44

FIGURA 29 DETECCIÓ D'UNA SOLA PERSONA DESPRÉS DE SORTIR EN UNA ENTRADA VERTICAL. 44

FIGURA 30 DETECCIÓ DE DUES PERSONES ENTRANT EN UNA ENTRADA VERTICAL..... 45

FIGURA 31 DETECCIÓ DE DUES PERSONES SORTINT EN UNA ENTRADA VERTICAL..... 45

FIGURA 32 DETECCIÓ D'UNA SOLA PERSONA ABANS D'ENTRAR EN UNA ENTRADA VERTICAL QUAN NOMÉS POT ENTRAR
UNA PERSONA MÉS..... 45

FIGURA 33 DETECCIÓ D'UNA SOLA PERSONA DESPRÉS D'ENTRAR I COMPLETAR L'AFORAMENT DEL LOCAL EN UNA
ENTRADA VERTICAL..... 45

Índex de codi

CODI 1 INCLUDE DEL FITXER CVUI.H.....	31
CODI 2 DEFINICIÓ DE LA FINESTRA "PEOPLECOUNTER".....	31
CODI 3 INICIALITZACIÓ I CREACIÓ DE LA FINESTRA "PEOPLECOUNTER"	31
CODI 4 EXEMPLE DE COM AFEGIR ELEMENTS AL FRAME	31
CODI 5 EXEMPLE D'ESCRIURE INFORMACIÓ SOBRE UNA IPLIMAGE.....	32
CODI 6 CODI PER CANVIAR EL FORMAT DE LA IMATGE DE IPLIMAGE A MATRIU.....	32
CODI 7 ESTRUCTURA PRINCIPAL DE LA FUNCIO MAIN.....	35
CODI 8 FRAGMENT DE CODI DE LA FUNCIO START	36
CODI 9 INICIALITZACIÓ D'UNA IMATGE.....	36
CODI 10 CODI DE LA FUNCIO SILHOUTTE_REDUCTION.....	37
CODI 11 CODI DE LA FUNCIO REJECT_SMALL_AND_LARGE_OBJECT	38
CODI 12 CODI PER RECORRE ELS PÍXELS D'UNA FILA D'UNA IMATGE	39
CODI 13 CODI PER DETERMINAR LA DIRECCIÓ DE MOVIMENT D'UN OBJECTE	39

1 Introducció

Des de principis de 2020, ens trobem en una situació de pandèmia sanitària ocasionada pel virus SARS-CoV-2, un tipus de coronavirus causant la pandèmia de la COVID-19. Aquesta pandèmia ha obligat la població a canviar molts costums. A més, per tal de prevenir la malaltia hem hagut de prendre algunes mesures com a població, com rentar-se les mans amb aigua i sabó sovint, utilitzar mascareta i el distanciament social entre altres [1].

El distanciament social el vam tenir molt present quan a principis de 2020 l'estat d'alarma ens va obligar a estar tancats a casa realitzant teletreball i classes en línia. Això va fer que no tinguéssim contacte amb altres persones, reduint la interacció social entre persones. Ara que l'estat d'alarma ja no és vigent i ens trobem en una "nova normalitat" clar seguir prenent algunes mesures per tal d'intentar reduir els contagis de COVID-19 entre la població .

Algunes de les mesures que s'han pres en aquesta nova normalitat és la reducció de l'aforament en alguns locals tant interiors com exteriors. Ens hem acostumat a veure a les entrades dels comerços, dels locals de restauració i dels locals d'oci cartells indicant que tenen un aforament limitat.

Molts d'aquests establiments que anomeno controlen l'aforament de manera manual, comptant el nombre de persones que entren i el nombre de persones que surten del local per tal de controlar que no se superi l'aforament màxim del local.

Per evitar aquesta càrrega de treball que tenen actualment alguns dels treballadors dels comerços o dels locals de restauració, es planteja utilitzar la intel·ligència artificial per realitzar un programa que mitjançant una càmera detectés la quantitat de persones que entren i surten de l'establiment.

La intel·ligència artificial (IA) és una part de la informàtica, dedicada al desenvolupament d'algorismes que permeten a una màquina comportar-se com si tingués una intel·ligència humana. La IA és un terme usat per referir-nos a les aplicacions que executen tasques complexes que abans requerien una aportació humana.

Actualment, la tecnologia de IA millora el rendiment i la productivitat a les empreses mitjançant l'automoció de processos o tasques que abans requerien un esforç humà. El principi fonamental de la IA és replicar i superar la forma que les persones perceben i reaccionen davant del món. La IA impulsada per diverses formes de Machine learning que reconeixen patrons en les dades per permetre prediccions poden afegir valor al negoci:

- Proporcionant una comprensió més completa de l'abundància de dades disponibles.
- Confiant en les prediccions per automatitzar tasques complexes o mundanes [2].

1.1 Motivació

La motivació per fer aquest treball sorgeix després de realitzar unes pràctiques externes a l'empresa TissUse GmbH de Berlín. TissUse és una empresa biotecnològica amb seu a Alemanya, a Berlín, que ha desenvolupat una plataforma única "Multi-Organ-Chip", **Figura 1**, que proporciona una visió preclínica inigualable a escala sistèmica mitjançant teixits humans. Aquesta plataforma tecnològica habilitant consisteix en una construcció miniaturitzada que simula de prop l'activitat de múltiples òrgans humans en el seu veritable context fisiològic [3].

Allà juntament amb el meu company de classe Albert Cañellas, vam estar desenvolupant un software de reconeixement de cèl·lules per tal de poder-les detectar i rastrejar per tal de comprovar la direcció a la qual es movien dins dels microcanals dels "Organ-chips" que ells mateixos desenvolupaven.

Els "Multi-Organ-Chip" de TissUse proporcionen un nou enfocament per predir, per exemple, la toxicitat, els perfils ADME i l'eficàcia in vitro, reduint i substituint les proves amb animals de laboratori i racionalitzant els assajos clínics en humans. I són també un clar exemple on la biotecnologia i la informàtica s'uneixen.

Amb els coneixements que vam adquirir d'anàlisi d'imatges realitzant les pràctiques hem decidit realitzar el TFG conjuntament. Pel fet que avui en dia hi ha diversos llenguatges de programació amb els seus avantatges i els seus inconvenients, es va optar per desenvolupar el mateix programa amb llenguatges diferents i poder després comparar els resultats. A més, també es va optar per fer servir dos mètodes diferents per veure així també si algun és més eficaç que l'altre. L'aplicació desenvolupada amb C++ faria servir un mètode de subtracció de frames i l'aplicació desenvolupada amb Python faria servir Xarxes Neuronals per detectar persones.



Figura 1 HUMIMIC Chip3, "Multi-Organ-Chip" de TissUse GMB

2 Descripció general del projecte

2.1 Glossari

A continuació s'especifica el significat que se li han donat a algunes paraules al llarg del treball:

- **Usuari:** nom genèric donat a qualsevol persona que utilitzi l'aplicació,
- **Intel·ligència Artificial:** es refereix als sistemes o les màquines que imiten la intel·ligència humana per executar tasques i que tenen la capacitat de millorar iterativament a partir de la informació que recopilen [2].
- **Interfície gràfica d'usuari (GUI):** és l'entorn visual d'imatges i objectes mitjançant el qual una màquina i un usuari interactuen [4].
- **Càmera:** dispositiu o aparell destinat a impressionar o a reproduir imatges [5].
- **Covid-19:** la malaltia causada per el virus infeccions SARS-CoV-2 [1].
- **USB:** presa de connexió universal d'ús freqüent en els ordinadors i altres dispositius electrònics [6].
- **HDMI:** interfície multimèdia d'alta definició i és el senyal HD més utilitzat per transferir àudio i vídeo d'alta definició a través d'un sol cable [7].
- **Frame:** cada una de les imatges que formen un vídeo [8].
- **RGB:** El model de color RGB d'un color fa referència a la composició del color en termes de les intensitats dels colors primaris amb què es forma: vermell, verd i blau —que formen l'acrònim anglès (Red, Green, Blue) [9].
- **HSV:** El model HSV (de l'anglès Hue, Saturation, Value – tonalitat, saturació, valor), defineix un model de color en termes dels seus components constituents en coordenades cilíndriques [10].

2.2 Objectius del projecte

L'objectiu principal del projecte és desenvolupar una aplicació d'escriptori que et permeti controlar l'aforament d'un local, comptabilitzant a través d'una càmera les persones que entren i surten d'un local.

Altres objectius:

- Dissenyar i desenvolupar una interfície gràfica que permeti l'usuari controlar els paràmetres i visualitzar els resultats.
- Permetre l'anàlisi de vídeo en temps real.
- Comparar l'eficàcia del programa en diferents mètodes de detecció.

2.3 Objectius del programa

El principal objectiu del programa serà proporcionar informació de la quantitat de persones que entre i surten d'un local.

Les funcionalitats que ha de cobrir l'aplicació són:

Descripció general del projecte

- Control de l'aforament a temps real, coneixent el nombre exacte de persones a l'interior de l'establiment.
- Redefinir l'aforament màxim de persones en l'establiment.

3 Requisits

En aquest punt es detallaran les regles de negoci del projecte, així com el diagrama de casos d'us, i els requisits tant funcionals com no funcionals.

3.1 Regles del negoci

01. Sempre hi ha d'haver una càmera connectada.
02. La càmera ha d'estar fixe.
03. El valor per defecte d'entrada és "vertical"
04. El valor per defecte d'aforament màxim és de 4 persones
05. Els fotogrames per segon mínim és 30 FPS.
06. Es mostri els usuaris que han entrat a l'establiment.
07. Es mostri els usuaris que han sortit de l'establiment.
08. S'informi quan s'ha completat l'aforament.

3.2 Diagrama dels casos d'ús

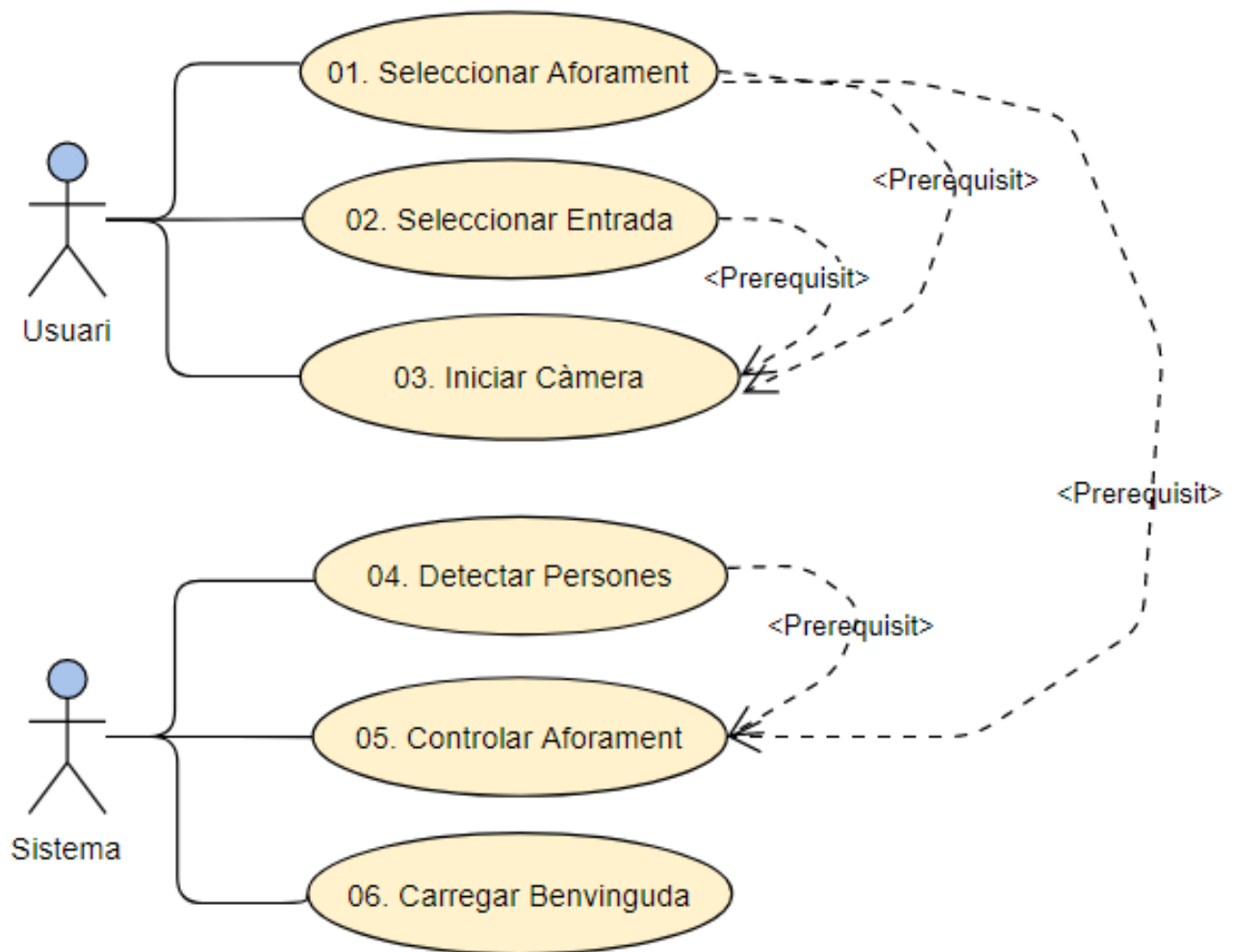


Figura 2 Diagrama de casos d'ús

En la Figura 1 es mostra el diagrama de casos d'ús des del punt de vista d'aquest TFG. Aquesta figura ha estat realitzada amb Visual Paradigm [11].

En el cas d'aquest treball algunes de les funcionalitats pel correcte funcionament de l'aplicació les assolix el sistema, que serien "04. Detectar Persones", "05. Controlar aforament" i "06. Carregar Benvinguda". Tot i que aquestes funcionalitats no les realitza cap actor pròpiament dit, són funcions importants per l'aplicació.

3.3 Especificació textual de casos d'ús

A continuació es detallen els diferents cassos d'ús que es mostren a la **Figura 2**.

3.3.1 Cdu 01 Seleccionar Aforament

Resum de la funcionalitat: Realitzar la selecció del número de persones que es vol que hi hagi com a màxim en el local.

Paràmetres d'entrada: Cap.

Paràmetres de sortida: Nombre de persones que es permet dins del local.

Actors: Usuari.

Precondició: Cap

Postcondició: Cap

Procés normal principal:

09. L'aplicació mostrarà una pantalla principal amb un número per defecte amb un botó de més (+) i un botó de menys (-).
10. L'usuari prem el botó (+) o (-) segons el seu interès.
11. L'aplicació anirà actualitat el número segons el que premi l'usuari.

3.3.2 Cdu 02 Seleccionar Entrada

Resum de la funcionalitat: Realitzar la selecció del tipus d'entrada del local, que podrà ser horitzontal o vertical.

Paràmetres d'entrada: Cap.

Paràmetres de sortida: Tipus d'entrada del local.

Actors: Usuari.

Precondició: Cap

Postcondició: Cap

Procés normal principal:.

01. L'aplicació mostrarà una pantalla principal on hi haurà un botó "Vertical" i un botó "horitzontal".
02. L'usuari pot prémer el botó que més li interessi.

3.3.3 Cdu 03 Iniciar Càmera

Resum de la funcionalitat: Funció que permet iniciar el funcionament del programa

Paràmetres d'entrada: Cap

Paràmetres de sortida: Cap

Actors: Usuari

Precondició: Cap

Postcondició: Imatge de càmera en directe.

Procés normal principal:

01. L'aplicació mostrarà una pantalla principal on hi haurà un botó
02. L'usuari pot prémer el botó per iniciar la càmera.

3.3.4 Cdu 04 Detectar Persones

Resum de la funcionalitat: Detectar persones a partir de la imatge captada per una càmera en directe.

Paràmetres d'entrada: Cap

Paràmetres de sortida: Número de persones detectades.

Actors: Sistema

Precondició: Que es llegeixi la imatge a través de la càmera

Postcondició: Cap

Procés normal principal:

01. El sistema llegirà els frames de la càmera i els analitzarà per detectar persones.
02. Si s'ha detectat una persona es comprovarà si aquesta està entrant o sortint del local.
03. El sistema mostrarà per pantalla el número de persones que han entrat i/o sortir del local.

3.3.5 Cdu 05 Controlar Aforament

Resum de la funcionalitat: Càlcul del número de persones que encara poden entrar al local sense superar l'aforament.

Paràmetres d'entrada: Límit d'aforament, i número de persones que han entrat al local.

Paràmetres de sortida: Número de persones que encara poden entrar al local sense superar l'aforament.

Actors: Sistema

Precondició: Que s'hagin detectat persones

Postcondició: Cap

Procés normal principal:

01. El sistema controlarà a partir del número de persones detectades que han entrat al local.
02. El sistema mostrarà per pantalla el número de persones que encara poden entrar al local segons l'aforament permès que hagi introduït l'usuari.

3.3.6 Cdu 06 Carregar Benvinguda

Resum de la funcionalitat: Carregar la pantalla de

Paràmetres d'entrada: Aforament i tipus d'entrada del local per defecte

Paràmetres de sortida: Aforament i tipus d'entrada del local seleccionat per l'usuari.

Actors: Sistema

Precondició: Cap

Postcondició: Cap

Procés normal principal:

01. El sistema mostrarà el menú principal a l'usuari.
02. L'usuari seleccionarà els paràmetres que li interessin.

3.4 Requisits no funcionals

3.4.1 Eficiència

Per tal de poder controlar l'aforament d'un local o establiment en temps real, el sistema ha de ser capaç de processar amb rapidesa cada fotograma de la càmera. A més, el nostre programa haurà de ser ràpid amb el càlcul i la detecció de persones, per així poder refrescar adequadament la pantalla i poder informar a l'usuari.

3.4.2 Accessibilitat

Per poder utilitzar l'aplicació no fa falta cap mena de registre, només una descàrrega, per tan és de fàcil accés.

Al presentar una interfície gràfica senzilla, amb explicacions de per què serveixen els diferents camps (a través de *títols*), permet ser utilitzada amb facilitat sense haver d'adquirir coneixement de la funcionalitat interna de l'aplicació.

3.4.3 Usabilitat

El disseny de l'aplicació és intuïtiu, gràcies a la distribució adequada dels botons i els camps de text, explicant quina funció presenta cada element. Per tant és una aplicació de fàcil ús, que pot ser utilitzada per qualsevol persona que ho necessiti, sense necessitat de tindre prèviament coneixements previs per la seva utilització.

3.4.4 Fiabilitat

La fiabilitat que ha de proporcionar l'aplicació ha de ser molt alta per tal que l'usuari no pugui produir un error. Per tal de reduir les possibilitats d'error sempre hi ha un paràmetre per defecte en tots els paràmetres que l'usuari pot modificar, a més, s'ha limitat l'entrada de valors per només acceptar valors correctes.

3.4.5 Escalabilitat

El disseny i la implementació de l'aplicació permeten que aquesta es pugui executar en diferents sistemes operatius. Gràcies als llenguatges utilitzats, Python i C++, no hi haurà problemes de compatibilitat entre plataformes.

3.4.6 Claredat del text

El disseny de la interfície gràfica de l'aplicació s'ha desenvolupat amb unes mides de text i unes fonts adequades i fixes, a més, també s'hi han utilitzat colors de fons i de la lletra

que contrastin per la seva correcta lectura. S'han usat frases curtes i entenedors per facilitar a l'usuari entendre la funció de tots els elements que hi ha a la interfície gràfica.

4 Disseny

Es tracta d'una aplicació amb una finalitat molt específica, principalment, la d'informar a l'usuari, que pot ser per exemple un treballador d'una botiga o local de restauració. Aleshores s'ha optat per donar-li un disseny on la interfície gràfica fos senzilla de fer servir i sense complicacions, ja que el més rellevant pels usuaris serà veure amb claredat, senzillesa i eficàcia les dades del seu interès.

Per tal de desenvolupar aquesta aplicació es pretenia que fos de fàcil ús i es pogués utilitzar en diferents plataformes. Donant molta importància a una interfície d'usuari senzilla i entenedora, per poder mostrar el resultat i que l'usuari el pogués interpretar ràpidament.

En aquest apartat es justificarà l'ús de les diferents eines i tecnologies utilitzades per desenvolupar aquesta aplicació.

4.1 Arquitectura del programa

4.1.1 Llenguatge

Com que no hi ha cap especificació del llenguatge a utilitzar, primer es va estudiar quin seria el més adient. Com que l'aplicació que es pretenia fer anàlisi de vídeo per poder detectar i contar persones, es va buscar quins eren els llenguatges més comuns en l'anàlisi de vídeo i imatge. De la cerca es va veure que els llenguatges més usats eren el Python i el C++.

La gran utilització d'aquests dos llenguatges és pel fet que els dos són compatibles amb la llibreria d'anàlisi d'imatges OpenCV, una de les més populars i conegudes del moment.

Com que es tracta d'un treball compartit, es va triar repartir els dos llenguatges més utilitzats, de manera que el meu company Albert Cañellas implementaria l'aplicació amb Python i aquest TFG es programaria amb C++. D'aquesta manera es pretén poder comparar les dues aplicacions i observar les diferències i similituds que hi ha entre elles, com el rendiment i l'eficàcia a l'hora de fer anàlisi d'imatges. A més, també es pretén comparar la facilitat d'ús de cadascun dels llenguatges a l'hora de programar una aplicació amb interfície gràfica.

El llenguatge C++ és un llenguatge de programació que tenia com a objectiu estendre el llenguatge de programació C permetent la manipulació d'objectes. En aquest sentit el C++ és un llenguatge híbrid, **Figura 3** [12].



**Figura 3 Logotip
C++**

4.1.2 Altres tecnologies

Microsoft Visual Studio

Per implementar la funcionalitat de l'aplicació es va utilitzar l'entorn de desenvolupament de Visual Studio utilitzant el llenguatge de programació C++.

Un altre motiu pels quals s'ha escollit aquest entorn és perquè ja l'havia utilitzat anteriorment i dona suport per a desenvolupar l'aplicació.

Microsoft Visual Studio és un entorn de desenvolupament integrat per Windows i macOS. És compatible amb múltiples llenguatges de programació, com C++, C#, Java o Python entre altres. Aquest entorn de desenvolupament permet als desenvolupadors crear aplicacions, llocs i aplicacions web, així com serveis web en qualsevol entorn que suporti la plataforma, **Figura 4**.

També inclou un editor de codi que admet IntelliSense, un component de completació de codi, així com la refactorització de codi. Altres eines que té integrat inclouen un perfilador de codi, dissenyador per la creació d'aplicacions GUI, dissenyador web, dissenyador de classes i dissenyador d'esquemes de base de dades [13].

A més, permet un desenvolupament amb el conjunt d'eines completes, des del disseny inicial, fins a la implementació final, incloent-hi:

- Millores en el rendiment d'IntelliSense per als arxius de C++.
- Desenvolupament local amb gran nombre d'emuladors.
- Accés simplificat a les proves d'explorador de solucions.
- L'experiència de primera classe de Git, per crear i clonar repositoris, administrar branques i resoldre conflictes de funció mitjançant combinació.



Figura 4 Logotip de Visual Studio

OpenCV

Llibreria OpenCV, es tracta d'una llibreria de codi obert per al processament d'imatges i visió per computador, desenvolupada originalment per Intel. És gratuït per ús comercial i no comercial, i es tracta d'una llibreria destinada principalment al processament en temps real. El gran nombre de funcions incorporades que implementen el processament d'imatges i algorismes de visió per computadora que fan que el desenvolupament d'aplicacions de visió per computador siguin fàcils i eficients, **Figura 5** [14].

Algunes de les seves característiques clau són:

- Optimitzat pel processament d'imatges en temps real i aplicacions de visió per computador.
- La interfície principal d'OpenCV està en C++
- També hi ha interfícies completes en altres idiomes com C, Python i Java
- Les aplicacions d'OpenCV s'executen en Windows, Android, Linux, Mac i iOS



Figura 5 Logotip d'OpenCV

Cvui

Sovint, el desenvolupament d'un projecte de visió per computador implica ajustar paràmetres d'una tècnica per aconseguir el resultat desitjat. Aquests paràmetres poden ser els llindars d'un algorisme de detecció de vores o la brillantor d'una imatge, per exemple. Si no utilitzeu cap interfície gràfica d'usuari (GUI) per modificar aquests paràmetres, heu d'aturar l'aplicació, ajustar el codi, executar l'aplicació de nou, avaluar-la i repetir-la fins que sigui bona. Això és tediós i consumeix molt de temps, **Figura 6**.

Aquest és el propòsit de cvui. És una llibreta d'interfície d'usuari C ++, només de capçalera i multiplataforma (Windows, Linux i OSX), construïda a sobre de les primitives de dibuix d'OpenCV. No té cap altra dependència que l'OpenCV, que en aquest cas ja estem utilitzant.

Com a resultat, la llibreria té una API amigable i semblant a C, sense classes / objectes i diversos components, per exemple, barra de seguiment, botó, text, entre d'altres.

Per utilitzar cvui a l'aplicació només cal incloure cvui.h al projecte i crear una imatge per representar els diferents component que interessin [15].

Algunes de les característiques de cvui són [16]:

- Interfície d'usuari lleugera i senzilla d'utilitzar;
- Només capçalera sense dependències externes (excepte OpenCV);
- Basat només en primitives de dibuix OpenCV (no calen OpenGL o Qt);
- API amigable i semblant a C (sense classes / objectes, etc.);
- Representeu fàcilment components sense preocupar-vos de la seva posició (mitjançant files / columnes);
- API de ratolí senzilla (però potent);
- Nombre modest de components de la IU (11 en total);
- Disponible en C ++ i Python (implementació pura, sense vinculacions).

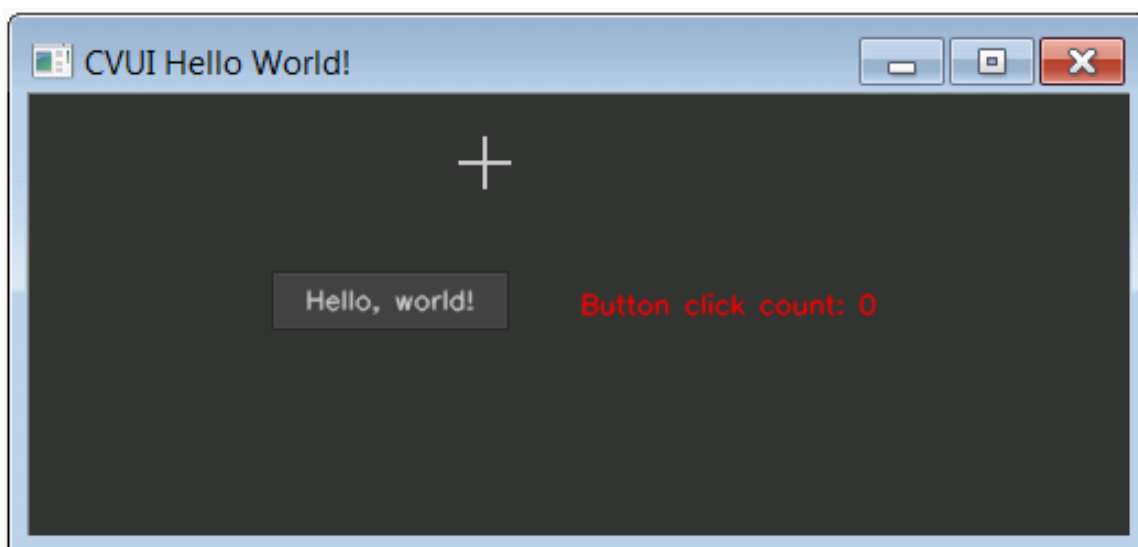


Figura 6 Aplicació bàsica de cvui amb un botó i un text

Dispositius externs

A conseqüència de l'objectiu principal de l'aplicació, controlar l'aforament d'un local a partir d'una imatge en directe, és necessària una càmera per poder executar l'aplicació. Algunes de les càmeres que es poden utilitzar poden estar connectades per wifi, per cable o poden venir ja incorporades a l'ordinador, com seria el cas de les webcams que porten els ordinadors portàtils.

Càmeres

Les característiques recomanades per tal que la càmera funcioni correctament amb el programa són: una resolució mínima de 720P i uns 30 fotogrames per segon mínim.

A continuació es descriuran tres càmeres de diferent connexió a l'ordinador que serien vàlides per usar l'aplicació.

Càmera de Wansview de vigilància exterior. Es tracta d'una càmera de vídeo amb connexió mitjanant wifi. Aquest tipus de connexió es realitza a través de la Ip de la càmera, **Figura 7** [17]. Característiques:

- Lent HD de 2 MegaPixels.
- Resolució de fins a 1080P.
- Connexió wifi estable de 2.4G.
- Protecció IP66 resistent a l'aigua, als cops i a temperatures d'entre -10 i 40 °C.
- Visió nocturna.
- Àudio bidireccional.



Figura 7 Wansview Cámara Vigilancia

Càmera Nikon D3020. Es tracta d'una càmera de fotografia connectada a una capturadora de vídeo mitjançant un cable HDMI. Per dur a terme aquesta connexió seria necessari tant el cable HDMI compatible amb la càmera com la capturadora de vídeo, **Figura 8** [18]. Característiques:

- Sensor CMOS de format DX de 24.2 MegaPixels.
- Resolució de fins a 1080P.
- Sortida mini HDMI.
- Micròfon ajustable.
- Adaptador per incorporar un trípode.



Figura 8 Càmera Nikon D3020

Càmera Logitech Brio Stream. Càmera webcam amb connexió USB, mitjançant un cable USB-C a USB-A [19], Per connectar aquest tipus de càmera solament seria necessari tenir el cable USB compatible amb la càmera, **Figura 9**. Característiques:

- Resolució variable de 720P fins a 4K
- Tecnologia RightLight 3 amb HDR
- Micròfon unidireccional
- Enfoc dinàmic



Figura 9 Càmera Logitech Brio Stream

Cables

Algunes de les càmeres es poden connectar a l'ordinador a través d'un cable USB, per tal que la imatge que captura la càmera es vegi en directe a la pantalla de l'ordinador, i en el nostre cas a través de l'aplicació que hem dissenyat **Figura 10** [20].



Figura 10 Cable USB tipo c a USB tipo A

Un altre mètode per capturar la imatge d'una càmera és a través d'un cable HDMI o mini HDMI segons l'entrada de la càmera **Figura 11** [21]. Per altra banda, pot ser que utilitzant aquest cable no sigui suficient i que per tant sigui necessària també una capturadora d'imatge **Figura 12** [22].



Figura 11 Cable Mini HDMI a HDMI



Figura 12 Capturadora de vídeo HDMI

4.2 Disseny de la interfície gràfica

4.2.1 Pantalles

LA interfície gràfica de l'aplicació es va dissenyar inicialment per ser intuïtiva, i fàcil de fer servir. Amb pocs paràmetres per tal que l'usuari pogués iniciar la funcionalitat del programa ràpidament.

EL disseny de la GUI es va realitzar després d'haver treballat amb el programa de detecció de persones, per la qual cosa es va dissenyar tenint en compte aquells paràmetres que a l'usuari li podria ser útil modificar. En aquest cas es va optar per fer una pantalla principal on afegir els paràmetres d'entrada del local, per si l'entrada era en horitzontal o vertical a la visualització, i el nombre de persones que l'usuari vol permetre que entrin dins de l'establiment o local. A part es va decidir fer una segona pantalla on es veuria en directa la imatge de la càmera i s'aniria mostrant el nombre de persones que entraven i sortien del local així com el número de persones que encara poden entrar al local sense superar l'aforament permès.

Primerament es va fer un esquema amb PowerPoint per organitzar els diferents elements de la pantalla

4.2.1.1 Pantalla principal

El disseny de la pantalla principal pretén ser molt minimalista i fàcilment entenedor. Com ja s'ha dit en el punt anterior primer es va fer un disseny amb PowerPoint **Figura 13**.



Figura 13 Esbós de la pantalla principal de la GUI

4.2.1.2 Pantalla de càmera

Com s'ha mencionat anteriorment, es va optar per dissenyar una pantalla diferent a l'hora de mostrar la imatge de la càmera en directe. Per fer-ho i representa un esquema de com serà la pantalla final s'ha fet un esbós inicial **Figura 14**.



Figura 14 Esbós de la pantalla de la càmera de la GUI

4.3 Disseny de l'estructura de l'aplicació

En aquest apartat es detallarà l'estructura de fitxers de l'aplicació

- PeopleCounter
 - > References
 - > External Dependencies
 - > Header Files
 - Cvui.h
 - > Resource Files
 - > Source Files
 - Source.cpp

5 Implementació

Per implementar la funcionalitat de l'aplicació, com ja s'ha mencionat a l'apartat **3. Disseny**, s'ha utilitzat l'entorn de desenvolupament Visual Studio i el llenguatge de programació C++. En tractar-se d'una aplicació per detectar persones que entren o surten d'un local s'assumeix que la càmera estarà estàtica. Aleshores es va optar per desenvolupar un algoritme que compara el primer frame detectat amb el segon i n'observa les diferències. En disposar d'una càmera estàtica col·locada a l'entrada d'un establiment les diferències principals que s'han d'observar són les persones que n'entren i en surten.

5.1 Interfície gràfica

Les pantalles de la interfície gràfica s'han detallat a l'apartat **4.2 Disseny de la interfície gràfica**. El disseny es va ser amb les següents estructures de codi.

5.1.1 Pantalla principal

A l'hora de desenvolupar la pantalla principal de la GUI de l'aplicació es va seguir l'esquema que s'havia plantejat i es va desenvolupar tant similar com es va poder segons les opcions que ens permetia la llibreria utilitzada, en aquest cas la llibreria cvui.

En el resultat final de la GUI que es mostra a la Figura 13, es poden apreciar algunes diferències:

- L'idioma de la GUI s'ha decidit fer amb anglès, ja que és un idioma internacional.
- S'ha decidit posar una explicació prèvia a tots els botons per facilitar l'ús de l'aplicació a l'usuari

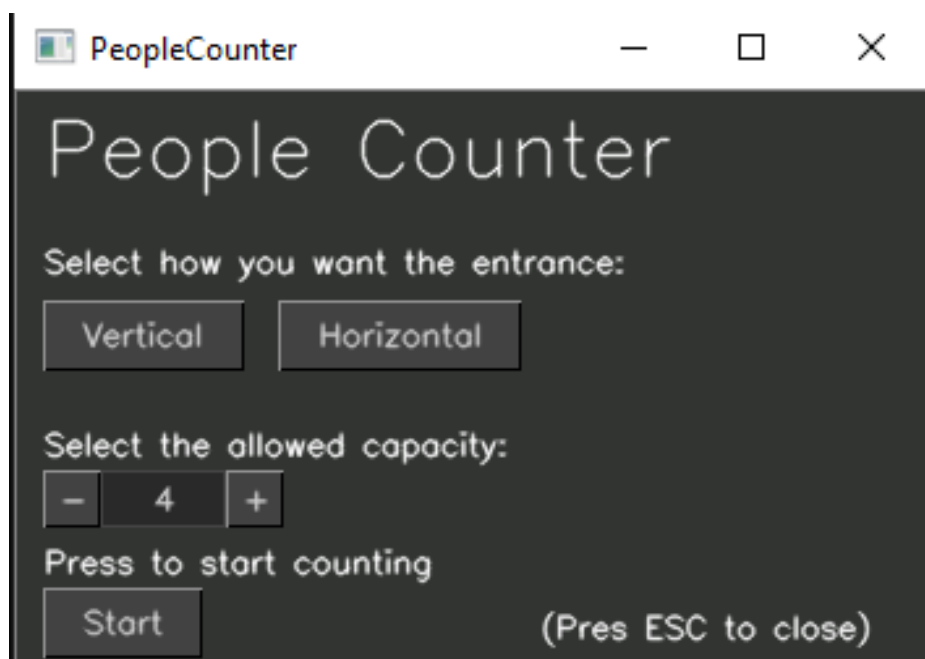


Figura 15 Pantalla principal de la GUI

Tal i com s'indica en el punt 4.1.2 Altres tecnologies, per tal de poder utilitzar la llibreria cal incloure el fitxer `cvui.h` de la següent manera:

```
#define CVUI_IMPLEMENTATION
#include "cvui.h"
```

Codi 1 Include del fitxer `cvui.h`

Primerament per tal de crear aquesta pantalla cal definir una finestra, en aquest cas l'hem anomenat "PeopleCounter".

```
#define WINDOW_NAME "PeopleCounter"
```

Codi 2 Definició de la finestra "PeopleCounter"

Un cop inclosa la llibreria cal inicialitzar-la, per fer-ho creem una finestra.

```
cv::namedWindow(WINDOW_NAME);
cvui::init(WINDOW_NAME);
```

Codi 3 Inicialització i creació de la finestra "PeopleCounter"

Per afegir elements, creem un frame, amb les mides que volguem i comencem a afegir els elements que ens interessin.

```
cv::Mat frame = cv::Mat(300, 500, CV_8UC3);
while (true) {

    frame = cv::Scalar(49, 52, 49);

    cvui::text(frame, 10, 60, "Select how you want the entrance: ", 0.4,
0xffffffff);

    if (cvui::button(frame, 10, 80, "Vertical") mode = 1;

}
```

Codi 4 Exemple de com afegir elements al frame

5.1.2 Pantalla de càmera

A l'hora de desenvolupar la pantalla de la càmera de la GUI de l'aplicació es va seguir l'esquema que s'havia plantejat, igual que es va fer amb la pantalla principal, i es va desenvolupar tan similar com es va poder segons les opcions que ens permetia OpenCV.

En el resultat final de la GUI que es mostra a la Figura 15, es poden apreciar algunes diferències:

- L'idioma de la GUI s'ha decidit fer amb anglès, ja que és un idioma internacional.
- S'ha indicat el nombre de persones que entren a la dreta i les que surten a l'esquerra, representant que les línies vermelles estan situades a la porta del local. A la part dreta de la càmera es representa dins del local i a la part esquerra de la càmera es representa la part exterior del local.

La implementació d'aquesta pantalla es realitza en la funció **Model1** o **Model2**, segons s'hagi escollit una visualització en horitzontal o en vertical. En el cas de la **Figura 15**, es veu una entrada vertical, per la qual cosa es faria servir la funció **Model1**. En aquesta funció s'afegeixen les línies vermelles que divideixen la imatge i els diferents quadres de text damunt de la imatge, en format **IplImage**. En el cas del control d'aforament es faria de la següent manera:

```
cvPutText(image, ("People that can get in: " +
NumberToString(total_in)).c_str(), cvPoint(0, 90), &font, textColor);
```

Codi 5 Exemple d'escriure informació sobre una **IplImage**

En aquest exemple imatge és el frame captat per la càmera, que després en la funció **Start** explicada en el punt **5.2.2 Start** farà que es mostri per pantalla. Primer, però, s'ha de canviar el format de la imatge de **IplImage** a **Matriu**:

```
cv::Mat m = cv::cvarrToMat(rawImage);
cv::imshow(WINDOW_NAME, m);
```

Codi 6 Codi per canviar el format de la imatge de **IplImage** a **Matriu**

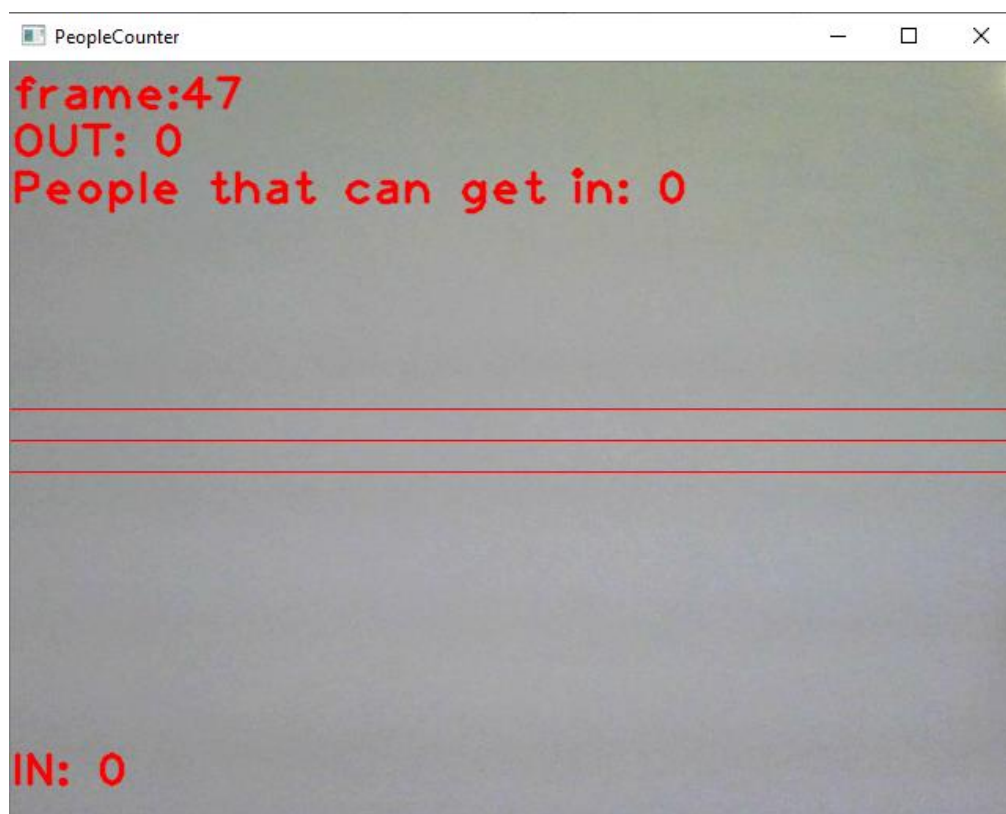


Figura 17 Pantalla de la càmera de la GUI, amb entrada horitzontal al local

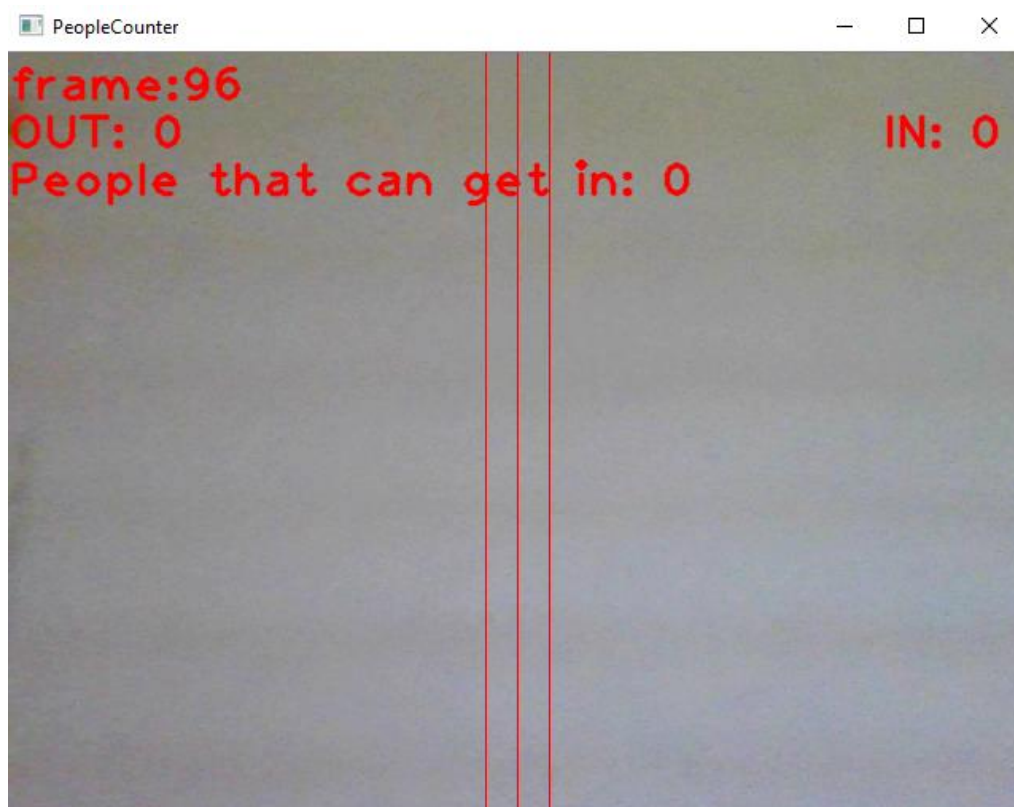


Figura 16 Pantalla de la càmera de la GUI, amb entrada vertical al local

5.2 Funcionalitat de l'aplicació

Probablement la part més important d'aquest treball és el correcte funcionament de l'aplicació. Per tal de garantir-ho s'ha organitzat l'estructura de l'algoritme en diferents funcions, que són:

- Main
- Start
- Update_mhi
- InitImages
- Silhoutte_reduction
- Model
- Mode2
- Reject_small_and_large_object
- Detect_object

5.2.1 Main

En la funció main, és on està definida la pantalla principal de la interfície gràfica. Aquesta funció és l'encarregada de mostrar tots els elements i botons de la GUI i llegir la informació que introdueixi l'usuari. Amb la informació que captí dels usuaris procedirà a inicialitzar la càmera si s'ha premut el botó START o per altra banda, tancar el programa si l'usuari ha premut la tecla ESC. Algunes de les estructures presents en el main s'han explicat en l'apartat **5.1.1 Pantalla principal**.

L'estructura principal de la funció main, segueix la següent estructura:

```
int main(int argc, int* argv[]) {
    ...

    cv::namedWindow(WINDOW_NAME);
    cvui::init(WINDOW_NAME);

    while (true) {
        ...

        frame = cv::Scalar(49, 52, 49);

        cvui::text(frame, 10, 10, "People Counter", 1, 0xffffffff);

        cvui::text(frame, 10, 60, "Select how you want the entrance:
", 0.4, 0xffffffff);

        ...

        if (cvui::button(frame, 10, 190, "Start")) {
            start();
        }

        cvui::update();

        cv::imshow(WINDOW_NAME, frame);
        cv::waitKey(30);
    }
}
```

```

        if (cv::waitKey(20) == 27) {
            break;
        }
    }
    return 0;
}

```

Codi 7 Estructura principal de la funció main

Tal com es pot veure en aquest fragment de codi, si l'usuari prem el botó Start, aquest crida la funció Start que explicarem en el punt **5.2.2 Start**. També es pot observar que si es prem la tecla “27” es fa un “break” i per tant es tanca el programa. La tecla “27” correspon a la tecla ESC. És per això que en la **Figura 14** s'indica que per tancar cal prémer la tecla ESC.

5.2.2 Start

La funció Start és la funció principal on hi ha un bucle principal que anirà llegint frames de la càmera periòdicament. Si es tracta del primer frame que es captura, aquest es tractarà d'una manera diferent, ja que a partir d'aquest primer frame s'inicialitzen alguns valors, com la mida del frame, el centre de la imatge i els valors del threshold, el nombre de persones que hi ha dins del local.

Un cop inicialitzades les variables tots els frames s'analitzen:

- Es canvia el format del frame de BGR a HSV.
- Es detecta i elimina el soroll de fons si es necessari
- Es detecten els objectes
- Finalment es transforma la imatge a Matriu i es mostra per pantalla

```

while (1)
{
    if (!pause) {
        // get a frame
        rawImage = cvQueryFrame(capture);
        if (!rawImage) break;
        ++nframes;
    }

    if (nframes == 1) {
        CvSize sz = cvGetSize(rawImage);
        InitImages(sz);
        ...
    }

    if (rawImage)
    {
        cvCvtColor(rawImage, hsvImage, CV_BGR2HSV);
        silhoutte_reduction(hsvImage, hsvImage, CV_BGR2HSV);
        ...
    }
}

```

```

tempImage);
    if (temp_image)    cvShowImage("Process Window",
    else cvDestroyWindow("Process Window");

    if (mode == 1)    mode1(rawImage);    //VERTICAL MODE
    if (mode == 2)    mode2(rawImage);    //HORIZONTAL MODE

    cv::Mat m = cv::cvarrToMat(rawImage);

    cv::imshow(WINDOW_NAME, m);
}

// User input:
int key = cvWaitKey(10);
if (key == 27)    break;
}

```

Codi 8 Fragment de codi de la funció start

5.2.3 Update_mhi

Aquesta funció realitza diverses modificacions en el frame captat per la càmera.

Modificacions del frame:

- Es converteix la imatge a escala de grisos
- Es busca la diferència entre el frame actual i l'anterior
- S'aplica un threshold
- Es calcula l'orientació del gradient de moviment
- Iterar sobre els components de moviment
- Dibuixa els centres i els rectangles de l'objecte detectat

També, segons l'entrada del local s'ha seleccionat es controlarà l'aforament del local d'una manera o d'una altre.

5.2.4 InitImages

En la funció InitImages s'inicialitzen les imatges de tipus IplImage* que seran necessàries durant l'execució per detectar objectes i rastrejar-los. Aquestes imatges s'inicialitzen com una imatge de 8 bits i un canal amb la mateixa mida que el frame que es vol analitzar. Un exemple d'inicialització seria el següent:

```
ImaskCodeBook = cvCreateImage(cvGetSize(rawImage), IPL_DEPTH_8U, 1);
```

Codi 9 Inicialització d'una imatge

5.2.5 *Silhouette_reduction*

En aquesta funció s'apliquen diversos thresholds a les imatges. Abans però, es separen els canals d'una matriu multicanal en múltiples matrius de canal únic, i un cop aplicats els thresholds es combinen les matrius per formar una única matriu multicanal.

```
static IplImage* h = cvCreateImage(cvGetSize(src), 8, 1);
static IplImage* s = cvCreateImage(cvGetSize(src), 8, 1);
static IplImage* v = cvCreateImage(cvGetSize(src), 8, 1);
cvSplit(src, h, s, v, 0);

cvThreshold(h, h, h_min, h_max, CV_THRESH_TOZERO);
cvThreshold(h, h, h_max, h_max, CV_THRESH_TOZERO_INV);
cvThreshold(s, s, s_min, s_max, CV_THRESH_TOZERO);
cvThreshold(s, s, s_max, s_max, CV_THRESH_TOZERO_INV);
cvThreshold(v, v, v_min, v_max, CV_THRESH_TOZERO);
cvThreshold(v, v, v_max, v_max, CV_THRESH_TOZERO_INV);

cvMerge(h, s, v, 0, dst);
```

Codi 10 Codi de la funció *Silhouette_reduction*

5.2.6 *Model1*

Aquesta funció és l'encarregada de dibuixar damunt del frame de la càmera, tots els elements que després es mostraran per pantalla, en aquest cas:

- El frame en el que estem
- El nombre de persones que entren al local
- El nombre de persones que surten del local
- El nombre de persones que encara poden entrar al local sense superar l'aforament
- Línies verticals que indiquen on està la porta del local.

La informació es detallada en el punt **5.1.2 Pantalla de càmera**.

5.2.7 *Model2*

Aquesta funció és l'encarregada de dibuixar damunt del frame de la càmera, tots els elements que després es mostraran per pantalla, en aquest cas:

- El frame en el que estem
- El nombre de persones que entren al local
- El nombre de persones que surten del local
- El nombre de persones que encara poden entrar al local sense superar l'aforament
- Línies horitzontals que indiquen on està la porta del local.

La informació es detalla en el punt **5.1.2 Pantalla de càmera**.

(La diferència entre aquesta funció i la funció **5.2.x Model1** és la disposició horitzontal o vertical de l'entrada del local.)

5.2.8 Reject_small_and_large_object

Aquesta funció el que fa és eliminar els diferents sorolls de fons que hi pot haver a la imatge i eliminar el contorn i punts no redundants per tal d'estalviar espai de memòria.

També es mira la mida de l'objecte detectat. En el cas que no estigui dins dels valors determinats s'ignora l'objecte, i en el cas que estigui dins dels valors determinats es dibuixarà el contorn de l'objecte. A continuació es detalla aquest fragment de codi:

```
while ((c = cvFindNextContour(scanner)) != NULL) {
    double c_len = cvContourPerimeter(c);
    double c_area = cvContourArea(c);
    if (c_len < (double)min_len || c_area < (double)min_area
        || c_len > (double)max_len || c_area > (double)max_area) {
        continue;
    }
    else {
        cvDrawContours(dst, c, cvScalarAll(255), cvScalarAll(0),
0, -1, 8, cvPoint(0, 0));
        CvMoments moments;
        CvRect rect = cvBoundingRect(c, 0);
        cvSetImageROI(dst, rect);
        cvMoments(dst, &moments, 1);
        double M00 = cvGetSpatialMoment(&moments, 0, 0);
        double M01 = cvGetSpatialMoment(&moments, 0, 1);
        double M10 = cvGetSpatialMoment(&moments, 1, 0);
        center[numContour].x = (int)(M10 / M00);
        center[numContour].y = (int)(M01 / M00);
        cvResetImageROI(dst);
        center[numContour] = cvPoint((rect.x +
center[numContour].x), (rect.y + center[numContour].y));

        cvDrawCircle(circleImage, center[numContour], 20,
cvScalarAll(255), -1, 8, 0);
        cvDrawCircle(rawImage, center[numContour], 10,
cvScalar(255, 0, 0), -1, 8, 0);
        cvRectangleR(rawImage, rect, cvScalar(255, 255, 255), 1,
8, 0);

        cvPutText(rawImage,
(NumberToString(center[numContour].x) + ", " +
NumberToString(center[numContour].y)).c_str(), center[numContour], &font,
CV_RGB(255, 0, 0));

        numContour++;
        if (numContour > numOfCenters) {
            numContour = numOfCenters;
            break;
        }
    }
}
```

Codi 11 Codi de la funció *Reject_small_and_large_object*

5.2.9 Detect_object

La funció Detect_object recorre tots els píxels de la fila o columna que s'estigui analitzant. Si es detecta un objecte, se'n guarden les seves coordenades. El codi per recórrer tots els píxels és el següent:

```
for (int i = 0; i < num_of_pixel_on_line; i++) {
    if (track_line[i].value) {
        track_line[i].time_count += 1;
        if (track_line[i].time_count >
            track_line[i].exist_time) {
            track_line[i].time_count = 0;
            track_line[i].value = false;
        }
    }
}
```

Codi 12 Codi per recorre els píxels d'una fila d'una imatge

Finalment si s'ha detectat un objecte es determina la seva direcció, es retornarà 1 si l'objecte entra al local i es retornarà 2 si l'objecte surt del local. En aquest cas particular amb objecte ens referim a persona. Part del codi de determinar la direcció s'especifica a continuació

```
///determine the direction of movement
if (_isDetected) {
    if (direction_point - detectable_direction > 0) { return 1; }
    //input
    else { return 2; } //output
}
}
```

Codi 13 Codi per determinar la direcció de moviment d'un objecte

6 Avaluació

6.1 Joc de proves

A continuació es detallen en una taula les diverses proves que s'han realitzat per comprovar el correcte funcionament de l'algorisme de l'aplicació i de la interfície gràfica.

Prova	Resultat esperat	Resultat
Iniciar el programa.	A l'iniciar el programa s'obre directament la GUI.	Correcte
Es prem la tecla "ESC" quan estem a la pantalla principal.	Es tanca el programa.	Correcte
Es prem la tecla "ESC" quan estem a la pantalla de càmera.	Es tanca la pantalla de càmera i torna a la pantalla principal.	Correcte
No es selecciona cap entrada i es prem el botó START.	S'inicialitza la pantalla de càmera amb l'entrada per defecte.	Correcte
Selecció entrada Horitzontal i es prem el botó START.	S'inicialitza la pantalla de càmera amb la configuració d'entrada horitzontal.	Correcte
Selecció entrada Vertical i es prem el botó START.	S'inicialitza la pantalla de càmera amb la configuració d'entrada vertical.	Correcte
No es modifica l'aforament i es prem el botó START.	S'inicialitza la pantalla de càmera amb l'aforament per defecte.	Correcte
S'incrementa l'aforament i es prem el botó START.	S'inicialitza la pantalla de càmera i es veu com l'aforament màxim és el que ha introduït l'usuari.	Correcte
Es disminueix l'aforament, mantenint un valor positiu i es prem el botó START.	S'inicialitza la pantalla de càmera i es veu com l'aforament màxim és el que ha introduït l'usuari.	Correcte
Es disminueix l'aforament, introduint un valor negatiu i es prem el botó START.	S'inicialitza la pantalla de càmera i es veu com l'aforament màxim ja s'ha assolit.	Correcte

Es modifica l'aforament a mida que entren i surten persones de l'establiment.	Es veu com el controlador d'aforament es va modificant.	Correcte
Es detecta una sola persona quan "entren" a l'establiment en entrada horitzontal.	Es veu com el comptador de persones que entren augmenta.	Correcte
Es detecta una sola persona quan "surten" de l'establiment en entrada horitzontal.	Es veu com el comptador de persones que surten augmenta.	Correcte
Es detecta una sola persona quan "entren" a l'establiment en entrada vertical.	Es veu com el comptador de persones que entren augmenta.	Correcte
Es detecta una sola persona quan "surten" de l'establiment en entrada vertical.	Es veu com el comptador de persones que surten augmenta.	Correcte
Es detecten persones quan "entren" a l'establiment en entrada horitzontal.	Es veu com el comptador de persones que entren augmenta.	Correcte
Es detecten persones quan "surten" a l'establiment en entrada horitzontal.	Es veu com el comptador de persones que surten augmenta.	Correcte
Es detecten persones quan "entren" a l'establiment en entrada vertical.	Es veu com el comptador de persones que entren augmenta.	Correcte
Es detecten persones quan "surten" a l'establiment en entrada vertical.	Es veu com el comptador de persones que surten augmenta.	Correcte

Taula 1 Joc de proves

6.2 Joc de proves amb fotografies

A continuació es mostren algunes captures de pantalla del joc de proves per tal de demostrar amb imatges el correcte funcionament.

6.2.1 Detecció d'una sola persona entrant en una entrada horitzontal

En la **Figura 18** es pot observar com la persona és detectada i encara poden entrar 4 persones dins del local. En la **Figura 19** es pot veure la persona just després de passar per l'entrada del local i com ara ja només poden entrar 3 persones.



Figura 18 Detecció d'una sola persona abans d'entrar en una entrada horitzontal.

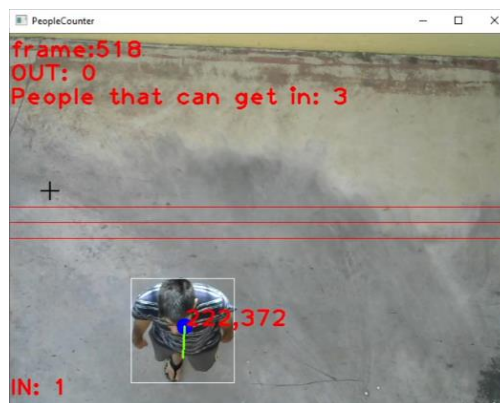


Figura 19 Detecció d'una sola persona després d'entrar en una entrada horitzontal.

6.2.2 Detecció d'una sola persona sortint en una entrada horitzontal

En la **Figura 20** es pot observar com la persona és detectada sortint del local, i en la **Figura 21** es pot veure la persona just després de passar per l'entrada del local i com ara ja tornen a poder poden entrar 4 persones al local.

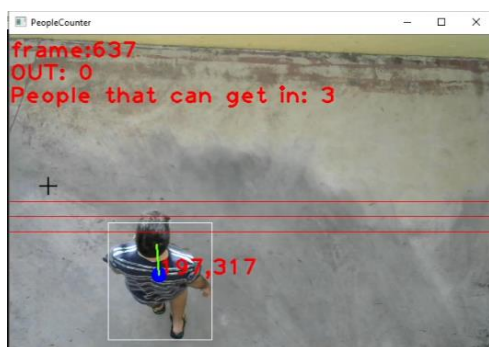


Figura 20 Detecció d'una sola persona abans de sortir en una entrada horitzontal.

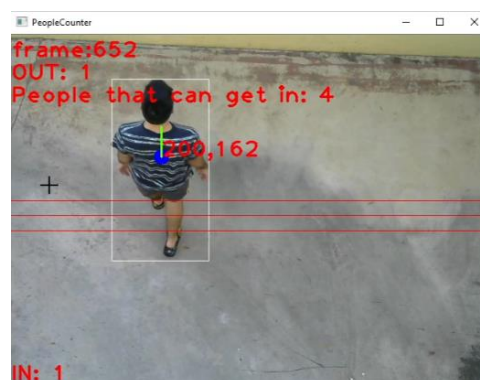


Figura 21 Detecció d'una sola persona després de sortir en una entrada horitzontal.

6.2.3 Detecció de més d'una persona en una entrada horitzontal

En la **Figura 22** es pot observar com es detecten dues persones alhora, una que ja ha entrat al local i l'altre que ho farà. Per altre banda, a la **Figura 23** es pot veure com es detecten dues persones sortint del local casi al mateix temps.

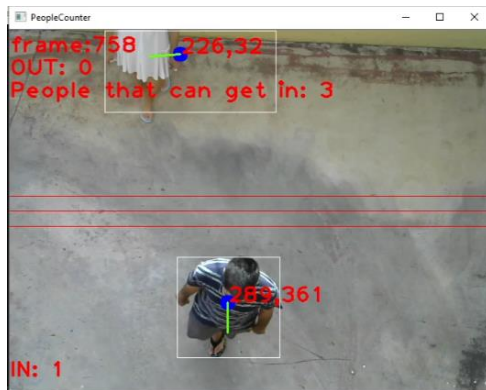


Figura 22 Detecció de dues persones entrant en una entrada horitzontal.



Figura 23 Detecció de dues persones sortint en una entrada horitzontal.

6.2.4 Detecció d'aforament complet en una entrada horitzontal

En la **Figura 24** es pot observar com la persona és detectada i encara pot entrar 1 persona dins del local. A la **Figura 25**, quan la persona ja ha entrat al local es pot veure com s'informa de l'aforament complet.

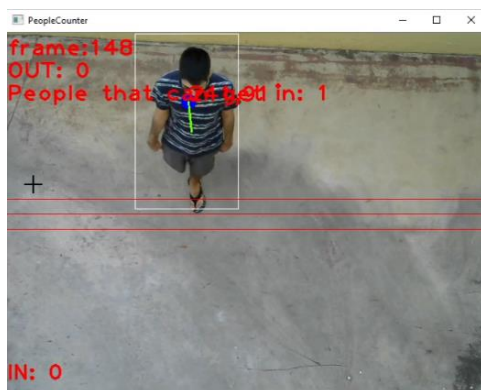


Figura 24 Detecció d'una sola persona abans d'entrar en una entrada horitzontal quan només pot entrar una persona més.



Figura 25 Detecció d'una sola persona després d'entrar i completar l'aforament del local en una entrada horitzontal.

6.2.5 Detecció d'una sola persona entrant en una entrada vertical

En la **Figura 26** es pot observar com la persona és detectada i encara poden entrar 4 persones dins del local. En la **Figura 27** es pot veure la persona just després de passar per l'entrada del local i com ara ja només poden entrar 3 persones.



Figura 26 Detecció d'una sola persona abans d'entrar en una entrada vertical.



Figura 27 Detecció d'una sola persona després d'entrar en una entrada vertical.

6.2.6 Detecció d'una sola persona sortint en una entrada vertical

En la **Figura 28** es pot observar com la persona és detectada sortint del local, i en la **Figura 29** es pot veure la persona just després de passar per l'entrada del local i com ara ja tornen a poder poden entrar 4 persones al local.

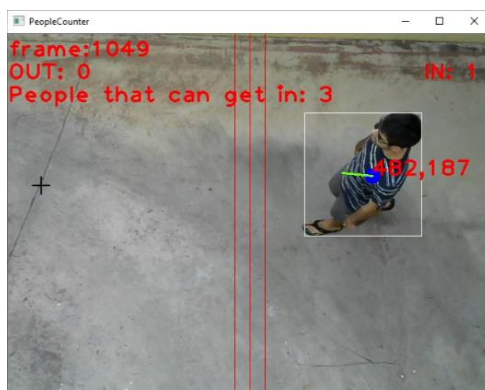


Figura 28 Detecció d'una sola persona abans de sortir en una entrada vertical.

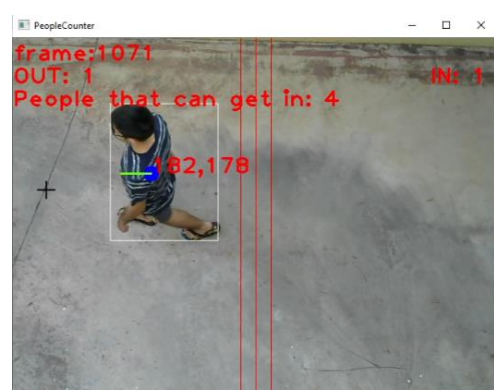


Figura 29 Detecció d'una sola persona després de sortir en una entrada vertical.

6.2.7 *Detecció de més d'una persona entrant en una entrada vertical*

En la **Figura 30** es pot observar com es detecten dues persones alhora, una que ja ha entrat al local i l'altre que ho farà. Per altre banda, a la **Figura 31** es pot veure com es detecten dues persones sortint del local.



Figura 30 Detecció de dues persones entrant en una entrada vertical.



Figura 31 Detecció de dues persones sortint en una entrada vertical.

6.2.8 *Detecció d'aforament complet en una entrada vertical*

En la **Figura 32** es pot observar com la persona és detectada i encara pot entrar 1 persona dins del local. A la **Figura 33**, quan la persona ja ha entrat al local es pot veure com s'informa de l'aforament complet.

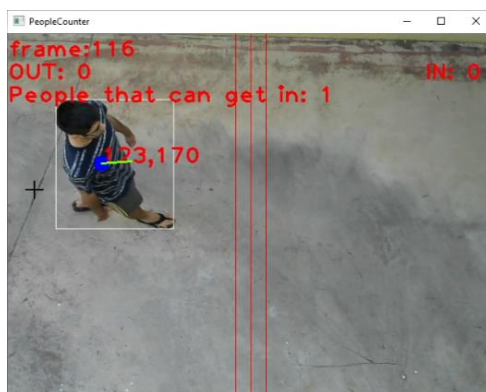


Figura 32 Detecció d'una sola persona abans d'entrar en una entrada vertical quan només pot entrar una persona més.



Figura 33 Detecció d'una sola persona després d'entrar i completar l'aforament del local en una entrada vertical.

7 Conclusions

7.1 Conclusions del funcionament de l'aplicació

A l'hora de realitzar les proves del funcionament de l'aplicació s'ha vist que s'han pogut avaluar tots els punts que hi havia previstos amb èxit. S'ha pogut avaluar com es detectaven correctament les persones i com es comptava que entraven o sortien del local en passar per la línia central de la pantalla, on teòricament se situaria l'entrada del local.

En fer el treball conjuntament amb el meu company Albert Cañellas, hem pogut comparar els resultats obtinguts qualitativament. S'ha pogut observar que el programa realitzat amb Python i per tant fent servir Xarxes Neuronals era més acurat a l'hora de detectar persones i en canvi el programa realitzat amb C++ a vegades detectava objectes, com plantes, ombres i cantonades, tal com es pot apreciar en la **Figura 23** i la **Figura 25**. Tot i això, com que els objectes que es detectaven eren estàtics no dificultaven el control de l'aforament. En els dos casos, però, si la persona que estava sent gravada per la càmera passava molt de pressa no es podia rastrejar correctament.

Tampoc ha sigut feina fàcil trobar una bona localització per executar el joc de proves, ja que es requeria una alçada considerable. Tant en el joc de proves que ha realitzat el meu company com el meu, s'ha dut a terme amb una alçada de la càmera de 4m aproximadament. El fet que es necessiti aquesta alçada és degut al fet que el programa està pensat per detectar persones des d'una vista aèria.

Si fem una comparació de l'experiència d'ús de l'aplicació per part d'un usuari, obtenim resultats similars. Vam demanar a diferents amics i familiars que fessin servir les dues aplicacions per veure si alguna d'elles els resultava més senzilla de fer servir que l'altre o si tenien problemes amb algun aspecte.

Pel que fa a la selecció dels diferents paràmetres es pot observar que el programa realitzat amb C++, és més senzill i no modifica paràmetres interns d'imatge i en canvi en el Python sí. És per això que si l'usuari té coneixements i vol poder modificar aquests paràmetres la GUI de Python serà més adient. Tot i això en els dos casos hi ha valors per defecte dels paràmetres pel que l'usuari no té perquè tocar res per començar a visualitzar la imatge de la càmera. Les dues aplicacions els van semblar fàcils de fer servir en aquest aspecte.

Si ens fixem en l'inici de la visualització de la càmera en directe, en les dues GUI es realitza de forma senzilla, l'única diferència que es pot veure és que en el cas de l'aplicació desenvolupada amb Python la imatge de la càmera apareix en la mateixa finestra i en canvi en l'altra aplicació, es tanca la finestra principal i se n'obre una altra amb la imatge capturada per la càmera. En cap de les dues van tenir dificultats pel que fa a l'inici de la càmera.

Un cop la càmera ja està encesa i controla l'aforament en el cas de l'aplicació en Python, es mostren el nombre de persones que entren i el que en surten en un quadre a la dreta de la imatge, tot i que sí que apareix una alerta un cop s'ha completat l'aforament. En l'aplicació amb C++, en canvi, es mostra aquesta informació al damunt de la imatge i a més, t'informa de la quantitat de persones que encara poden entrar al local sense completar l'aforament. En aquest cas sí que van coincidir bastant en el fet que la manera en què es mostra en l'aplicació desenvolupada en C++ era més intuïtiva i més fàcil d'entendre, ja que no requeria fer cap càlcul per saber quanta gent més pot entrar al local.

Finalment a l'hora de tancar l'aplicació, tot i que en pantalla principal de l'aplicació de C++, s'indiqui que s'ha de tancar amb la tecla "esc", això va portar algun entrebanc. Pel que podem concloure que en aquest aspecte la GUI desenvolupada amb Python era millor.

7.2 Conclusions personals

La realització d'aquest treball ens ha ajudat a aprofundir en la programació d'anàlisi d'imatge a temps real i d'intel·ligència artificial. Personalment m'ha semblat un projecte molt atractiu i interessant sobretot al poder-li donar una funcionalitat real en la situació social que estem vivint actualment. També m'ha permès utilitzar els coneixements adquirits durant les pràctiques externes realitzades a l'empresa TissUse GmbH, tot i que aplicades a un àmbit diferent.

Durant el desenvolupament del programa han sorgit algunes dificultats, com la poca documentació que hi havia respecte OpenCv en C++ respecte a la documentació que hi havia de Python. Tant pel que fa a la documentació oficial com els diferents fòrums que es poden trobar per internet.

Crec que l'aplicació té un bon disseny i això fa que sigui entenedora i fàcil de fer servir, tant per persones que tenen coneixements d'anàlisi d'imatges com persones que no. Permetent visualitzar de manera ràpida i concreta els resultats obtinguts de l'anàlisi d'imatge en directe.

En general, m'ha semblat un projecte molt enriquidor i una molt bona experiència dins de l'àmbit del desenvolupament de software.

8 Referències

- [1] ‘Pandemia de COVID-19 - Wikipedia, la enciclopedia libre’. [Online]. Available: https://es.wikipedia.org/wiki/Pandemia_de_COVID-19. [Accessed: 28-Jul-2021].
- [2] ‘¿Qué es la inteligencia artificial (IA)? | Oracle España’. [Online]. Available: <https://www.oracle.com/es/artificial-intelligence/what-is-ai/#ai-enterprise>. [Accessed: 28-Jul-2021].
- [3] ‘TissUse GmbH | Emulating Human Biology’. [Online]. Available: <https://www.tissuse.com/en/>. [Accessed: 01-Jul-2020].
- [4] ‘Interfaz gráfica de usuario o GUI: Qué es y Para qué sirve | Workana’. [Online]. Available: <https://i.workana.com/glosario/que-es-la-interfaz-grafica-de-usuario-gui/>. [Accessed: 15-Aug-2021].
- [5] ‘càmera | enciclopèdia.cat’. [Online]. Available: <https://www.enciclopedia.cat/ec-gdlc-e00024243.xml>. [Accessed: 15-Aug-2021].
- [6] ‘USB | Definición | Diccionario de la lengua española | RAE - ASALE’. [Online]. Available: <https://dle.rae.es/USB>. [Accessed: 15-Aug-2021].
- [7] ‘What is the function of the HDMI cable?’ [Online]. Available: <https://cie-group.com/how-to-av/videos-and-blogs/what-is-hdmi-high-definition-multimedia-interface>. [Accessed: 15-Aug-2021].
- [8] ‘Cómo utilizar los fotogramas en vídeo (frame rate) - CasanovaFotoBlog’. [Online]. Available: <https://www.casanovafoto.com/blog/2014/10/frame-rate/>. [Accessed: 17-Aug-2021].
- [9] ‘Model de color RGB - Viquipèdia, l’enciclopèdia lliure’. [Online]. Available: https://ca.wikipedia.org/wiki/Model_de_color_RGB. [Accessed: 17-Aug-2021].
- [10] ‘Model de color HSV - Viquipèdia, l’enciclopèdia lliure’. [Online]. Available: https://ca.wikipedia.org/wiki/Model_de_color_HSV. [Accessed: 17-Aug-2021].
- [11] ‘Herramienta de diagrama de caso de uso libre’. [Online]. Available: <https://online.visual-paradigm.com/es/diagrams/solutions/free-use-case-diagram-tool/#>. [Accessed: 14-Aug-2021].
- [12] B. Stroustrup, ‘The C++ Programming Language, 3rd Edition’, *Vasa*, p. 962, 1994.
- [13] ‘IDE de Visual Studio 2019: software de programación para Windows’. [Online]. Available: <https://visualstudio.microsoft.com/es/vs/>. [Accessed: 15-Aug-2021].
- [14] ‘Introduction - OpenCV Tutorial C++’. [Online]. Available: <https://www.opencv-srf.com/p/introduction.html>. [Accessed: 27-Jul-2021].
- [15] ‘cvui: A GUI lib built on top of OpenCV drawing primitives | LearnOpenCV’. [Online]. Available: <https://learnopencv.com/cvui-gui-lib-built-on-top-of-opencv-drawing-primitives/>. [Accessed: 15-Aug-2021].

- [16] ‘cvui - (very) simple UI lib built on top of OpenCV drawing primitives’. [Online]. Available: <https://dovyski.github.io/cvui/>. [Accessed: 15-Aug-2021].
- [17] ‘Wansview Cámara Vigilancia WiFi Exterior, 1080P Cámara IP WiFi de Seguridad con Visión Noturna Detección de Movimiento Audio Bidireccional, Soporta Alexa RTSP Onvif, IP66 Impermeable, W6 (Blanco) : Amazon.es: Electrónica’. [Online]. Available: <https://www.amazon.es/dp/B08CXDZMKH?tag=camarasvigiauto-21&linkCode=osi&th=1&psc=1>. [Accessed: 15-Aug-2021].
- [18] ‘Cámara Digital Nikon D3200’. [Online]. Available: https://www.nikon.es/es_ES/product/discontinued/digital-cameras/2016/d3200#tech_specs. [Accessed: 15-Aug-2021].
- [19] ‘Cámara web HD Logitech C505 con micrófono de largo alcance’. [Online]. Available: <https://www.logitech.com/es-es/products/webcams/c505-hd-webcam.960-001364.html?crd=34>. [Accessed: 15-Aug-2021].
- [20] ‘Cable USB Tipo C, RAVIAD Cable USB C a USB 3.0 Cable Tipo C Carga Rápida y Sincronización Compatible con Galaxy S10/S9/S8/Note 10, Huawei P30/P20, Mi A1/Mi A2 y más - 1M, Gris : Amazon.es: Informática’. [Online]. Available: https://www.amazon.es/RAVIAD-Rápida-Sincronización-Compatible-Galaxy/dp/B089GD78ZW/ref=sr_1_5?__mk_es_ES=ÅMÅŽÕÑ&dchild=1&keywords=cable+usb-c+a+usb&qid=1629045925&sr=8-5. [Accessed: 15-Aug-2021].
- [21] ‘Rankie Cable Mini HDMI a HDMI, vídeo 4K, Compatible con Ethernet, 3D y ARC, 1,8 m, Negro : Amazon.es: Electrónica’. [Online]. Available: https://www.amazon.es/Rankie-Cable-vídeo-Compatible-Ethernet/dp/B01KRKO4MM/ref=sr_1_5?adgrpid=59028305474&dchild=1&gclid=CjwKCAjw9uKIBhA8EiwAYPUS3PrmWbCGh_PDv5A4QutnjESIShg6bFfI-Oikk6xN91Qax4FdDQhuGhoCaTkQAvD_BwE&hvadid=275357092486&hvdev=c&hvlocphy=1005429&hvnetw=g&hvqmt=e&hvrnd=7347689989545647613&hvtag=rgid=kwd-300393310709&hydadcr=23390_1797049&keywords=cable+mini+hdmI&qid=1629041836&sr=8-5. [Accessed: 15-Aug-2021].
- [22] ‘DIWUER Capturadora de Video HDMI, 4K HDMI a USB 2.0 Convertidor Video Audio, HDMI Vídeo Game Capture 1080P 30FPS para Edite Video/Juego/Transmisión/Enseñanza en línea : Amazon.es: Informática’. [Online]. Available: https://www.amazon.es/DIWUER-Capturadora-Convertidor-Transmisión-Enseñanza/dp/B08BC52KCT/ref=sr_1_2_sspa?__mk_es_ES=ÅMÅŽÕÑ&crd=3C4AJFI07VRC0&dchild=1&keywords=capturadora+video+hdmI&qid=1629042214&sprefix=capturadora%2Caps%2C195&sr=8-2-spons&psc=1&spLa=ZW5jcmlwdGVkUXVhbGhmaWVvPUEzVDRNNVREWVVJNUZLJmVuY3J5cHRlZElkPUEwMzc1NjUwMIA2SDAwSTBRWVhaVSZlbnNyeXB0ZWZlZElkPUEwMjc4MjM0M0ZVRDRDRIBRUjA5OSZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=. [Accessed: 15-Aug-2021].
- [23] ‘Releases - OpenCV’. [Online]. Available: <https://opencv.org/releases/>. [Accessed: 22-Aug-2021].

9 Annexes

9.1 Guia d'instal·lació

A continuació es detallaran els passos a realitzar per poder instal·lar els programes i llibreries necessàries per poder executar l'aplicació realitzada:

01. Descarregar i instal·lar Visual Studio 2019 [13].
02. Descarregar el .zip del codi del projecte.
03. Descomprimir el .zip.
04. Obrir el projecte des del Visual Studio 2019.
05. Descarregar l'opencv 3.4 [23],
06. Afegir les llibreries d'OpenCV al projecte:
 - a. Obrim les propietats del projecte
 - b. En l'apartat VC++ Directories > Include Directories afegim el path "XXX/opencv/build/include".
 - c. En l'apartat VC++ Directories > Library Directories afegim el path "XXX/opencv/build/x64/VC14/lib".
 - d. En l'apartat Linker > Additional Dependencies afegim el nom del fitxer "opencv_world3412d.lib" que trobem dins del path de l'apartat c.
07. Executem el projecte.