



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona



GRAU EN ENGINYERIA INFORMÀTICA

ESPECIALITAT EN COMPUTACIÓ

TREBALL FINAL DE GRAU

---

# VÍDEO ANALÍTICA INTEL·LIGENT EN TEMPS REAL

---

ARNAU RUANA RAMOS

Director

VÍCTOR BAUTISTA SAIZ

Tutor

GERARD ESCUDERO BAKX

22/10/2021

# Resum

---

Aquest projecte té com a objectiu principal el desenvolupament *software* d'una analítica de vídeo en temps real mitjançant algorismes basats en Intel·ligència Artificial per a l'empresa Lanaccess. Aquest desenvolupament engloba totes les parts de *back-end* i *front-end* necessàries per a la integració de l'analítica dins l'empresa així com la creació, entrenament i testeig de les respectives Xarxes Neuronals utilitzades.

# Resumen

---

Este proyecto tiene como objetivo principal el desarrollo *software* de una analítica de video en tiempo real mediante algoritmos basados en Inteligencia Artificial para la empresa Lanaccess. Este desarrollo engloba todas las partes de *back-end* y *front-end* necesarias para la integración de la analítica dentro de la empresa así como la creación, entrenamiento y testeo de las respectivas Redes Neuronales utilizadas.

# Abstract

---

The main objective of this project is the software development of a real-time video analytic using algorithms based on Artificial Intelligence for the company Lanaccess. This development encompasses all the parts of back-end and front-end needed for the integration of the analytic within the company as well as the creation, training and testing of the respective Neural Networks used.

# Índex

---

<b>1</b>	<b>Introducció</b>	<b>11</b>
<b>2</b>	<b>Contextualització</b>	<b>12</b>
2.1	Definició de conceptes . . . . .	12
2.1.1	Intel·ligència Artificial . . . . .	12
2.1.1.1	Machine Learning . . . . .	13
2.1.1.2	Deep Learning . . . . .	13
2.1.1.3	Xarxa Neuronal . . . . .	13
2.1.1.4	Funció d'Activació . . . . .	15
2.1.1.5	Model . . . . .	16
2.1.1.6	Dataset . . . . .	16
2.1.2	Vídeo Analítica . . . . .	16
2.1.2.1	Metadades . . . . .	17
2.1.2.2	Frame . . . . .	17
2.1.2.3	Cerca Forense . . . . .	18
2.1.2.4	Edge computing . . . . .	18

2.2	Descripció del problema . . . . .	18
2.3	Solucions prèvies . . . . .	19
2.3.1	Bosch Forensic Search Plug-in . . . . .	19
2.3.2	Axxonsoft Forensic Search . . . . .	19
2.3.3	Avigilon Video Analytics . . . . .	20
2.4	Actors implicats . . . . .	20
<b>3</b>	<b>Abast</b>	<b>21</b>
3.1	Objectiu principal . . . . .	21
3.2	Subobjectius . . . . .	22
3.3	Requeriments . . . . .	22
3.4	Arquitectura . . . . .	22
3.5	Obstacles i riscos . . . . .	23
<b>4</b>	<b>Metodologia</b>	<b>24</b>
4.1	Metodologia de treball . . . . .	24
4.2	Seguiment . . . . .	26
<b>5</b>	<b>Planificació temporal</b>	<b>27</b>
5.1	Descripció de les tasques . . . . .	27
5.1.1	Gestió del projecte [GP] . . . . .	28
5.1.1.1	Contextualització i abast [GP-C] . . . . .	28
5.1.1.2	Planificació temporal [GP-T] . . . . .	28
5.1.1.3	Pressupost i sostenibilitat [GP-S] . . . . .	28
5.1.1.4	Memòria parcial [GP-I] . . . . .	28
5.1.1.5	Memòria final [GP-F] . . . . .	29

5.1.1.6	Presentació [GP-P]	29
5.1.1.7	Reunions [GP-R]	29
5.1.2	Treball previ [TP]	29
5.1.2.1	Autoaprenentatge [TP-A]	30
5.1.2.2	Preparació de l'entorn de treball [TP-E]	30
5.1.3	Models [MD]	30
5.1.3.1	Gènere [MD-G]	30
5.1.3.2	Edat [MD-E]	30
5.1.3.3	Color roba superior [MD-S]	31
5.1.3.4	Color roba inferior [MD-I]	31
5.1.4	Integració i testeig [IT]	31
5.1.5	Optimització i millora [OM]	31
5.2	Representació gràfica	32
5.3	Recursos necessaris	34
5.3.1	Humans [R1]	34
5.3.2	Hardware [R2]	34
5.3.3	Software [R3]	34
5.3.3.1	Python	35
5.3.3.2	Anaconda	35
5.3.3.3	Tensorflow	35
5.3.3.4	GitLab (Git)	35
5.3.3.5	Slack	36
5.3.3.6	Overleaf	36
5.4	Gestió del risc	36

<b>6</b>	<b>Gestió econòmica</b>	<b>37</b>
6.1	Costos de personal . . . . .	37
6.1.1	Programador . . . . .	37
6.1.2	Cap del projecte . . . . .	38
6.1.3	Total . . . . .	38
6.2	Costos per activitat . . . . .	38
6.3	Costos genèrics . . . . .	39
6.3.1	Software . . . . .	39
6.3.2	Hardware . . . . .	39
6.3.3	Internet . . . . .	40
6.3.4	Total . . . . .	40
6.4	Contingències . . . . .	41
6.5	Imprevistos . . . . .	41
6.6	Cost total . . . . .	41
6.7	Control de gestió . . . . .	42
<b>7</b>	<b>Sostenibilitat</b>	<b>43</b>
7.1	Dimensió econòmica . . . . .	43
7.2	Dimensió ambiental . . . . .	44
7.3	Dimensió social . . . . .	44
<b>8</b>	<b>Canvis realitzats</b>	<b>45</b>
<b>9</b>	<b>Solució plantejada</b>	<b>47</b>
9.1	Visió general . . . . .	48
9.2	Models . . . . .	51

9.2.1	Predicció del gènere . . . . .	51
9.2.1.1	Dataset . . . . .	51
9.2.1.2	Model . . . . .	52
9.2.2	Predicció de l'edat . . . . .	52
9.2.2.1	Dataset . . . . .	53
9.2.2.2	Model . . . . .	53
9.2.3	Detecció de cares . . . . .	54
9.3	Plugins . . . . .	54
9.3.1	Gender . . . . .	55
9.3.2	Age . . . . .	56
9.3.3	Face . . . . .	56
9.3.4	Agender . . . . .	56
9.4	Gestors . . . . .	56
9.4.1	Host . . . . .	57
9.4.2	Gateway . . . . .	57
9.4.3	Media Server . . . . .	57
9.4.4	View . . . . .	57
9.5	Configuració . . . . .	58
<b>10</b>	<b>Treball futur</b>	<b>60</b>
10.1	Millorar LAVA-AGENDER . . . . .	60
10.2	Desenvolupar LAVA-COLORS . . . . .	61
<b>11</b>	<b>Conclusions</b>	<b>62</b>
<b>12</b>	<b>Agraïments</b>	<b>64</b>





# Índex de figures

---

1.1	Logo de l'empresa Lanaccess Telecom [1] . . . . .	11
2.1	Diagrama que mostra la relació entre les diferents branques de la IA [2] .	13
2.2	Exemple de xarxa neuronal densa [3] . . . . .	14
2.3	Exemple de CNN per classificació [4] . . . . .	15
2.4	Exemple d'una vídeo analítica d'extracció de vehicles [5] . . . . .	17
2.5	Figura que mostra la relació i interacció entre les tres capes de processament de dades [6] . . . . .	18
3.1	Imatge que mostra l'arquitectura utilitzada (extret d'un document privat de Lanaccess) . . . . .	23
4.1	Figura que mostra les diferents premisses del mètode <i>Kanban</i> [7] . . . . .	25
4.2	Figura que mostra un exemple de la representació del progrés d'un projecte en el mètode <i>Kanban</i> (elaboració pròpia) . . . . .	25
5.1	Diagrama de Gantt realitzat amb l'eina Gantter (elaboració pròpia) . . .	33
9.1	Diagrama que mostra com es realitza la interacció entres les diferents parts implicades en el funcionament de l'analítica (elaboració pròpia) . . .	48

9.2	Diagrama de flux que mostra l'ordre d'execució de les diferents parts de la analítica sense tenir en compte les configuracions (elaboració pròpia)	49
9.3	Exemple d'un frame analitzat amb la nostra analítica	50
9.4	Mostra de les imatges que composen el dataset utilitzat per a la predicció del gènere (extracció del <i>dataset</i> [8])	52
9.5	Mostra de les imatges que composen el dataset utilitzat per a la predicció de l'edat (extracció del <i>dataset</i> [9])	53
9.6	Demostració del funcionament del filtrat de cares per NMS [10]	55
9.7	Exemple de configuració d'un dispositiu en el LAVA	58
9.8	Exemple de configuració d'un flux de vídeo en el LAVA	59
9.9	Opció que permet publicar el vídeo resultant en el LAVA	59

# Índex de taules

---

5.1	Taula resum de les tasques, durada, dependències i recursos necessaris (elaboració pròpia) . . . . .	32
6.1	Taula resum amb el sou brut i la retribució per cada perfil (elaboració pròpia) . . . . .	38
6.2	Taula resum amb l'estimació del cost per activitat i el total (elaboració pròpia) . . . . .	39
6.3	Taula resum amb el cost total del <i>software</i> utilitzat (elaboració pròpia) . .	40
6.4	Taula resum amb els costos genèrics totals (elaboració pròpia) . . . . .	41
6.5	Taula resum amb els costos dels imprevistos en funció de la seva probabilitat (elaboració pròpia) . . . . .	41
6.6	Taula resum amb el cost total del projecte segons cada categoria (elaboració pròpia) . . . . .	42

# Introducció

---

L'empresa Lanaccess Telecom és una empresa mitjana àmpliament reconeguda en el sector de la videovigilància la qual s'ha forjat, al llarg dels seus vint-i-cinc anys, un nom dins la indústria gràcies a productes *hardware* com el Recorder [11] o solucions *software* com el VMS [12]. Durant els últims anys, també ha volgut endinsar-se en el món de la Intel·ligència Artificial mitjançant la sortida al mercat de productes com el LAVA (LAnaccess Video Analytics) [13].



Figura 1.1: Logo de l'empresa Lanaccess Telecom [1]

La nostra feina consisteix en ampliar les funcionalitats del LAVA afegint-hi una nova analítica encarregada de la predicció del gènere i l'edat de les persones que apareguin en un o diversos fluxos de vídeo. Aquesta solució estarà basada en Xarxes Neuronals que duran a terme les prediccions mentre es guarden les metadades pertinents per a possibles futures cerques.

Així doncs amb aquesta memòria pretenem exposar tot el treball realitzat al llarg dels nou mesos que ha englobat el conveni de pràctiques amb aquesta empresa juntament amb el respectiu Treball Final de Grau que s'ha estat realitzant en paral·lel. L'explicació abastarà des de la posada en context del treball i la seva gestió fins al treball futur que es podria fer per millorar-lo passant per l'explicació i justificació de tots els passos que hem seguit per al seu desenvolupament.

# Contextualització

---

Per donar una mica de context direm que aquest TFG pertany a la modalitat B (conveni de cooperació educativa amb empresa) i, per consegüent, sorgeix d'una de les necessitats de l'empresa tecnològica, esmentada anteriorment, Lanaccess. Aquesta necessitat consisteix en la realització d'un projecte per a l'anàlisi forense mitjançant analítica de vídeo en temps real utilitzant algorismes basats en Intel·ligència Artificial. Tot això desenvolupat sota el marc del paradigma *edge computing*, el paradigma computacional per excel·lència dels darrers anys.

## 2.1 Definició de conceptes

A continuació definirem alguns conceptes clau a l'hora d'entendre els termes i expressions que usarem al llarg d'aquest document. Per tal de fer-ho el més entenedor possible, dividirem les definicions en diferents grups segons l'àmbit tecnològic al qual pertanyin i adjuntarem les imatges que creguem oportunes.

### 2.1.1 Intel·ligència Artificial

La Intel·ligència Artificial (IA) es pot definir com la capacitat que tenen els ordinadors de realitzar tasques associades amb comportaments intel·ligents com ara “raonar” o “pensar” per sí mateixes [14] [15].

Aquesta disciplina, al llarg de la història, s'ha subdividit en diferents camps com el de Processament del Llenguatge Natural (NPL), Sistemes Experts o Aprenentatge Màquina

entre d'altres. Seguidament desglossarem aquest últim camp en uns quants conceptes que són, igualment, de gran importància.

#### 2.1.1.1 Machine Learning

El *Machine Learning* (ML), o aprenentatge automàtic, és una branca de la IA que consisteix en l'entrenament d'un model, concepte que definirem a continuació (secció 2.1.1.5), mitjançant algorismes dissenyats per a la cerca i abstracció de característiques i patrons en quantitats immenses de dades amb l'objectiu de predir o prendre decisions amb futures dades [2] [16].

#### 2.1.1.2 Deep Learning

El *Deep Learning* (DL), també conegut com aprenentatge profund, és un subconjunt del *Machine Learning* que, com a diferència, utilitza xarxes neuronals de múltiples capes. Això implica que els algorismes d'abstracció de patrons són aplicats a cada una d'elles permetent, d'aquesta manera, l'aprenentatge de dades molt més complexes, com ara la identificació de patrons en imatges [17] [18].

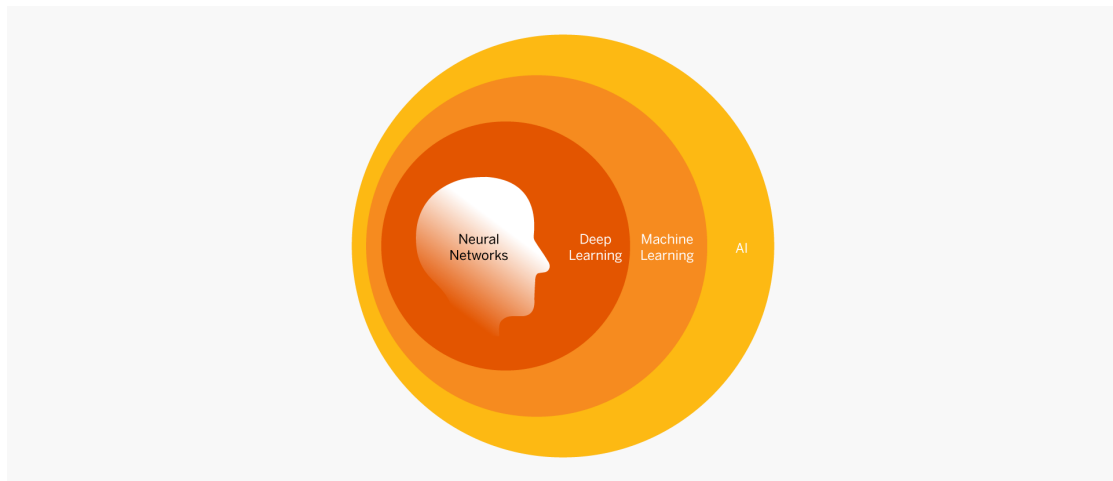


Figura 2.1: Diagrama que mostra la relació entre les diferents branques de la IA [2]

#### 2.1.1.3 Xarxa Neuronal

Una xarxa neuronal (NN), en termes computacionals, es pot definir com la interconnexió de neurones artificials que pretenen simular la xarxa neuronal biològica. Cada neurona rep com a entrada la sortida de les neurones de l'anterior capa multiplicades per un

determinat pes i aplica una funció d'activació (definida a l'apartat 2.1.1.4) concreta per calcular la seva sortida [19] [20] [21].

La última capa de la xarxa és, per tant, la que conté els resultats esperats. Per exemple 0 si s'ha detectat que hi ha un gat en una imatge o 1 si es creu que hi ha un gos. En aquest mateix exemple la capa d'entrada seria la imatge en qüestió.

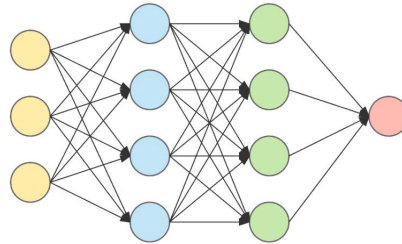


Figura 2.2: Exemple de xarxa neuronal densa [3]

Existeixen diversos tipus de xarxes neuronals que s'adapten a cada una de les necessitats que hi poden haver [22] [23]. Les *Recurrent Neural Networks* (RNN), *Radial Basis Function Neural Networks* (RBFNN) o *Feedforward Neural Networks* (FNN) són algunes de les NN més conegudes i utilitzades en les quals no hi entrarem en detalls ja que no les farem servir en el nostre treball. No obstant, sí que detallarem en profunditat les característiques més rellevants de les *Convolutional Neural Networks* (CNN), les xarxes per excel·lència en l'abstracció de patrons en imatges.

**Xarxes Neuronals Convolucionals** Les CNN són xarxes neuronals compostes principalment per diverses capes convolucionals interconnectades entre sí, anomenades també convolucions. Cada una d'aquestes convolucions es pot definir com un mecanisme de filtre que, a mesura que un patró es repetix en les dades d'entrada, memoritza la ubicació on s'acostuma a produir i la importància que se li ha de donar de cara a la sortida o resultat final [24]. Podríem dir, llavors, que cadascuna de les convolucions que formen una xarxa s'encarrega d'analitzar una de les característiques que té l'entrada i, per consegüent, totes elles en conjunt seran capaces de fer prediccions sobre aquesta [25].

Aquestes convolucions acostumen a anar acompanyades per capes que redueixen la dimensionalitat de les dades d'entrada anomenades *Pooling Layers* [26]. Principalment serveixen per poder tenir més filtres a canvi d'anar perdent qualitat en les dades d'entrada. Això implica reduir considerablement el nombre de neurones necessàries i, consegüentment, el temps d'entrenament necessari del model. Addicionalment, a la sortida de la part convolucional de la xarxa, s'hi adhireix l'encarregada de realitzar la classificació o regressió pertinents depenent del tipus de problema.

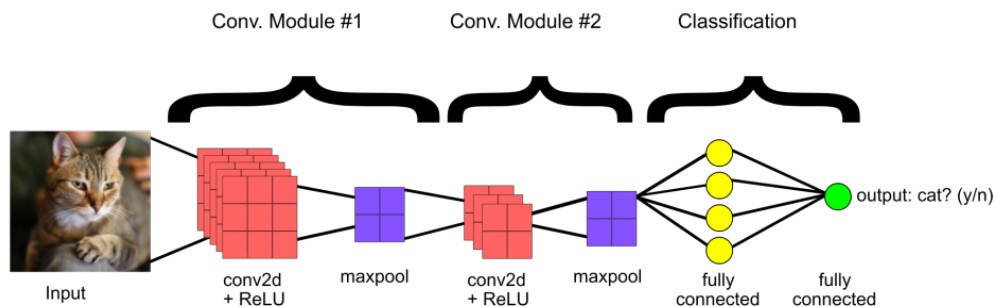


Figura 2.3: Exemple de CNN per classificació [4]

#### 2.1.1.4 Funció d'Activació

Les funcions d'activació (FA) són una component clau en les NN ja que, mitjançant equacions matemàtiques, s'encarreguen de determinar la sortida que produeix una neurona i, per tant, la sortida final de la xarxa [27] [28] [29].

Existeixen diversos tipus de funcions d'activació, utilitzades segons les necessitats de la xarxa neuronal depenent del tipus de problema a resoldre. Seguidament definirem algunes de les funcions d'activació més conegudes i utilitzades que seran de vital importància per a la comprensió d'aquest treball.

**Linear** Aquesta funció d'activació ve a ser la funció identitat. És a dir, el valor d'entrada és el mateix que es dona com a sortida. És altament utilitzada per a sortides en problemes de regressió.

$$f(x) = x$$

**ReLU** Aquesta és la més utilitzada en les capes ocultes de les xarxes neuronals. És una funció que només deixa passar els valors positius, és a dir, tots els valors negatius passen a valer 0.

$$ReLU(x) = \max(0, x)$$

**Sigmoid** Funció d'activació que s'encarrega de retornar valors dins del rang  $[0 - 1]$ . Com més gran sigui el valor, més s'aproparà a 1 i com més petit, a 0. És molt similar a la funció d'activació lineal. També és comunament utilitzada en sortides de models basats en regressió amb la diferència que s'usa quan el valor de sortida està normalitzat entre 0 i 1.

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}$$



**Softmax** Funció d'activació per excel·lència en la capa de sortida de models basats en classificació. S'utilitza quan les classes són mútuament exclusives una de l'altra. Retorna un vector de valors que, conjuntament, sumen 1 i que es poden interpretar com probabilitats de pertànyer a cada una de les classes respectivament.

$$\text{Softmax}(\vec{x})_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \text{ on } K \text{ nombre de classes}$$

#### 2.1.1.5 Model

Un model (MD) és, bàsicament, un fitxer fruit d'una Xarxa Neuronal prèviament entrenada que és utilitzat com a eina per a predir dades futures [30].

#### 2.1.1.6 Dataset

Un *dataset* (DS), com el seu nom en anglès indica, és un conjunt de dades que, en l'àmbit de la Intel·ligència Artificial, són utilitzades per a entrenar i testear els models [31] [32].

La principal característica dels *datasets* és que no tenen un format preestablert i, per tant, poden contenir qualsevol tipus de dades i estar ordenats seguint qualsevol criteri. El que sí és indispensable és que continguin tant les dades d'entrada que llegirà el model com la sortida (o conjunt d'elles) que s'espera per part d'ell. D'aquesta manera, durant l'entrenament del model, es pot saber en qualsevol moment si els resultats que està predient són correctes o no.

Una altra característica bastant recurrent és que estiguin subdividits en dos subconjunts: entrenament i testeig. Això serveix per simplificar el processament de les dades que ha de realitzar, amb posterioritat, el programa encarregat de llegir el DS.

### 2.1.2 Vídeo Analítica

La vídeo analítica (VA), també coneguda com analítica de vídeo, fa referència a aplicacions *software* encarregades de generar automàticament descripcions de tot el que està passant en un vídeo. Aquestes descripcions s'anomenen metadades i es poden usar per enumerar o detectar qualsevol tipus d'objecte en un vídeo, com persones o vehicles [33] [34] [35].

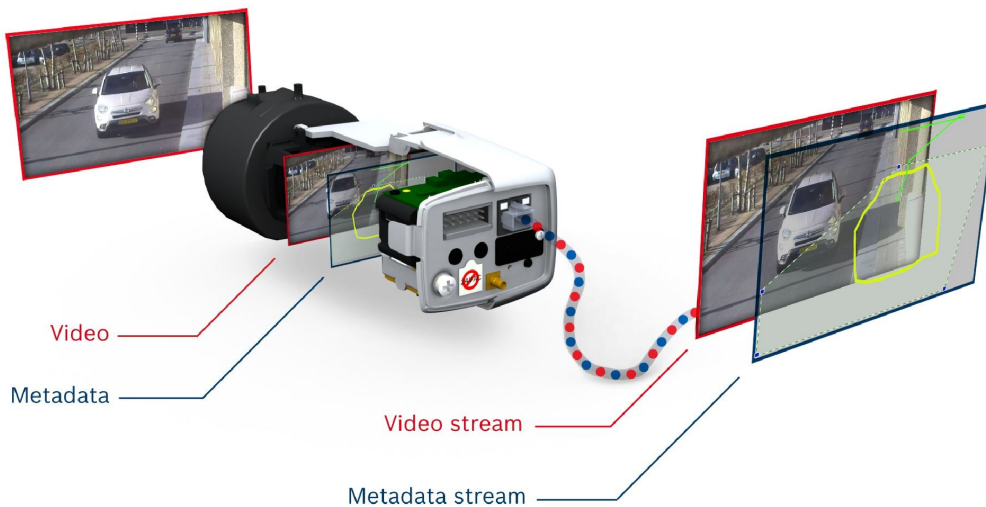


Figura 2.4: Exemple d'una vídeo analítica d'extracció de vehicles [5]

#### 2.1.2.1 Metadades

Les metadades són, formalment, dades que descriuen altres dades i que, d'alguna manera, estan ocultes [36]. És a dir, en una imatge no podem veure dades com qui va ser la persona responsable de realitzar la fotografia, o la ubicació on es va fer; no obstant, les metadades d'aquella imatge en qüestió sí que les emmagatzema internament.

En el camp de la vídeo analítica definirem les metadades com tot aquell conjunt de dades que, en un principi, no se sabien i que, gràcies al tractament del vídeo mitjançant determinats models, s'han arribat a extreure o predir. Per exemple, en un vídeo no tenim informació de quantes persones han aparegut llevat que un *software* de reconeixement de persones les vagi comptant i enumerant al llarg de la gravació.

#### 2.1.2.2 Frame

La paraula *frame*, traduïda de l'anglès com “marc”, fa referència a una de les moltes imatges que, en seqüència, composen un vídeo.

### 2.1.2.3 Cerca Forense

La Cerca Forense (FS) es pot definir com l'acció de filtrar tots els frames que componen una seqüència de vídeo amb l'objectiu de trobar ràpidament certes dades d'interès establertes per l'usuari. Això és possible gràcies a la constant extracció de metadades que realitzen les analítiques i que són emmagatzemades en unes bases de dades especialitzades [37].

### 2.1.2.4 Edge computing

L'*edge computing* (EC) pot ser definit com un paradigma de computació que pretén repartir en diferents nivells i “apropar”, en la mesura del possible, tota o part de la càrrega computacional i l'emmagatzematge de les dades a l'usuari final d'una aplicació o servei amb l'objectiu de minimitzar el temps de resposta i l'ample de banda usat [38].

Qualsevol tipus de computació que es realitzi fora del núvol per passar a fer-se relativament a prop de l'usuari, sobretot quan es tracta d'aplicacions que requereixen processar informació en temps real, forma part d'aquest paradigma [39].

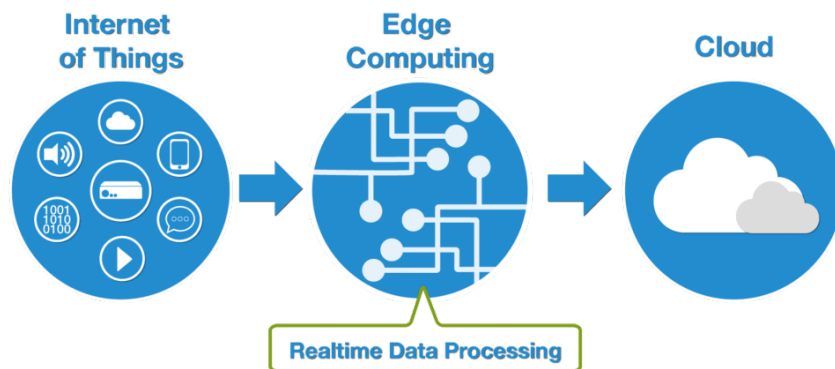


Figura 2.5: Figura que mostra la relació i interacció entre les tres capes de processament de dades [6]

## 2.2 Descripció del problema

Els sistemes de videovigilància estan contínuament expandint-se per tal d'incloure més càmeres i, alhora, més capacitat d'emmagatzematge per encabir el màxim volum de dades possible. Fer el rastrejament d'informació de gran valor per al client (com per

exemple, la de trobar un sospitós) en aquesta inimaginable quantitat d'hores de vídeo gravades pot esdevenir una tasca extremadament complexa.

Tornant a l'exemple del sospitós, quan es comet un crim es poden tardar hores, o fins i tot dies, en localitzar el presumpte autor en les gravacions. Tot i disposar de servidors dedicats exclusivament a vídeo-analítica es poden tardar diversos minuts en processar només una hora de gravació. Tot aquest temps invertit pot arribar a ser un gran problema ja que és temps perdut en la cerca i captura del subjecte.

L'emmagatzematge continu de les dades clau mitjançant l'anàlisi de vídeo en temps real (metadades) permet agilitzar infinitament la complexa cerca descrita amb anterioritat convertint, així, l'anàlisi de les gravacions en un filtrat instantani (o quasi instantani) de les dades per tenir-les a la nostre disposició quan sigui necessari.

## 2.3 Solucions prèvies

El treball que volem desenvolupar no és nou i, per tant, ja ha estat tractat amb anterioritat. Noms reconeguts com Bosch, Aviglion o Axxonsoft són algunes de les empreses que han desenvolupat un software força similar al que volem realitzar nosaltres o, si més no, amb propòsits semblants.

A continuació descriurem i analitzarem de manera breu les solucions oferides per aquestes empreses així com adjuntarem els respectius enllaços on poder trobar més informació per a cada un dels seus projectes.

### 2.3.1 Bosch Forensic Search Plug-in

L'empresa Bosch ens brinda un *plug-in* dedicat exclusivament a l'anàlisi forense mitjançant càmeres amb diverses analítiques de vídeo. Segons el web [5] les analítiques utilitzen metadades per a la identificació d'objectes o incidents de manera ràpida i fàcil.

Aquesta aplicació ofereix solució per a problemes com la cerca de nens perduts, evidenciar conductes criminals o, entre d'altres, alertar preventivament abans que es produeixi alguna situació considerada perillosa.

### 2.3.2 Axxonsoft Forensic Search

Aquesta solució d'Axxonsoft està composta per tres *plug-ins* diferents: MomentQuest, FaceSearch i ANPRSearch. Per a obtenir informació més detallada sobre cadascuna d'elles podeu visitar el web [40].

La primera analítica va dirigida a la identificació de vehicles que creuin una determinada zona i, posteriorment, apareguin en una altra, ambdues determinades per l'usuari. La segona se centra més en la identificació facial i en la cerca de les cares que, en un percentatge elevat, coincideixin amb la imatge desitjada. L'última va enfocada a la detecció i lectura de les matrícules dels vehicles per a quan siguin sol·licitades per l'usuari.

### **2.3.3 Avigilon Video Analytics**

La solució proposada per Avigilon és molt similar a les altres i, de fet, barreja analítiques d'ambdues [41]. Reconeixement facial, detector de matrícules o detecció de situacions perilloses són algunes de les solucions de les quals disposen.

També volem puntualitzar que disposa d'un detector d'aparença que emmagatzema la informació del gènere d'una persona i el seu color de roba.

## **2.4 Actors implicats**

Els actors implicats en el desenvolupament d'aquest projecte som l'equip de vídeo analítica i d'Intel·ligència Artificial de l'empresa, els quals vetllarem, en tot moment, pel correcte desenvolupament del producte; juntament amb el director i tutor d'aquest treball.

Per altra banda, el sistema que volem desenvolupar va dirigit a un ventall molt ampli de sectors. Principalment des de la banca o supermercats fins als sectors penitenciaris o d'oci (com casinos) passant per hospitals i fàbriques. D'entrada no va dirigit a usuaris particulars. Qualsevol d'aquests actors darrerament esmentats es poden beneficiar d'aquest sistema, sempre i quan disposin de càmeres que puguin ser objecte d'aquesta tecnologia basada en vídeo analítica en temps real.

# Abast

---

*ATENCIÓ: aquest capítol ha patit lleugers canvis respecte del GEP i, per tant, algunes parts podrien no correspondre amb la versió final del treball. Podeu trobar més detalls sobre els canvis realitzats a la secció 8.*

A continuació sintetitzarem, per objectius, l'explicació que hem dut a terme en la descripció del problema (secció 2.2) definint i analitzant quin serà l'objectiu principal del projecte i en quants i quins subobjectius estarà dividit.

També parlarem sobre els requisits que haurà de tenir l'analítica desenvolupada i els possibles obstacles i riscos que ens podem trobar al llarg del seu desenvolupament.

## 3.1 Objectiu principal

L'objectiu principal d'aquest treball és el d'extreure i emmagatzemar diverses metadades de persones sobre un flux de vídeo (o diversos). Per a aconseguir-ho, caldrà implementar una solució de vídeo analítica basada en Xarxes Neuronals que, en base a les deteccions de persones sobre un vídeo en temps real, s'extregui la informació rellevant. Aquesta informació es divideix en tres categories: gènere, edat i color de la roba.

## 3.2 Subobjectius

Com ja hem dit, tenim tres grans categories de les quals extreure metadades. Hem considerat oportú dividir l'objectiu principal en els següents quatre subobjectius, representant cada una de les característiques:

1. **Extracció del gènere:** s'espera obtenir un model que classifiqui la persona en qüestió en home, dona o desconegut (en cas que no es pugui determinar).
2. **Extracció de l'edat:** s'espera classificar la persona en un rang d'edat específic que determini, aproximadament, la seva edat mitjançant un model. Els rangs exemple (poden variar segons convingui) són, en anys: [0-10], [10-20], [20-35], [35-50], [50-65], [65-120] o desconegut.
3. **Extracció del color de roba superior:** en aquest cas volem determinar, amb un model, quin és el color de la part superior de la roba: desconegut, negre, blanc, blau, vermell, ...
4. **Extracció del color de roba inferior:** anàlogament a la característica anterior volem detectar el color de la roba de la part inferior de la persona: desconegut, verd, rosa, lila, groc, ...

## 3.3 Requeriments

La solució a desenvolupar ha d'estar integrada en la plataforma de vídeo analítica intel·ligent de Lanaccess. També requereix que sigui optimitzada per poder funcionar en temps real ocupant la menor quantitat de recursos possible amb l'objectiu de poder analitzar múltiples fluxos de vídeo simultàniament.

## 3.4 Arquitectura

En la xarxa local de cada oficina, estació o centre s'analitzaran els fluxos de vídeo disponibles en la xarxa de videovigilància. Les metadades resultants de les anàlisis s'emmagatzemaran en un servidor centralitzat. Des de qualsevol client del VMS web es podrà realitzar una cerca forense sobre aquestes dades emmagatzemades.

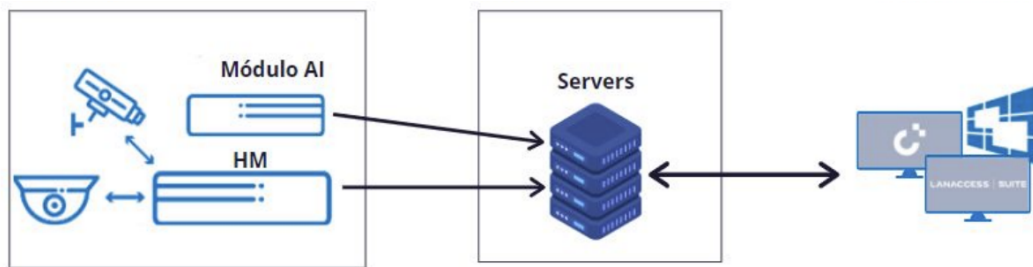


Figura 3.1: Imatge que mostra l'arquitectura utilitzada (extret d'un document privat de Lanaccess)

### 3.5 Obstacles i riscos

El principal obstacle a destacar és la inexperiència de l'autor en l'àmbit tecnològic en el qual ha de treballar. Més enllà d'algunes assignatures impartides a la facultat relacionades amb el *Machine Learning* com APA o amb la Intel·ligència Artificial com IA, l'autor no ha treballat ni pràcticament tocat l'àmbit del *Deep Learning* i, molt menys, ha desenvolupat un projecte basat en aquesta disciplina. L'autor tampoc no ha treballat mai per a ningú ni per a cap empresa, la qual cosa fa que se sotmeti a una pressió i exigència extres a l'hora de desenvolupar el treball.

Pel que fa als factors de risc podem destacar la possible manca de temps que es pot produir degut a les limitacions anteriorment descrites, així com la no correspondència dels resultats obtinguts amb els esperats ja que, al tractar-se d'entrenar models, pot esdevenir una tasca molt complexa el fet d'obtenir resultats satisfactoris i suficientment fiables.



# Metodologia

---

En aquest capítol explicarem quina metodologia hem triat per al desenvolupament del nostre treball així com els mètodes de seguiment emprats per tal d’assolir, en la mesura del possible, els nostres objectius.

## 4.1 Metodologia de treball

L’empresa no ens ha marcat, estrictament, cap metodologia de treball a seguir. Per tant, hem hagut d’investigar quines metodologies són les més conegudes i quina d’elles es pot adaptar millor al nostre projecte de cara a obtenir el màxim benefici possible al aplicar-la.

Després de buscar i investigar diferents metodologies de treball existents [42] [43] [44] creiem que la més adient per a nosaltres seria la denominada mètode *Kanban*. Aquest mètode es basa en el principi “stop starting, start finishing” i ens brinda una gran flexibilitat a l’hora de reajustar les prioritats quan sigui necessari en cas que sorgeixin imprevistos. Degut a tot això, aquesta metodologia també ens permet canviar el subprojecte en el qual treballar segons ens convingui.

Pensem que aquesta metodologia és la més adequada ja que en el nostre treball ens pot interessar reajustar les prioritats i, per exemple, deixar de treballar en el desenvolupament de l’extracció de gènere (deixar-la en segon pla) i passar a fer l’extracció de la roba inferior si ho creiem oportú. Això és degut al fet que els subobjectius definits anteriorment (secció 3.2) no tenen prerequisits entre sí, és a dir, es poden desenvolupar paral·lelament.

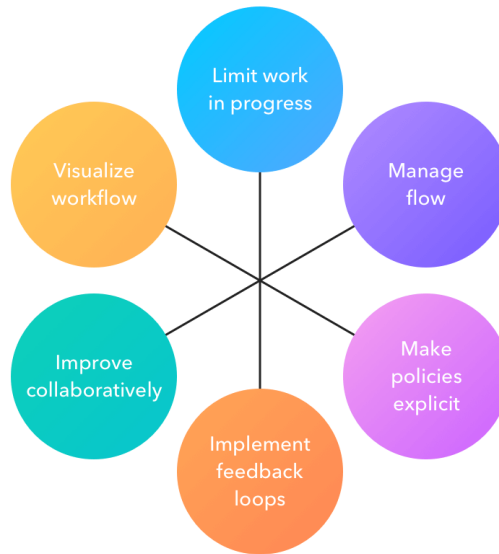


Figura 4.1: Figura que mostra les diferents premisses del mètode *Kanban* [7]

Adicionalment a tot el que hem dit, la metodologia *Kanban* ens brinda la possibilitat de tenir, en tot moment, una representació visual en forma de taula del progrés del projecte. També ens ajuda a incrementar la nostra productivitat enfocant-nos en cada una de les petites tasques a desenvolupar individualment.

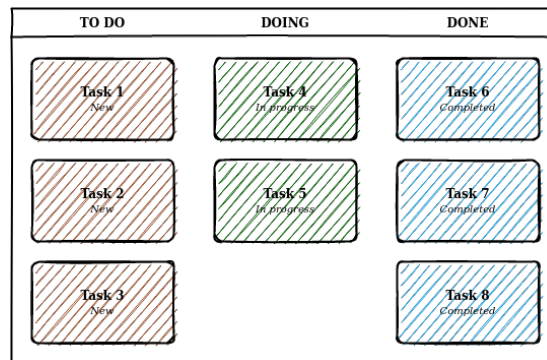


Figura 4.2: Figura que mostra un exemple de la representació del progrés d'un projecte en el mètode *Kanban* (elaboració pròpia)

## 4.2 Seguiment

Per una banda tenim el seguiment del projecte amb el nostre tutor el qual serà majoritàriament via correu electrònic, tot i que, puntualment, també podríem fer alguna videotrucada per facilitar la comunicació. Majoritàriament seran preguntes sobre la part escrita i oral del TFG, és a dir, la memòria i la posterior presentació.

Per l'altra banda tenim el seguiment que realitzem amb l'equip de vídeo analítica de l'empresa per corroborar que els objectius es van assolint satisfactòriament en el temps establert. Aquest seguiment es farà mitjançant una videotrucada setmanal, així com reunions en presencial si calen els dies que estiguem tots a l'oficina i no teletreballant. Així mateix disposem de la plataforma *Slack* (secció 5.3.3.5) per mantenir el contacte diari amb l'objectiu de preguntar qualsevol dubte o dificultat que tinguem. En aquest cas les preguntes seran relacionades amb el conjunt d'aplicacions a desenvolupar.

# Planificació temporal

---

*ATENCIÓ: aquest capítol ha patit alguns canvis respecte del GEP i, per tant, algunes parts podrien no correspondre amb la versió final del treball. Podeu trobar més detalls sobre els canvis realitzats a la secció 8.*

Amb el principal objectiu de finalitzar aquest treball en les dates previstes complint amb els objectius descrits en la secció d'abast (3) és de gran importància realitzar una planificació del temps segons les diverses tasques a desenvolupar, així com saber els recursos que seran necessaris per al desenvolupament del treball. De tot això parlarem en aquesta secció i, a més, adjuntarem algunes taules i diagrames que puguin ajudar visualment a entendre-ho.

Aquest treball va començar el passat dia 1 de febrer del 2021 i tindrà una dedicació prevista de 4 hores diàries de dilluns a divendres amb possibles variacions. Seguint les previsions de temps que descriurem a continuació preveiem un total de 500 hores per al treball.

## 5.1 Descripció de les tasques

Seguidament detallarem totes les tasques que involucren el nostre projecte i estimarem un nombre d'hores per a cada una d'elles. També les agruparem per temàtica i les identificarem per tal d'abreviar el text en les posteriors taules o diagrames i aconseguir major claredat.

### **5.1.1 Gestió del projecte [GP]**

La gestió del projecte és vital a l'hora de planificar, definir i documentar el treball de final de grau. Engloba totes les tasques no tècniques des del curs obligatori GEP fins a la redacció de la memòria final i les reunions amb el director i/o tutor. El temps total que abastarà aquesta part és el sumatori de les subtasques definides a continuació, és a dir, 170 hores aproximadament.

#### **5.1.1.1 Contextualització i abast [GP-C]**

Part fonamental de qualsevol projecte en la qual es defineixen les seves bases posant-lo en context i definint el seu abast. És tant important ja que, entre d'altres coses, es defineix l'objectiu principal que té el treball així com la metodologia que s'aplicarà. El temps emprat en la realització d'aquesta part han estat unes 24 hores.

#### **5.1.1.2 Planificació temporal [GP-T]**

En aquest apartat es defineixen les diferents fases que tindrà el projecte, s'estima la durada aproximada de cada una d'elles i s'analitzen les possibles dependències que podrien tenir. La durada de la planificació temporal ha estat de 12 hores aproximadament.

#### **5.1.1.3 Pressupost i sostenibilitat [GP-S]**

Aquí es desenvolupa una tasca similar a la de l'apartat anterior amb la diferència que es planifica tot el que té a veure amb els termes econòmics del projecte. S'estima el pressupost que serà necessari per dur-lo a terme i es fa una justificació àmplia sobre la seva sostenibilitat. Per al desenvolupament d'aquesta tasca creiem que seran necessàries una mitjana de 15 hores.

#### **5.1.1.4 Memòria parcial [GP-I]**

En aquest punt corregirem tot allò que ens hagi comunicat el nostre tutor o tutora del GEP sobre les entregues prèvies. També posarem en comú tota la feina dels punts anteriors en un sol document aconseguint, d'aquesta manera, tenir un sòlid fonament, en qüestió d'un mes, del nostre TFG. S'estima invertir un màxim d'una o dues hores, de mitjana, en la rectificació de cada apartat. Per tant, ens dona un total de 5 hores de dedicació aproximadament.

#### 5.1.1.5 Memòria final [GP-F]

Un cop tenim les bases del nostre treball (memòria parcial), i a mesura que avancem en la part tècnica, la nostra funció és anar redactant la memòria final del projecte. Això implica detallar tots i cada un dels passos que seguim, documentar i justificar les decisions que prenem en tot moment i adjuntar les imatges o gràfiques que creguem oportunes per la correcta explicació del treball. No sabem exactament el temps que necessitarem per a fer aquesta part ja que depèn de molts factors. Serem generosos i ho deixarem en 80 hores de treball.

#### 5.1.1.6 Presentació [GP-P]

Consisteix en la defensa i presentació del treball davant del tribunal encarregat d'avaluar-lo. És necessari haver acabat la memòria per tal de poder-lo presentar. Igual que en l'apartat anterior, considerem que és molt difícil predir el temps que ens portarà la preparació de la defensa. Per fer una predicció que no es quedi curta direm que necessitem unes 20 hores de dedicació.

#### 5.1.1.7 Reunions [GP-R]

Les reunions es duen a terme setmanalment i serveixen per fer el seguiment del treball segons els objectius setmanals o mensuals previstos. En el nostre cas les reunions seran de no més de 30 minuts amb alguns dels representants de l'equip de desenvolupament de l'empresa. Concretament amb la *Product Manager* i el *Lead Developer* que és, alhora, el director d'aquest projecte. Això prendrà un total de 10 hores.

També realitzarem totes les reunions que siguin necessàries amb el responsable d'Intel·ligència Artificial per tal de resoldre dubtes o complicacions que ens vagin sorgint al llarg del projecte. Aquestes seran de durada molt variable segons les necessitats del moment. Tirarem llarg i direm que implicarà un màxim de 4 hores.

En total, el temps estimat per a les reunions és de 14 hores.

#### 5.1.2 Treball previ [TP]

El treball previ a realitzar abans d'un projecte pot ser molt important a l'hora d'afrontar-lo més eficientment un cop es comenci. En el nostre cas hem necessitat realitzar les següents dues tasques com a preparació durant un total de 21 hores.

#### 5.1.2.1 Autoaprenentatge [TP-A]

En primer lloc, com ja vam comentar anteriorment a l'apartat de dificultats (secció 3.5), el nostre treball requereix d'uns coneixements que a la facultat no s'han impartit o, si més no, no s'han donat amb la profunditat necessària com per a desenvolupar aquest projecte sense dificultats. Per consegüent ha estat indispensable formar-nos en l'àmbit del *Deep Learning* així com en de les seves eines o programari més utilitzats en l'actualitat pels programadors que sí són experts en aquesta branca de la IA.

Això ha implicat llegir pausadament la documentació de TensorFlow (secció 5.3.3.3), una de les APIs més famoses en l'actualitat en referència al *Machine Learning*. Aquesta tasca es va dur a terme al llarg de la primera setmana de feina i, per tant, ha implicat 20 hores de dedicació.

#### 5.1.2.2 Preparació de l'entorn de treball [TP-E]

Aquest projecte ha requerit de la instal·lació d'algun *software* i de les versions específiques necessàries que no posseïem anteriorment. Concretament hem instal·lat Anaconda, TensorFlow, i algunes llibreries de Python (secció 5.3.3.1). Tot això en conjunt ha implicat 1 hora de feina.

#### 5.1.3 Models [MD]

Aquesta seria la part tècnica del projecte en la qual es desenvolupen els models de reconeixement demanats. Un total de 182 hores són les previstes.

##### 5.1.3.1 Gènere [MD-G]

El desenvolupament d'aquesta tasca consisteix en trobar un model que reconegui i classifiqui, satisfactòriament, el gènere d'una persona donada la seva cara. Un primer prototip d'aquest model que va funcionar quasi a la perfecció a la primera (si més no l'entrenament i testeig en el *dataset*), va comportar unes 12 hores de treball aproximadament.

##### 5.1.3.2 Edat [MD-E]

Aquesta tasca s'assembla molt a l'anterior però amb la diferència que s'ha de classificar l'edat de la persona enlloc del seu gènere. La realització d'aquesta classificació és molt més complexa que l'anterior i portem un total de 80 hores invertides amb resultats no

del tot satisfactoris. Creiem que en una setmana o dues més (20/40 hores) hauríem de ser capaços d'acabar-ho enllestint. En total haurà comportat 110 hores de dedicació en mitjana.

#### **5.1.3.3 Color roba superior [MD-S]**

Aquest model consisteix en el reconeixement del color de la roba de la part superior d'una persona. Encara no hem començat amb aquesta part però creiem que no hauria de ser tan complicat com l'anterior tasca ja que només hem d'obtenir un color. Amb una mitjana de 40 hores hauria de ser suficient.

#### **5.1.3.4 Color roba inferior [MD-I]**

Anàlogament, aquest model consisteix en la classificació del color de la part inferior de la roba d'una persona. Un cop realitzada la tasca anterior aquesta s'espera que sigui molt semblant. Creiem que amb unes 20 hores aproximadament podríem tenir-ho fet.

#### **5.1.4 Integració i testeig [IT]**

Arribats a aquest punt interessa integrar, adaptar i provar tot el treball tècnic previ (MD) en el sistema general de l'empresa. S'ha de provar que els models actuïn en temps real en el *hardware* proporcionat per l'empresa: targetes gràfiques Jetson Nano (secció 5.3.2). No és gens fàcil estimar el temps que podríem necessitar per a fer aquesta part del treball així que posarem un màxim de dues setmanes, la qual cosa és equivalent a 40 hores.

#### **5.1.5 Optimització i millora [OM]**

A partir d'aquí, les hores restants s'haurien de dedicar a la optimització i/o millora constant del producte final, ja sigui afegint noves funcionalitats o optimitzant les ja existents. En cas de deixar-ho tot enllestit abans de la data límit també ens podríem dedicar al desenvolupament d'altres productes que necessiti l'empresa, tot i que, en un principi, ja no estarien dins del marc del nostre treball de final de grau.

El temps que ens queda per a aquesta part són 87h si utilitzem les 500 hores de treball estimades anteriorment.



## 5.2 Representació gràfica

Seguidament mostrarem una taula que contindrà, de forma resumida, tots el requisits de cada tasca explicada amb anterioritat pel que fa a les seves dependències, hores de dedicació o recursos. També adjuntarem el que es denomina com diagrama de *Gantt* en el qual apareixerà la planificació del projecte al llarg del temps.

Tasca	Temps	Dependències	Recursos
GP	170h	GP-C, GP-T, GP-S, GP-I, GP-F, GP-P, GP-R	R1, R3
GP-C	24h	TP-A	R1, R3
GP-T	12h	GP-C	R1, R3
GP-S	15h	GP-T	R1, R3
GP-I	5h	GP-S	R1, R3
GP-F	80h	GP-I, MD	R1, R3
GP-P	20h	GP-F, OM	R1, R3
GP-R	14h	-	R1, R3
TP	21h	TP-A, TP-E	R1, R2, R3
TP-A	20h	-	R1
TP-E	1h	TP-A	R1, R2, R3
MD	182h	MD-G, MD-E, MD-S, MD-I	R1, R2, R3
MD-G	12h	TP	R1, R2, R3
MD-E	110h	TP	R1, R2, R3
MD-S	40h	TP	R1, R2, R3
MD-I	20h	TP	R1, R2, R3
IT	40h	MD	R1, R2, R3
OM	87h	IT	R1, R2, R3
<b>Total</b>	<b>500h</b>		

Taula 5.1: Taula resum de les tasques, durada, dependències i recursos necessaris (elaboració pròpia)

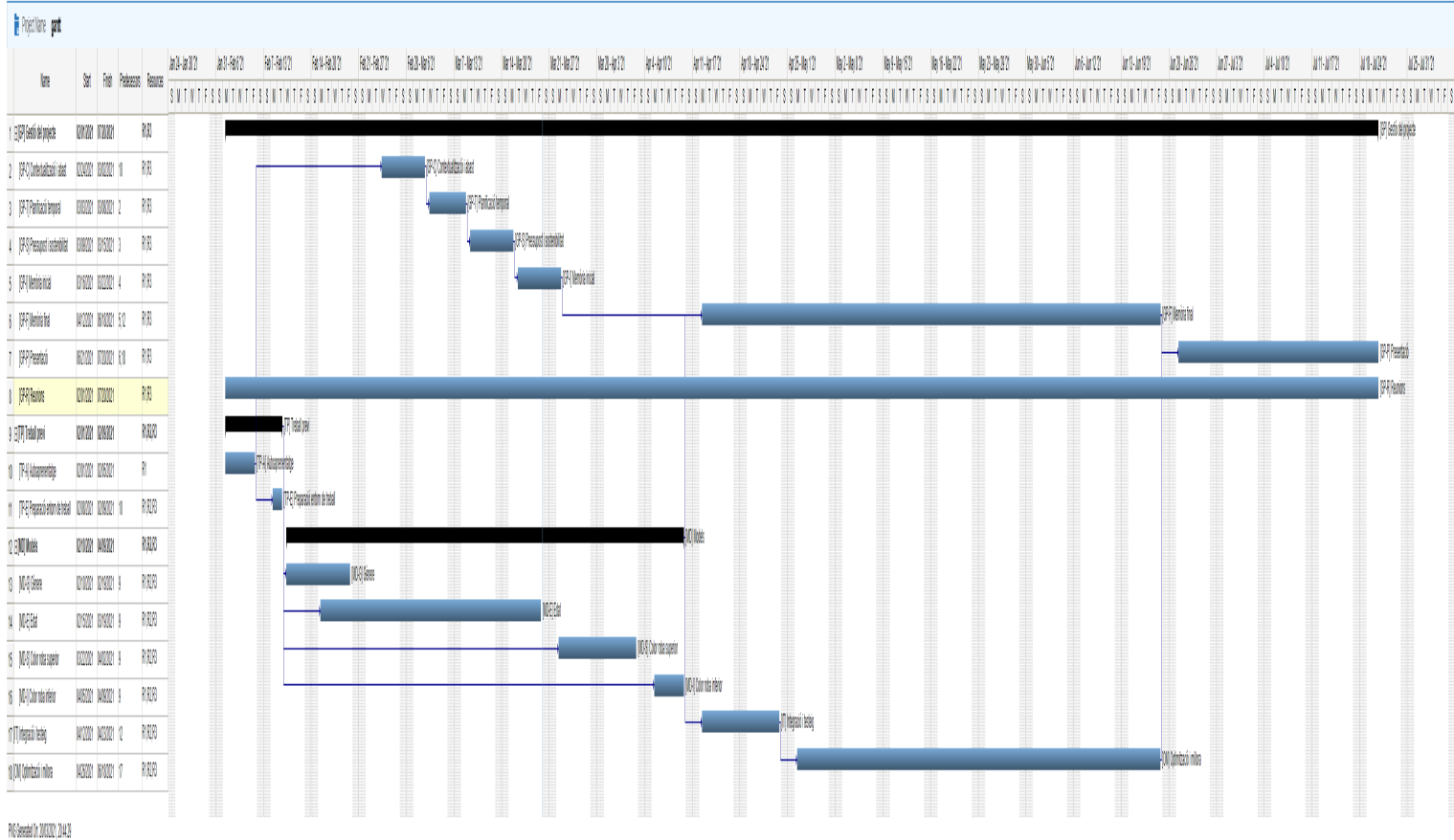


Figura 5.1: Diagrama de Gantt realitzat amb l'eina Ganttter (elaboració pròpia)

## 5.3 Recursos necessaris

Tot projecte requereix d'uns determinats recursos per arribar a ser desenvolupat. A continuació enumerarem els recursos humans i materials que són necessaris per a la correcta realització del nostre projecte.

### 5.3.1 Humans [R1]

Els recursos humans necessaris són un director de projecte, un director de producte, un tutor (encarregats de l'organització del producte) i un programador (encarregat del seu disseny i desenvolupament).

### 5.3.2 Hardware [R2]

Pel que fa als requisits *hardware* necessitem un ordinador portàtil (per poder treballar des de casa o a l'oficina amb el mateix entorn) que tingui una targeta gràfica considerablement potent amb l'objectiu d'agilitzar, el màxim possible, la computació paral·lela de l'entrenament dels models. El portàtil en qüestió ha estat cedit per la pròpia empresa i és un Lenovo Legion [45] el qual conté una gràfica força potent, la NVIDIA GTX 1660Ti [46].

També necessitem alguna màquina que contingui targetes gràfiques NVIDIA Jetson Nano [47] per a poder realitzar les proves d'eficiència i rendiment en les mateixes condicions en les quals s'executarà el producte final per als clients. Òbviament això ens ho entrega també la pròpia empresa ja que tenen un entorn de testeig per a tots els models que es desenvolupen, els quals compten amb aquestes gràfiques en particular.

### 5.3.3 Software [R3]

Els requisits *software* o programari requerits ens agradaria explicar-los amb una mica més de detall ja que, en alguns casos, pot ser molt útil per a entendre millor quines eines hem utilitzat, i com les hem combinat per a dur a terme el projecte. Per això enumerarem primer els programes utilitzats i, posteriorment, els explicarem un a un.

Per a desenvolupar els models software hem necessitat un entorn de desenvolupament de Python per executar TensorFlow per a la creació i entrenament dels models. Aquest entorn hem decidit que fos Anaconda. També serà necessari alguna mena de control de versions com Git i GitLab. Finalment usarem Slack com a eina col·laborativa amb l'empresa i Overleaf com a editor online per a documents en LaTeX.

Tots aquests programes són o tenen versions gratuïtes i, per tant, no incrementen els costos del projecte, sobre els quals parlarem més endavant en aquesta memòria (secció 6).

#### 5.3.3.1 Python

Aquest serà el llenguatge de programació utilitzat al llarg del desenvolupament de l'entrenament i testeig dels models. Hem escollit aquest llenguatge [48] degut a la gran quantitat de llibreries i utilitats que estan presents i que són de gran ajuda per a l'entrenament de Xarxes Neuronals. Pandas [49], Matplotlib [50], NumPy [51] o TensorFlow [52] són algunes d'aquestes llibreries tan utilitzades per la comunitat científica.

#### 5.3.3.2 Anaconda

Per a gestionar els paquets i versions de Python podríem haver utilitzat simplement l'eina integrada que porta anomenada Pip [53] amb la qual es poden instal·lar nous paquets d'un conjunt de repositoris autoritzats. No obstant, des de l'empresa, ens van recomanar utilitzar Anaconda [54], un gestor de paquets i d'entorns per a Python. Amb aquesta eina resulta extremadament senzill organitzar les llibreries necessàries en diferents entorns, un per cada aplicació.

#### 5.3.3.3 Tensorflow

TensorFlow és un *framework* de codi obert completament orientat al desenvolupament d'aplicacions basades en *Machine Learning*. Inclou un conjunt de llibreries molt importants i àmpliament conegudes com Keras [55] o Scikit-learn [56] que simplifiquen en gran mesura el desenvolupament de models basats en IA. També inclou un conjunt de *datasets* preparats per a ser descarregats i provats.

#### 5.3.3.4 GitLab (Git)

Com en tot projecte que precisa de codi per a ser desenvolupat, no podia faltar una sistema de control de versions (Git [57]) i un servidor web on penjar els nostres repositoris. L'empresa ja té un servidor GitLab [58] i, per tant, serà el lloc on anirem guardant tots els nostres avenços.

### 5.3.3.5 Slack

Com a eina de missatgeria instantània per a mantenir les comunicacions amb l'empresa, i en concret amb el nostre equip, hem utilitzat Slack [59]. Això és una eina col·laborativa per a empreses (majoritàriament) que permet la connexió entre diferents persones mitjançant un o diversos canals de missatgeria, utilitzats cada un d'ells segons convingui a l'empresa o client.

### 5.3.3.6 Overleaf

Finalment necessitàvem algun programa que ens permetés realitzar el redactat, en LaTeX, del GEP i memòria final del TFG. Aquest *software* hem decidit que fos Overleaf [60], un editor online i col·laboratiu per a documents en LaTeX amb el qual ja tenim força experiència d'anteriors treballs.

## 5.4 Gestió del risc

Com tot projecte existeixen riscos i dificultats que podem no haver tingut en compte en un primer moment. Com que aquests imprevistos poden sorgir, hem decidit curar-nos en salut i deixar la possibilitat d'utilitzar, també, alguns caps de setmana o dies festius puntuals per tal de dur a terme tasques com la pròpia redacció de la memòria o la solució dels imprevistos en cas que sigui necessari.

La manca de temps és una de les adversitats més comunes en projectes grans i ambiciosos com aquest i, consegüentment, creiem molt oportú deixar un generós coixí d'hores de treball com a marge per no haver d'agafar el temps d'altres tasques si es produeixen aquestes circumstàncies.

Una altra dificultat que es podria donar és el baix rendiment de les solucions proposades un cop testejades en el sistema de l'empresa. Això implicaria una quantitat substancial d'hores dedicades a la optimització i millora del producte final. D'entrada les 87h de les quals disposem per a realitzar-ho haurien de ser suficients, no obstant, sempre es pot donar la lamentable situació en què haguem de refer de zero alguns dels models per aconseguir el rendiment esperat. És aquí on també entraria en acció el marge d'hores que comentàvem.

Finalment hem decidit deixar un total de 100 hores per a poder cobrir qualsevol d'aquests imprevistos (o de nous) que hi puguin aparèixer. Aquestes, sumades a les 500 hores que ja teníem assignades, donarien un total de, en el pitjor dels casos, 600 hores de dedicació al desenvolupament d'aquest TFG.

# Gestió econòmica

---

*ATENCIÓ: aquest capítol ha patit alguns canvis respecte del GEP i, per tant, algunes parts podrien no correspondre amb la versió final del treball. Podeu trobar més detalls sobre els canvis realitzats a la secció 8.*

Un cop realitzada la planificació temporal del projecte, és hora d'identificar i d'estimar els costos necessaris que comportarà el seu desenvolupament amb l'objectiu de determinar, també, la seva viabilitat econòmica. A continuació realitzarem un estudi detallat de tots i cada un dels costos que implica el nostre projecte dividits en subcategories.

## 6.1 Costos de personal

Començarem identificant el cost per persona que comporta cada un dels perfils que intervenen directament en el projecte. Abans de començar val a dir que, com ja hem comentat amb anterioritat en aquesta memòria, el projecte pertany a la modalitat tipus B i, per tant, l'autor del projecte està fent, a l'hora, pràctiques en una empresa. El preu mínim a pagar per les empreses de la borsa de treball establert per la facultat d'Enginyeria Informàtica a estudiants de grau és de 9 € l'hora.

### 6.1.1 Programador

El rol de programador serà adquirit, exclusivament, per l'autor d'aquest projecte ja que serà l'encarregat de desenvolupar i programar la solució esperada. Com ja hem dit, el sou brut serà de 9 € l'hora. Aquest sou va molt encaminat a la mitjana del sou brut que

cobren els programadors júnior (pocs anys d'experiència professional) segons un parell de fonts consultades [61] [62].

### 6.1.2 Cap del projecte

Aquest segon rol pertany a un total de 5 persones: director i tutor del projecte, *product manager*, el cap d'intel·ligència artificial i el mateix autor del treball. Per calcular el sou per hora, el qual no podem saber per motius de protecció de dades, ens hem basat en la mitjana del sou que cobra un programador sènior segons les mateixes fonts citades anteriorment. Aquest sou és d'aproximadament 35000 € l'any, la qual cosa implica un total de 18 €/h.

El director del projecte i la product manager inverteixen un conjunt d'hores a fer reunions de seguiment com ja vam detallar en l'apartat de planificació temporal. El tutor, dedica el seu temps en corregir o assessorar l'alumne amb el seu treball de grau. Finalment, hem cregut oportú afegir, també, el cap de IA ja que és qui inverteix un total d'hores més gran en ajudar l'autor a complir els seus objectius i, per consegüent, no el podíem passar per alt.

### 6.1.3 Total

Seguidament adjuntem una taula [6.1] en la qual resumim els sous definits amb anterioritat i calculem la part proporcional a la Seguretat Social de cada perfil per extreure la retribució total de cada un d'ells.

Perfil	Sou brut	Seguretat Social	Retribució
Programador	9 €/h	2.7 €/h	11.7 €/h
Cap de projecte	18 €/h	5.4 €/h	23.4 €/h

Taula 6.1: Taula resum amb el sou brut i la retribució per cada perfil (elaboració pròpia)

## 6.2 Costos per activitat

Un cop definits els sous de cada perfil ja podem fer una estimació dels costos per activitat segons les definides en el diagrama de *Gantt* que ja vam treballar prèviament. Per fer-ho ens ajudarem d'una taula [6.2] on podrem observar, de manera molt visual i esquemàtica, la tasca sobre la qual estem parlant i el seu cost total que implicarà segons les hores dedicades per cada perfil. Tots els costos estan calculats tenint en compte el sobrecost que suposa la Seguretat Social.

Tasca	Cap de projecte	Programador	Cost
GP	170h	-	3978 €
GP-C	24h	-	561.6 €
GP-T	12h	-	280.8 €
GP-S	15h	-	351 €
GP-I	5h	-	117 €
GP-F	80h	-	1872 €
GP-P	20h	-	468 €
GP-R	14h	-	327.6 €
TP	-	21h	245.7 €
TP-A	-	20h	234 €
TP-E	-	1h	11.7 €
MD	22h	160h	2386.8 €
MD-G	-	12h	140.4 €
MD-E	15h	95h	1462.5 €
MD-S	5h	35h	526.5 €
MD-I	2h	18h	257.4 €
IT	15h	25h	643.5 €
OM	25h	62h	1310.4 €
<b>Total</b>	<b>232h</b>	<b>268h</b>	<b>8564.4 €</b>

Taula 6.2: Taula resum amb l'estimació del cost per activitat i el total (elaboració pròpia)

## 6.3 Costos genèrics

En aquest subapartat utilitzarem novament unes taules per representar els costos genèrics que són presents al llarg del projecte com recursos *hardware* o *software* entre d'altres i calcularem les amortitzacions corresponents a cada un d'ells.

### 6.3.1 Software

Com vam comentar en la gestió temporal els recursos *software* que necessitem no impliquen quasi bé una pujada del pressupost degut a la gratuïtat de la gran majoria del programari utilitzat. En la taula [6.3] podem observar aquest fet.

### 6.3.2 Hardware

El *hardware*, a diferència del *software*, sí que implica una petita inversió en l'obtenció d'un portàtil de gamma mitja-alta per poder treballar amb la fluïdesa adequada. L'ordinador



Software	Cost al mes	Mesos	Cost total
Overleaf	0€	5	0€
Slack	0€	5	0€
Python	0€	5	0€
Anaconda	0€	5	0€
Jupyter Notebook	0€	5	0€
Tensorflow	0€	5	0€
GIT	0€	5	0€
GitHub/GitLab	0€	5	0€
Gantter	5€	5	25€
<b>Total</b>	<b>5€</b>	<b>5</b>	<b>25€</b>

Taula 6.3: Taula resum amb el cost total del *software* utilitzat (elaboració pròpia)

cedit per l'empresa va ser un Lenovo Legion que, segons aquest web [45], el seu preu actual en el mercat és d'aproximadament 1200€.

L'amortització d'aquest producte al llarg del temps, tenint en compte una mitjana de 4 anys de vida útil, és de 25€ al mes ( $1200 \div 4 \div 12$ ).

### 6.3.3 Internet

Per altra banda tenim els costos relacionats amb la factura d'internet. L'adquisició de 600Mbs simètrics de fibra òptica ha estat, i és, un *must have* per al projecte que volem realitzar. La gran quantitat de dades i d'imatges que necessitem descarregar o pujar a internet, fins a 24GB en alguns casos, converteixen la connexió a internet en un dels pilars fonamentals del projecte a l'hora d'agilitzar aquest procés fins a un màxim d'uns quants minuts i no haver de perdre hores i hores.

La factura d'internet és doncs de 43.50€ mensuals. Això implica un total de 217.5€ al llarg dels cinc mesos de dedicació.

### 6.3.4 Total

En aquesta taula [6.4] observem els costos totals genèrics del nostre projecte de manera gràfica. Els costos de *hardware* estan extrets de l'amortització del portàtil al llarg dels diferents mesos i no del seu cost d'adquisició.

<b>Recurs</b>	<b>Cost</b>
Software	25 €
Hardware	125 €
Internet	217.5 €
<b>Total</b>	<b>367.5 €</b>

Taula 6.4: Taula resum amb els costos genèrics totals (elaboració pròpia)

## 6.4 Contingències

En la gestió econòmica de qualsevol projecte cal tenir en compte els imprevistos o les complicacions que hi puguin sorgir. És recomanable crear una partida de contingència per estar preparats, econòmicament parlant, per afrontar qualsevol d'aquestes situacions.

Generalment s'acostumen a reservar entre un 10% i un 20% per als plans de contingència de projectes relacionats amb el *software*. Nosaltres ho deixarem en un punt mitjà: 15%.

El total del cost de contingència és obtingut de la següent manera:  $(cost\_per\_activitat + cost\_general) \cdot 0.15 = (8564.4 + 367.5) \cdot 0.15 = \mathbf{1339.79 \text{ €}}$

## 6.5 Imprevistos

Per últim cal calcular també els costos i les possibilitats de que els imprevistos esdevinguin una realitat. Per calcular els costos d'aquest plans alternatius multiplicarem la despesa que impliquen per la respectiva probabilitat de que succeeixin.

En la posterior taula [6.5] podem observar l'increment del cost que suposarien.

<b>Imprevist</b>	<b>Temps</b>	<b>Cost</b>	<b>Risc</b>	<b>Total</b>
Manca de temps	40h	468 €	25%	117 €
Resultats no esperats	60h	702 €	35%	245.7 €
<b>Total</b>	<b>100h</b>	<b>1170 €</b>	<b>-</b>	<b>362.7 €</b>

Taula 6.5: Taula resum amb els costos dels imprevistos en funció de la seva probabilitat (elaboració pròpia)

## 6.6 Cost total

Finalment calcularem el cost total que suposaria el projecte final fent el sumatori de totes les quantitats totals estimades anteriorment. La taula corresponent és la [6.6].

<b>Categoria</b>	<b>Cost</b>
Activitats	8564.4 €
Genèrics	367.5 €
Contingència	1339.8 €
Imprevistos	362.7 €
<b>Total</b>	<b>10634.4 €</b>

Taula 6.6: Taula resum amb el cost total del projecte segons cada categoria (elaboració pròpia)

## 6.7 Control de gestió

A mesura que avancem amb el nostre projecte anirem fent les revisions dels costos que siguin necessàries amb l'objectiu d'assegurar que el desviament dels ja predits en aquesta secció esdevingui el mínim possible.

Gràcies a dur això a terme podrem detectar més fàcilment si es produeixen o no desviacions en els costos prèviament estimats i a quina categoria hi apareixen. Amb tot, serem capaços de determinar si les desviacions produïdes s'han de cobrir amb els diners destinats a la gestió dels imprevistos o amb el de les contingències. En el cas extrem que se sobrepassessin tots els límits i es requerissin una quantitat molt més gran de diners que la que podem cobrir no ens quedaria més remei que replanificar el pressupost per adaptar-lo a les noves circumstàncies. Les fórmula que usarem per a fer aquest control és:  $cost\_estimat - cost\_real$ .

# Sostenibilitat

---

Un cop realitzada l'enquesta d'*Edinsost* [63] dirigida al món universitari en referència a la sostenibilitat ens hem adonat del nostre nivell real de coneixement al respecte. Tot i que les universitats, en general, cada cop fan més i més esforços per formar els seus estudiants amb una gran visió sostenible, la manca de projectes en els quals poder aplicar aquests coneixements fa que, d'alguna manera, haguem desenvolupat la part teòrica molt per sobre de la pràctica.

Considerem que tenim força nocions sobre la importància de la sostenibilitat en el món de les TIC. No obstant, aquest és el primer projecte de gran magnitud en el qual hem d'aplicar tots aquests coneixement adquirits al llarg del grau en Enginyeria Informàtica.

Dividirem aquest informe en tres grans apartats fent al·lusió a cada una de les dimensions que ha de tenir present tot projecte tecnològic: l'econòmica, l'ambiental i la social.

## 7.1 Dimensió econòmica

Un cop analitzat el pressupost total que necessitem pel desenvolupament d'aquest projecte podem determinar si el seu cost és el més ajustat possible econòmicament parlant i no fem despeses innecessàries.

Observem que els sous del personal són els que són i no poden ser reduïts ja que, entre d'altres, la pròpia facultat així els estableix per la no precarització de la professió. Així mateix, les hores que dedicarem a cada activitat creiem que han estat força ben assignades i no quedaran marges de temps gaire notables. Si, per qualsevol motiu, acabéssim

amb temps alguna de les activitats, sempre és pot invertir el temps restant en l'avenç d'alguna altra que pugui comportar més temps del previst.

Pel que fa als recursos materials veiem que el pressupost està molt ben ajustat. En quant a *software* hem optat per utilitzar tot el programari gratuït o de codi obert que hem pogut i, com a gran despesa material, només ens caldrà un ordinador portàtil. Malgrat ser un portàtil de bones prestacions, creim que pot ser molt ben amortitzat al llarg dels anys per a qualsevol altre projecte que l'empresa necessiti desenvolupar; ja que no oblidem que el PC es propietat d'ella i pot ser utilitzat per qualsevol treballador.

## 7.2 Dimensió ambiental

En relació a la dimensió ambiental podem dir que, degut a la pandèmia, treballarem la gran majoria d'hores des de casa. Això comporta una reducció del desplaçament que hauríem de fer en cas d'haver d'anar a treballar a l'oficina els dies laborables, la qual cosa es tradueix en una reducció del consum d'un vehicle propi o, fins i tot, del transport públic.

Per altra banda comentar que, degut al tipus de feina i de projecte, tot es farà de manera digital i telemàtica. Per tant, l'ús de paper serà ínfim. Només caldrà imprimir la memòria final d'aquest document ja que, la resta d'entregues, es faran via Atenea o Racó. Els documents que hagi d'entregar-nos l'empresa també seran compartits únicament via telemàtica per correu electrònic o Slack.

## 7.3 Dimensió social

Aquest projecte no tan sols ens aportarà un gran coneixement tècnic relacionat amb el camp de la Intel·ligència Artificial, sinó que també ens brindarà moltes capacitats no tècniques com la comunicació amb l'equip de l'empresa, la capacitat d'organització i la gran capacitat d'expressió escrita i oral, entre d'altres.

Per acabar, ens agradaria afegir que aquest projecte ajudarà en gran mesura a totes aquelles empreses que contractin el producte ja que els estalviarà grans quantitats de temps i recursos realitzar qualsevol tipus d'analítica de vídeo.

# Canvis realitzats

---

En aquest capítol parlarem sobre els canvis que ha patit aquest treball respecte de la planificació inicial que vam fer per a l'entrega del GEP. Com podreu observar les dues seccions que no s'han vist involucrades en cap canvi són: contextualització (secció 2) i metodologia (secció 4).

En un primer moment, el projecte semblava que tirava a la perfecció cap endavant, els objectius s'estaven complint, fins i tot, abans de les dates previstes. El model de gènere va ser entrenat i testejat força ràpid, tenint en compte la nostra experiència, i amb resultats aparentment molt satisfactoris. La cosa va començar a complicar-se amb el model d'edat. Per moltes vegades que entrenéssim el model amb un o altre dataset, per molts paràmetres que toquéssim del TensorFlow els resultats eren força decebedors. Fins i tot, vam provar a desenvolupar un únic model amb dues sortides (gènere i edat) per veure si milloraven els resultats. Vam veure que no només no milloraven sinó que empitjoraven els de la predicció del gènere. Finalment, després d'investigar diverses solucions d'altres persones, vam donar amb un mètode que, aparentment, funciona força bé per a predir edats de persones anomenat *Ordinal Regression* [64]. Un cop fetes les proves i l'entrenament amb aquest nou mètode vam adonar-nos que, en efecte, els resultats, tot i no ser perfectes, milloraven força.

Un cop acabada aquesta part, l'empresa va considerar que era millor seguir amb la integració d'aquests dos models (*back-end* i *front-end*) abans de desenvolupar els dos restants (color roba superior i inferior) per deixar enllestida una de les dues parts del treball. Això va implicar moltes hores de desenvolupament de les analítiques en C++ [65] ja que, no només havíem de programar-les en un llenguatge no utilitzat fins al moment per a Xarxes Neuronal i models (C++) sinó que era necessari, en la mesura del possible, adaptar-se a la manera de programar i organitzar codi de l'empresa. Volíem

que el nostre sistema quedés perfectament integrat en Lanaccess i, per això, vam haver de basar-nos en solucions anteriors que ja tenien programades per tal de homogeneïtzar al màxim el codi dels repositoris.

Posteriorment vam començar amb la implementació del *front-end*, àmbit en el qual teníem relativament poca experiència degut a l'especialitat de computació. No obstant vam entendre força ràpid la mecànica. Un cop acabada aquesta part, per fi, vam poder testejar tota l'analítica en conjunt. Malauradament els resultats no s'adequaven amb els obtinguts amb anterioritat en els models. Primerament pensàvem que serien problemes de programació i, en part, sí que ho eren. Després de rectificar diversos *bugs* vam tornar a provar i, tot i millorar lleugerament els resultats, no acabaven de ser els esperats. La conclusió que en podem extreure és que, degut a la falta d'experiència, no vam pensar que les cares amb les quals entrenaria el nostre model (*dataset*) no tindrien perquè representar la majoria de cares en un flux de vídeo on podien aparèixer amb diferents tonalitats de colors o amb diferents inclinacions.

Des de llavors s'ha treballat en realitzar diverses millores o solucions per als models (versió 2.0), així com començar la resta de models. També, ara que tenim molts més coneixements, s'ha intentat entrenar el model d'edat per regressió (fins ara era per classificació) per veure si els resultats milloraven. No obstant, el TFG (i la memòria), per manca de temps, es paren en aquest estat del desenvolupament que podríem anomenar versió 1.0 de la analítica. Una versió en la qual tots els components funcionen adequadament malgrat la possible falta de precisió de les Xarxes Neuronals.

És per tot això comentat anteriorment que hem hagut de realitzar força canvis en la nostra planificació temporal ja que no vam mesurar correctament el temps que necessitaríem invertir en les parts “integració i testeig” (secció 4.1.4) i “optimització i millora” (secció 4.1.5); probablement degut a que tampoc sabíem ben bé en que consistia fins que l'empresa ens ho va confirmar. Fins i tot no han estat suficients les 100 hores que vam donar-nos de marge per si sorgien complicacions. També hem hagut de posposar la data de presentació del TFG a finals d'octubre: Lanaccess va demanar una pròrroga extraordinària de les pràctiques per poder-les allargar fins el dia 31 d'octubre i, vam considerar oportú allargar el nostre TFG igualment, per tal que tinguí més sentit i aconseguir anar en paral·lel. Per últim i com és d'esperar, aquests canvis provocaran, també, un increment en el pressupost final del nostre projecte ja que treballarem per a l'empresa 4 mesos més dels exposats fins al moment en la secció 5.

# Solució plantejada

---

Abans de començar és important remarcar que, degut a la falta d'experiència que tenim en aquesta disciplina i que el *software* a desenvolupar no és per a ús personal sinó per a integrar-se dins l'entorn de vídeo analítiques de Lanaccess, la gran majoria de decisions importants han estat prèviament pensades i consensuades amb tots els membres de desenvolupament i producció implicats.

Un cop aclarit això, passarem a detallar tot el procés que hi ha hagut darrere del desenvolupament de les analítiques al llarg de tots aquests mesos. Per poder explicar-ho amb la major claredat possible dividirem la secció en subapartats seguint l'estructura de directoris que hi tenim en el repositori de codi (adjuntat externament amb aquest document).

No només explicarem tota la part de codi feta per nosaltres sinó que també adjuntarem i comentarem breument les parts essencials que ja estan fetes per l'empresa però que són igualment necessàries per a l'execució i integració de la nostra analítica amb la resta de les que ja existeixen prèviament a Lanaccess.

Començarem fent una vista general del que seria l'analítica resultant, els seu flux d'execució i els resultats que n'obté; amb posterioritat, desglossarem en subapartats cada una de les parts de les quals es compona. Acabarem amb un apartat que exposi el procés de configuració de l'analítica.



## 9.1 Visió general

Com ja hem dit, la nostra feina consisteix en predir l'edat i el gènere de diferents persones en un vídeo. La solució passa per entrenar dues xarxes, una per l'extracció del gènere i l'altre per la de l'edat, aprofitant així el model de detecció de cares del qual disposa l'empresa. Posteriorment hem hagut de programar tot el *back-end* i la lògica que hi ha d'arrere de les xarxes així com el dibuixat de les metadades com a *front-end* de l'anàlítica.

Abans de començar, és necessari conèixer la nomenclatura que s'utilitza a l'empresa per a referir-se a cada una de les parts de la nostra analítica. Primerament tenim els *lavfilter*, encarregats de l'execució d'un model. Els *laai* són un altre tipus de *plugin* encarregats de dirigir els filtres del tipus anterior. Finalment, tenim els *managers* o gestors que s'encarreguen de la correcta execució, configuració i direcció de tots ells.

Seguidament adjuntem un diagrama (figura 9.1) que resumeix a la perfecció totes les parts implicades en aquest projecte i com es comuniquen entre sí. Més endavant en aquesta secció les explicarem una a una.

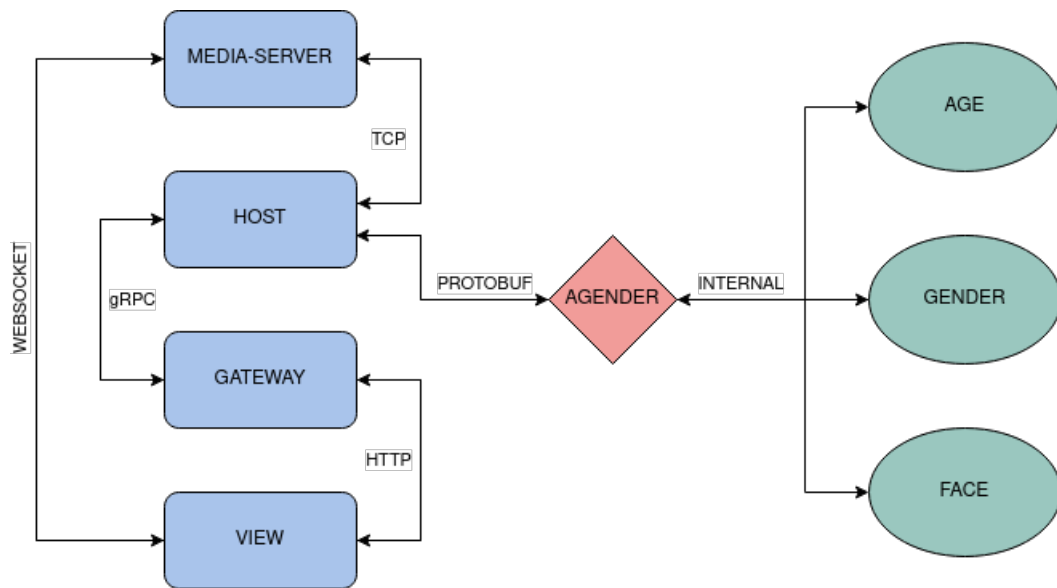


Figura 9.1: Diagrama que mostra com es realitza la interacció entre les diferents parts implicades en el funcionament de l'anàlítica (elaboració pròpia)

A l'esquerra, pintats de blau, podem veure tots els gestors encarregats del correcte funcionament de l'aplicació. Entre ells es comuniquen de formes molt variades, com s'observa en el diagrama: WebSocket [66], gRPC [67], TCP [68], HTTP [69]. Tots aquests són protocols en els quals no hi entrarem ja que no són l'objectiu d'aquest treball. De color verd hem marcat tots els *plugins* del tipus *lavfilter* (segons la nomenclatura de l'empresa)

que, bàsicament, són els filtres que carreguen i executen una Xarxa Neuronal prèviament entrenada. Aquests es comuniquen internament amb l'altre tipus de *plugin*, marcat de color vermell, *laai*. Un altre filtre que, com a diferència, gestiona quins *lavfilters* executar en cada moment i en quin ordre. Finalment, observem com la comunicació que es dona entre *host* i *agender* es realitza mitjançant ProtoBuf (Protocol Buffer) [70], una manera ràpida i modular d'estructurar les dades al nostre gust sense preocupar-nos del llenguatge de programació. Aquests protobufs són els encarregats de “transportar” les metadades de les cares detectades.

Un cop vista, de forma general, la interacció que hi ha entre les diferents parts de l'analítica toca explicar quin és el seu flux d'execució. És a dir, quines parts s'executen abans o com a prerequisits de quines altres. En la figura 9.2 podem observar l'ordre d'execució esperat. En aquesta figura no incloem la part de configuració com *gateway* o *media-server* ja que, de cara al flux d'execució, no són tan rellevants.

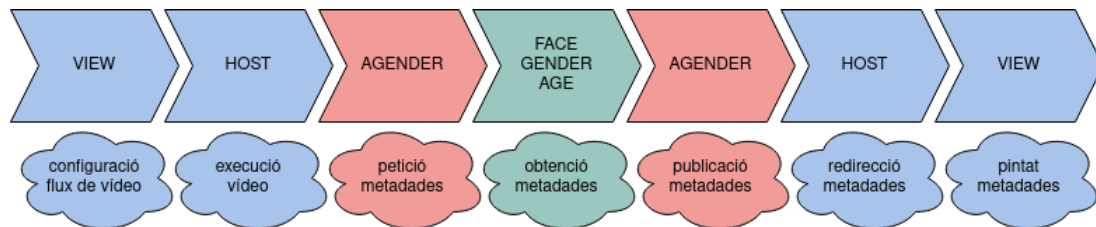


Figura 9.2: Diagrama de flux que mostra l'ordre d'execució de les diferents parts de la analítica sense tenir en compte les configuracions (elaboració pròpia)

Malgrat la figura anterior s'expliqui per sí mateixa, hem considerat oportú fer-hi un petit resum següent. Primerament en el *view* configurem quin flux de vídeo volem analitzar. Aquest és enviat al *host* perquè el llegeixi i processi frame a frame enviant-los cap al *agender*. Un cop allà, el *laai* s'encarrega d'obtenir les metadades necessàries executant tots els *lavfilters* dels quals disposa en l'ordre adequat (el qual veurem més endavant). Un cop ha rebut totes les metadades les envia, en format ProtoBuf, cap al *host*. Finalment les metadades arriben al web *view* on els resultats són pintats.

Per si encara queden alguns dubtes, a continuació enumerarem els passos seguits amb un exemple real i, entre parèntesis, marcarem quina de les parts s'encarrega de tal funció. Ho exposarem de forma semblant a un diàleg:

1. (VIEW) Configurar el *path* del vídeo desitjat: “/videos/dues\_cares.mp4”.
2. (HOST) Procedeixo a llegir el primer frame del vídeo i enviar-lo al *agender*.
3. (AGENDER) Quantes cares hi ha en la imatge?
4. (FACE) Hi ha dues cares i estan en les següents posicions.

5. (AGENDER) Quin gènere té la primera cara?
6. (GENDER) Dona.
7. (AGENDER) Quina edat té la primera cara?
8. (AGE) Entre 20 i 30 anys.
9. (AGENDER) Quin gènere té la segona cara?
10. (GENDER) Home.
11. (AGENDER) Quina edat té la segona cara?
12. (AGE) Entre 30 i 40 anys.
13. (AGENDER) Gràcies, procedeix a publicar les metadades associades amb aquest frame i passar-li al *host*.
14. (HOST) Gràcies, ara li passo al *view*.
15. (VIEW) Gràcies, procedeix a pintar les metadades en el web.

Per acabar amb aquesta secció adjuntem una captura de pantalla del que vindria a ser la sortida del gestor *view* per a un vídeo qualsevol. Gràcies a ella podem observar de quina manera hem optat per pintar les metadades per a la nostra analítica. D'ara en endavant l'anomenarem LAVA-AGENDER.

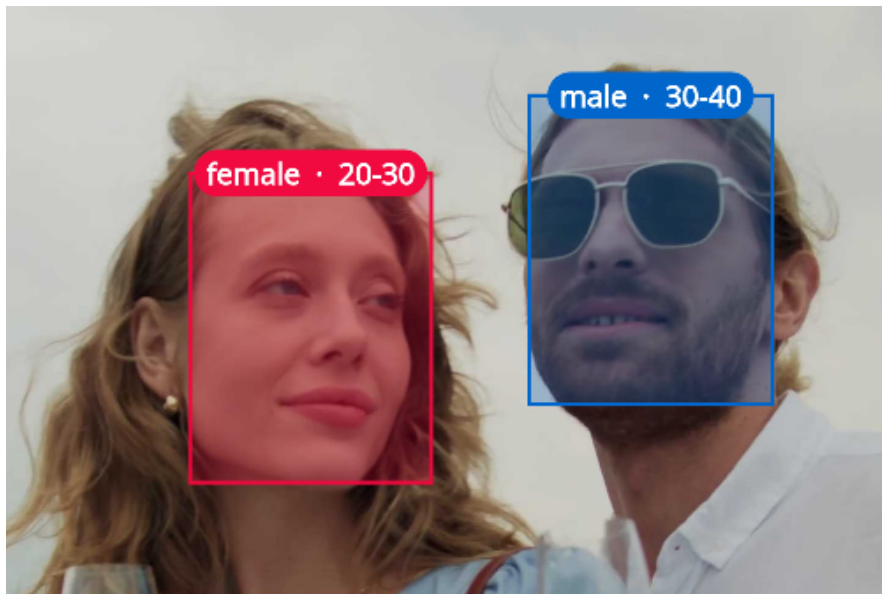


Figura 9.3: Exemple d'un frame analitzat amb la nostra analítica

## 9.2 Models

Un cop analitzat el problema i investigat sobre els models que eren necessaris ens va sorgir un gran dilema només començar: implementàvem un únic model que retornés ambdues característiques alhora (gènere i edat) o els separàvem en dos models diferents retornant una de les característiques respectivament? La primera solució era més eficient i també més simple de cara a la posterior integració, ja que requeriria de menys codi a desenvolupar. Per altra banda, la segona opció era infinitament més modular que la primera a canvi de processar dues vegades la mateixa imatge. Finalment, després d’haver-les entrenat i provat les dues versions ens vam decantar per la segona ja que no només obtenia millors resultats sinó que fomentava la modularitat i, si fos necessari, podria ser dividida en dues analítiques separades en qualsevol moment.

La solució, no obstant, requereix d’un tercer i últim model que s’encarregui de detectar les cares de les persones, ja que els altres dos s’han entrenat només amb cares. Aquest model no ha calgut que el desenvolupem nosaltres ja que l’hem reutilitzat d’un que ja utilitzava prèviament l’empresa per a altres analítiques.

### 9.2.1 Predicció del gènere

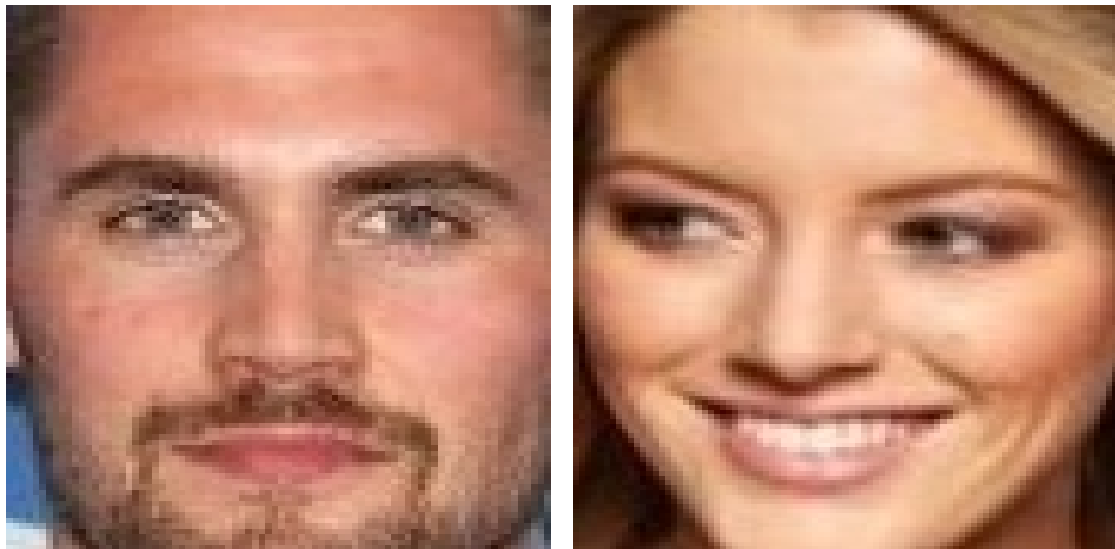
Realitzar la predicció del gènere d’una persona sembla, a priori, una tasca relativament senzilla. S’entrena un model que donada una imatge amb una persona aprengui a diferenciar quines característiques són més probables en homes i quines en dones per així acabar predient el gènere.

L’entrenament s’efectua amb un dataset de múltiples imatges amb les cares de diferents persones i, per mitjà d’un model basat en classificació, és dona com a sortida dos valors simbolitzant les probabilitats que té la imatge (segons l’algorisme) de ser un home o una dona respectivament.

#### 9.2.1.1 Dataset

El *dataset* utilitzat consta d’aproximadament 58700 imatges (85x110 píxels aproximadament) que contenen cares retallades de persones. Està subdividit en 47000 imatges per a entrenament i 11500 imatges destinades a l’avaluació dels resultats. La part d’entrenament, alhora, està dividida en 23200 cares de dona i 23800 cares d’home. Per altra banda el de validació conté 5800 cares de cada categoria.

Si voleu buscar més informació sobre el *dataset* la podeu trobar en el següent enllaç [8].



(a) Cara d'home

(b) Cara de dona

Figura 9.4: Mostra de les imatges que componen el dataset utilitzat per a la predicció del gènere (extracció del *dataset* [8])

#### 9.2.1.2 Model

El model, a petició i recomanació de l'empresa, utilitza una xarxa del tipus ResNet-18 [71] [72] [73]. Aquesta xarxa està fortament basada en les CNN, explicades en la contextualització del treball (secció 2.1.1.3), amb la diferència que introdueix els denominats blocs residuals que, en poques paraules, permeten ometre l'entrenament d'algunes capes per aconseguir, en la majoria dels casos, millors resultats. La sortida d'aquesta ResNet està connectada a una capa per classificació que utilitza *softmax*. És evident, doncs, que l'entrada del model és una cara (128x128) i la sortida són dos valors simbolitzant les probabilitats que té la imatge d'entrada de ser home o dona respectivament.

#### 9.2.2 Predicció de l'edat

Sobre el paper, predir l'edat d'una persona és força més difícil que el seu gènere ja que hi poden intervenir infinitat de factors. D'entrada ho hem enfocat homòlogament a la predicció del gènere: entrenem un model que, donada una cara de persona classifiqui el seu rang d'edat. Els rangs són: [0-10], [10-20], [20-30], [30-40], [40-50], [50-60], [60-70], [70+].

Malgrat sigui un problema que també es pot resoldre per regressió sempre i quan el dataset ho permeti, hem decidit realitzar les proves amb un model basat en classificació

per tenir els dos models una mica més homogeneïtzats entre sí.

#### 9.2.2.1 Dataset

El *dataset* utilitzat s'anomena FairFace Dataset [9]. Consta d'unes 108000 imatges de les quals 86700 són per a entrenar i les restants 11000 per a validar els resultats. En aquest cas la mida de les imatges sí és constant: 224x224 píxels.



Figura 9.5: Mostra de les imatges que componen el dataset utilitzat per a la predicció de l'edat (extracció del *dataset* [9])

#### 9.2.2.2 Model

En aquest cas el model és una CNN però que utilitza un mecanisme per a classificar lleugerament diferent del normal. Concretament utilitzem una tècnica denominada *OrdinalRegression* que vam extreure de l'interessant article [64]. En ell podreu trobar més informació al respecte.

L'entrada del model són imatges de 60x60 (seguint les indicacions de l'article) i la sortida són 7 valors que sumats representen el rang d'edat pertinent. Això es deu a que, com indica l'article, cada sortida només ens diu si l'edat és més gran o no que el rang en qüestió. És a dir, enlloc de tenir una *softmax* de 8 valors representant les probabilitats de cada rang, tenim 7 sortides i cada una d'elles amb una *softmax* que indica la probabilitat de pertànyer o no a aquell rang.

Com és un concepte força dens d'entendre ho representarem amb un exemple. Una conjunt de sortides de l'estil [1, 1, 1, 0, 0, 0, 0] significa que el rang d'edat és [30-40] ja que el sumatori de valors és 3. Un escenari amb tot zeros seria rang [0-10] i, amb tot uns, rang [70+].

### 9.2.3 Detecció de cares

Com hem dit, aquest tercer model ja ens el proporciona l'empresa. Resumidament, i com el seu nom indica, s'encarrega de detectar totes les possibles cares que hi ha en una escena de 448x448 píxels. La sortida de la xarxa és una matriu de 8232x16. Segons ens han explicat, el primer valor representa tots els segments (solapats entre sí) de la imatge d'entrada on s'hi valora la probabilitat de que hi existeixi una cara. El segon valor són el nombre d'atributs que es guarden per a cada segment: la probabilitat d'existència de cara, la capça que l'engloba i la posició dels llavis i dels ulls entre d'altres.

Com es d'esperar, a nosaltres només ens interessen cinc dels setze atributs que retorna la xarxa. Només necessitem saber quina és la certesa de que sigui una cara (probabilitat) i la posició del rectangle que l'engloba (costat esquerre, costat dret, costat superior, costat inferior).

Us estareu preguntant que passa amb les 8232 cares que retorna la xarxa. Doncs bé, la resposta és que, per obtenir el comportament esperat, necessitem utilitzar una funció externa anomenada NMS (Non-Maximum Supression) [74]. Aquesta s'encarrega de “filtrar” i eliminar totes les deteccions que siguin redundants (falsos positius) per acabar obtenint el nombre de cares real que hi hagi. Podeu trobar més informació sobre aquest funcionament a la figura 9.6 o a l'enllaç [75].

## 9.3 Plugins

Fins aquí ja tenim els tres models que seran necessaris per al correcte funcionament de l'analítica. Ara toca posar-los en comú. El primer que podríem pensar és que la part dura ja està feta i ara només queda carregar aquests models i, mentre es llegeix un flux de vídeo amb OpenCV [76], es realitzen les prediccions pertinents (crides al model per obtenir els resultats). La realitat, no obstant, no és exactament així. Si bé ho podríem

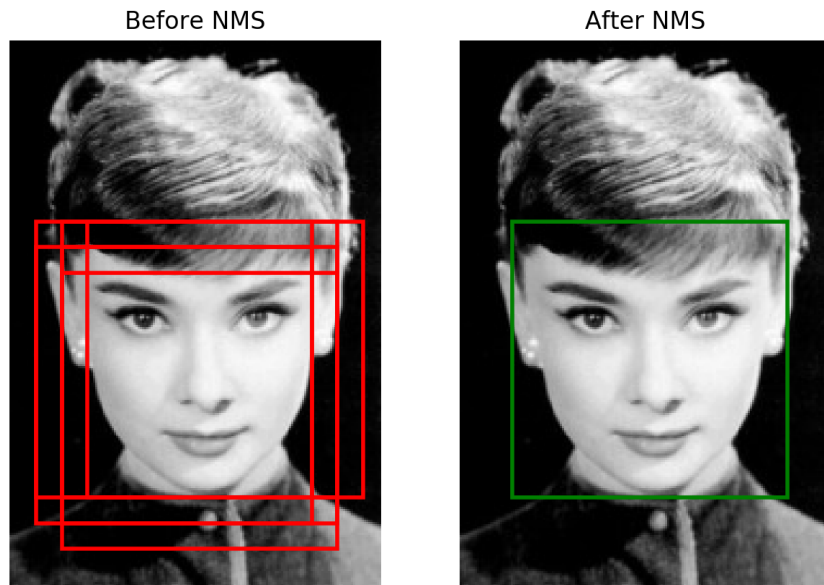


Figura 9.6: Demostració del funcionament del filtrat de cares per NMS [10]

fer servir com a entorn de proves o per a ús personal quan es tracta d'integrar-ho en una empresa ens toca seguir les pautes que ja estan marcades. Una d'aquestes pautes és l'eficiència i, per tant, executar-ho amb Python, que recordem que és un llenguatge interpretat, no seria la millor manera per fer-ho. A més, com s'executarà en càmeres amb gràfiques no gaire potents, no podem programar-ho de forma seqüencial, ja que no volem que s'interrompi el flux de vídeo a l'espera dels resultats.

La feina a realitzar ara és la programació en C++ [65] (un dels llenguatges més eficients) dels models. És a dir, el *back-end* encarregat de carregar els models i executar-los per obtenir els resultats esperats. A aquests programes els anomenem filtres i serveixen com a mòduls o plugins per al LAVA (LAnaccess Video Analytics).

### 9.3.1 Gender

Filtre que s'encarrega de l'execució de la xarxa neuronal de gènere. Rep una imatge, la qual s'espera que sigui una única cara, i retorna un valor binari representant el gènere predit. El seu resultat li envia al *laai* del qual depèn.



### 9.3.2 Age

Filtre que s'encarrega de l'execució de la xarxa neuronal d'edat. Rep una imatge, la qual s'espera que sigui una única cara, i retorna un valor numèric (entre 0 i 7) que representa en quin rang d'edat es troba la persona. El seu resultat li envia al *laai* del qual depèn.

### 9.3.3 Face

Filtre que s'encarrega de l'execució de la xarxa neuronal de detecció de cares. Rep una imatge, la qual s'espera que sigui una frame del vídeo, i retorna les cares i les coordenades on les ha trobat. El seu resultat li envia al *laai* del qual depèn.

### 9.3.4 Agender

Aquest filtre es complica notòriament respecte dels altres. Fins ara tenim tres xarxes que retornen tot el que ens interessa (cares, gènere i edat). No obstant aquestes xarxes s'han de posar en comú per treballar a l'uníson. Aquesta és la funció d'aquest filtre de la categoria *laai*: dirigir l'execució de les demés xarxes i posar en comú els seus resultats.

Per a que es pugui entendre millor posarem per exemple el nostre cas ordenant els passos que hem de seguir per tal d'obtenir els resultats esperats. Primerament necessitem detectar totes les cares d'una imatge, i seguidament, per cada cara volem extreure el seu gènere i edat. És evident que no podem executar aquests últims dos filtres directament amb la imatge d'entrada sinó que necessitem un pas entremig (detecció de cares). És aquí on entra en joc el *laai* per dirigir quan executar cada una i com combinar els seus resultats.

## 9.4 Gestors

Aquesta és l'última part de l'engranatge. Sense ells el funcionament dels *plugins* no es podria donar degut a que, en conjunt, s'encarreguen de llegir el flux de vídeo desitjat, enviar frame a frame les imatges a les xarxes (*plugins*) i pintar els resultats obtinguts sobre el vídeo d'entrada.

Aquesta és la part que ja venia feta per l'empresa i la qual està preparada per funcionar amb els filtres (mòduls) que hem desenvolupat prèviament a la secció 9.3. Degut a això no hem hagut de tocar pràcticament res a excepció de dues coses que esmentarem a continuació.

### 9.4.1 Host

Aquesta és la última peça que forma el que vindria a ser el *back-end*. Per consegüent, està programat en C++ i la seva funció és fer de “director d’orquestra” dels plugins. Bàsicament s’encarrega de llegir i enviar cada un dels frames d’un vídeo a les xarxes que tingui prèviament carregades (segons un arxiu de configuració que les especifica) i imprimir els resultats obtinguts per pantalla (format text) si el nivell de verbositat ho permet. També serveix per enviar-li al *media-server* les metadades rebudes per part del *agender*.

### 9.4.2 Gateway

Aquest altre gestor s’encarrega d’una funció molt més concreta però importantíssima. El seu objectiu és informar el *view* quin canal de vídeo ha d’analitzar i quines són les configuracions de les analítiques que haurà d’utilitzar el *host*. Les configuracions depenen de cada analítica. En el nostre cas s’envia el seu cost, valor que representa quant complexa és una analítica d’executar, i el que denominem com a *detection delta* que representa el temps, en mil·lisegons, que ha de passar entre dues execucions consecutives de les analítiques. Dit d’una altra manera, fa referència a la freqüència o *framerate* amb què l’analítica serà executada.

El llenguatge de programació amb el qual està programat aquest gestor és Rust [77]. A diferència de l’anterior 9.4.1 sí que hem hagut d’afegir la part de les configuracions que li pertocuen a la nostra analítica.

### 9.4.3 Media Server

El *media server* també comparteix el mateix llenguatge (Rust), no obstant la funció que desenvolupa és diferent. Aquest s’encarrega de rebre les metadades, en format Protobuf, del *host* per enviar-li al *view* per a pintar (format WebSocket). En aquest cas no ha calgut programar absolutament res ja que les funcionalitats són universals per a tots els *plugins*.

### 9.4.4 View

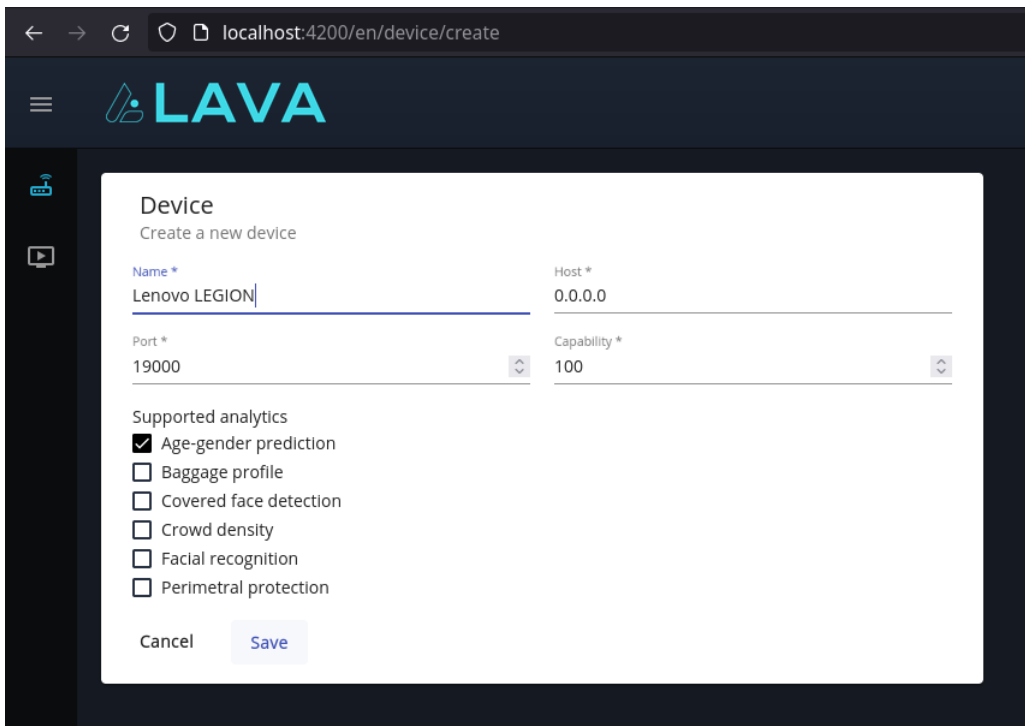
Per últim passem al que es podria definir com la part de *front-end* del desenvolupament. Per mitjà de la comunicació per WebSocket amb el *media server*, el *view* està constantment rebent el flux de vídeo i les seves metadades. La seva funció és pintar-les en el servei web.

Com és lògic, aquesta part sí que l'hem hagut de tocar per tal de pintar les cares i les prediccions de gènere i edat al nostre gust. El llenguatge de programació utilitzat per a la seva programació ha estat TypeScript [78].

## 9.5 Configuració

A continuació parlarem sobre la manera en la qual configurem el LAVA per a poder executar la nostra analítica en local.

Un cop compilat tot el codi necessari i posat en marxa tots els gestors ens tocarà configurar inicialment el *view* per aconseguir que es comporti com volem. Primerament és el torn de configurar els dispositius encarregats de l'execució de les analítiques. Normalment els dispositius seran les càmeres de videovigilància, no obstant, per a l'execució en local, serà el nostre PC. En la figura 9.7 mostrem com ho hem configurat nosaltres.



The screenshot shows the LAVA web interface in a browser window at localhost:4200/en/device/create. The interface has a dark blue header with the LAVA logo and a sidebar with icons for device management. The main content area is a white form titled 'Device' with the subtitle 'Create a new device'. The form contains the following fields and options:

- Name \***: Text input with 'Lenovo LEGION' entered.
- Host \***: Text input with '0.0.0.0' entered.
- Port \***: Text input with '19000' entered.
- Capability \***: Dropdown menu with '100' selected.
- Supported analytics**: A list of checkboxes:
  - ☒ Age-gender prediction
  - ☐ Baggage profile
  - ☐ Covered face detection
  - ☐ Crowd density
  - ☐ Facial recognition
  - ☐ Perimetral protection
- Buttons**: 'Cancel' and 'Save' buttons at the bottom.

Figura 9.7: Exemple de configuració d'un dispositiu en el LAVA

En referència a la imatge anterior, és important esmentar que els últims tres camps són de gran importància. Els dos primers (*host* i *port*) són absolutament necessaris per tal que la intercomunicació entre el *view* i el *gateway* és doni amb èxit. Pel que fa al camp de *capability* és important que es posi un valor igual o superior a la complexitat de

l'analítica, que acostuma a ser 20. Si no el *view* interpretarà que l'analítica que es vol executar supera la capacitat del dispositiu en què s'executa i, per tant, no ho permetrà.

El següent pas serà configurar el flux de vídeo que volem analitzar. Per fer-ho s'haurà d'omplir una plantilla en la qual especificarem el nom que li volem donar, la direcció de la càmera a la qual ens volem connectar així com l'usuari i contrasenya requerits entre d'altres. En el nostre cas, figura 9.8, hem optat directament per posar la ruta a un fitxer de vídeo local i, per consegüent, l'usuari i contrasenya no són necessaris.

**New video source**  
Create a new video source and configure its analytics

**01 Step 01 Source**

Name \*  
Agender TEST

Url \*  
file:///home/arnau/ai-host/couple-mid.mp4

Username

Password

Camera id \*  
0

Remote unit name \*  
0

☐ Thermal camera

**Analytics**

☒ Age-gender prediction ☐ Baggage profile ☐ Covered face detection ☐ Crowd density ☐ Facial recognition

Figura 9.8: Exemple de configuració d'un flux de vídeo en el LAVA

Fins aquí ja tindríem tot el sistema LAVA operatiu, ben configurat i funcionant correctament. No obstant, no seríem capaços de veure el vídeo resultant enlloc si no activéssim primer l'opció de publicar el vídeo. Amb aquesta opció marcada el gestor *media-server* entra en acció degut a que, com ja hem explicat, s'encarrega de redirigir les metadades que li arriben al *host* cap al *view* perquè les pugui pintar.

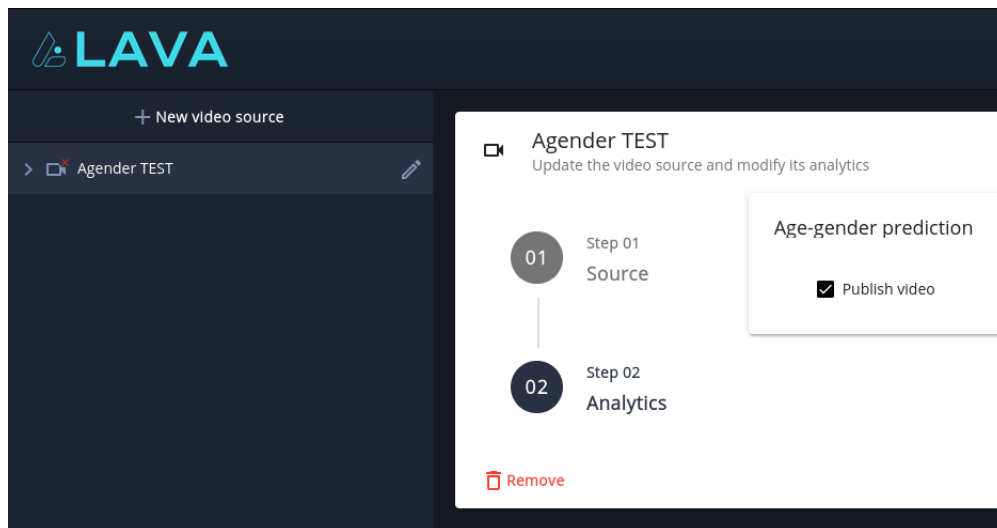


Figura 9.9: Opció que permet publicar el vídeo resultant en el LAVA

# Treball futur

---

Un cop definida i explicada la solució a la qual hem arribat (secció 9) per resoldre el problema toca adonar-se de totes aquelles coses que es podrien millorar o afegir en un futur. Ho desglossarem en dues parts: la millora de l'analítica que acabem de crear i el desenvolupament de la nova analítica que, per falta de temps, no vam arribar a fer; com s'esmenta en la secció de canvis realitzats (secció 8).

## 10.1 Millorar LAVA-AGENDER

En primer moment podríem pensar que aquesta analítica ja està acabada però, com tot *software*, sempre està subjecte a canvis, correccions o, fins i tot, millores. Ara, que portem més temps d'experiència que quan vam començar a desenvolupar-la, és bon moment per a fer balanç i definir quin seria el treball futur a realitzar amb aquesta analítica.

El més senzill i raonable seria millorar, en la mesura del possible, els models que utilitzem per a fer les prediccions. Tot i que les versions que tenim actualment ja han estat força millorades al llarg del temps, al tractar-se de Xarxes Neuronals sempre poden ser millors: canviant els *datasets*, ajustant alguns paràmetres del TensorFlow o, simplement, tornar a entrenar la xarxa són tècniques que sempre hem de tenir presents a l'hora de millorar la fiabilitat de qualsevol model.

També podem seguir diferents estratègies i provar si esdevenen millors que les originals. En el nostre cas volem provar si ajuntant les dues sortides (gènere i edat) en un sol model, ara que tenim molta més experiència que quan vam començar, fa que la xarxa predigui

millor. Malgrat en un inici no funcionés adequadament, no és descabellat pensar això ja que és una manera de tenir més informació sobre les persones. Fins ara no es té en compte el gènere a l'hora de donar l'edat, no obstant, creiem que es molt important ja que, en funció d'això, les característiques en què s'haurà de fixar la xarxa són substancialment diferents. Per exemple, en el cas de ser un home en la barba o en els llavis i en el de dona en el cabell. Òbviament no té perquè ser sempre així, però en la gran majoria del casos hauria de funcionar.

Adicionalment ens agradaria tornar a provar l'entrenament del model d'edat per regressió. Creiem que és el més lògic i, si aconseguíssim bons resultats, de cara a retornar el rang d'edat en el qual pertany només caldria agafar el valor i ficar-lo dins d'un rang de més/menys cinc anys, per posar un exemple.

Finalment, si cap dels intents de millora anteriors funcionessin, caldria tirar de l'últim recurs que, fins a dia d'avui, se'ns ha acudit. Consistiria en entrenar els models amb el cos sencer de les persones enlloc de només cares. D'aquesta manera aconseguiríem tenir molta més informació per fer les prediccions, com per exemple la manera de caminar de les persones o la forma de vestir. L'inconvenient d'aquest mètode és que ens obligaria a canviar dràsticament gran part de l'analítica al ja no tractar-se de cares. És per això que deixem aquest mètode com a última opció.

## 10.2 Desenvolupar LAVA-COLORS

Per altra banda tenim el desenvolupament dels models de detecció del color de roba que havíem inclòs com a planificació inicial. Ara que tenim més experiència, creiem que ens resultaria més senzill entrenar amb bons resultats aquests dos models restants.

Per a aconseguir-ho necessitaríem un model de persones a cos complet que, casualment, també està desenvolupat per l'empresa. Després agafaríem la peça de roba superior i inferior de manera aproximada dividint el cos en quatre parts: cap (20%), part superior (30%), part inferior (40%) i peus (10%). Un cop separades les parts, les passariem per un filtre que retornés, o bé el color mitjà que hi detecta, o bé el color principal i secundari.

# Conclusions

---

Un cop cobert el desenvolupament d'aquest projecte ha arribat el moment de recapitular i extreure conclusions sobre el treball realitzat durant aquests mesos. Seguidament donarem les conclusions que creguem més oportunes i rellevants a nivell personal, d'empresa i general. Només així serem capaços de fer balanç i trobar quins han estat els èxits i dificultats que hem tingut al llarg del desenvolupament del TFG per poder fer, també, una mica d'autocrítica en cas que fos necessari.

Començarem dient que la realització d'aquest projecte ha estat, en el seu conjunt, una experiència força gratificant. Sempre és un gran repte personal endinsar-se en un món en el qual no s'hi tenen pràcticament coneixements ni experiència i, a base d'esforç i perseverança, acabar treien endavant un treball. Val a dir també que, sota el nostre punt de vista, no ha estat en absolut una feina senzilla de dur a terme. De fet ens ha resultat més difícil del que inicialment havíem previst. Aquesta dificultat afegida ha estat fortament lligada a la falta d'experiència en la disciplina i a la seva complexitat que, juntament amb les mesures de seguretat preses per l'empresa com a conseqüència de la COVID-19, han impedit que realitzéssim alguns dels objectius en el temps que havíem previst. És per això que ens hem vist en l'obligació de crear una secció en la qual parléssim, exclusivament, dels canvis que s'havien realitzat i els motius que ens duïen a fer-los, així com el per què de les prediccions que havíem calculat inicialment de manera no del tot encertada.

Parlant del tema de les restriccions sanitàries ens agradaria afegir que els cinc primers mesos de pràctiques han estat poc habituals i especialment durs degut a la falta de contacte amb l'equip de desenvolupament de Lanaccess. No obstant, aquesta mesura de treballar telemàticament ens ha aportat altres coses molt positives. Ens ha ajudat a ser més responsables i ens ha desenvolupat la capacitat d'autoaprenentatge i d'organització

així com l'eficiència a l'hora de treballar. Si haguéssim de fer una autocrítica sobre nosaltres mateixos diríem que, en molts casos, ens hem pres massa el projecte com un treball d'universitat prioritzant més el procediment que els resultats finals i no com a feina per a una empresa que busca resultats per tenir millor viabilitat econòmica.

Aquest conjunt de dificultats, errades i obstacles que ens han anat sorgint al llarg del desenvolupament, juntament amb els grans esforços que realitzàvem per a sobrepassar-los, ens han deixat un regust lleugerament agredolç a la part de l'entrenament dels models ja que considerem que la quantitat d'hores dedicades a la programació, testeig, i sobretot, millora dels diferents models, no ha anat acompanyada dels resultats finals que ens hagués agradat obtenir. Lligant amb aquest tema, ens agradaria remarcar que, a dia d'avui, amb tota l'experiència que hem adquirit en aquest temps, veiem de manera més clara quins han estat els nostres errors i com creiem que es podrien millorar. Això justifica la inserció del capítol que parla sobre els canvis futurs que s'haurien de dur a terme a la nostra analítica, ja sigui per millorar-la en fiabilitat o per estendre-la amb noves característiques.

Creiem oportú recalcar també que, en un primer moment, aquesta feina ens generava certs dubtes a l'hora de triar-la. Els camps de la Intel·ligència Artificial i del Processament de Vídeo eren conceptes completament nous per a nosaltres i, fins i tot, no ens acabaven de fer el pes tot i pertànyer a la nostra especialitat (computació). No obstant, vam decidir fer el pas i acceptar el repte. Ara podem concloure que estàvem bastant equivocats ja que, en general, ens han sorprès molt positivament ambdues branques de l'Enginyeria Informàtica.

Per acabar i a títol personal, ens encantaria destacar el gran paper que juguen les pràctiques en empresa per a un estudiant de grau. Més enllà de si el TFG es realitza o no juntament amb aquestes creiem que és un recurs, proporcionat per les universitats, que esdevé un punt de partida molt sòlid, i pràcticament necessari, de cara a la incorporació al món professional. La visió que et dona sobre el món laboral, l'experiència i els coneixements transversals que et proporciona i la posada en comú de tots els conceptes que s'han après durant tots els anys de facultat fan de les pràctiques en empresa una recomanable manera de començar a pensar en el nostre futur per trobar feines o màsters que s'adeqüin, en la mesura del possible, als nostres gustos i necessitats.



# Agraïments

---

M'agradaria començar aquesta secció donant les gràcies a la FIB i, en general, a la UPC per haver-me permès dur a terme aquestes pràctiques, així com per l'exel·lent i estricta formació acadèmica que m'han atorgat al llarg dels últims cinc anys.

En segon lloc vull agrair al Gerard Escudero per la feina que ha realitzat com a tutor del meu TFG i per haver-lo acceptat. També vull agrair-li a la meva tutora del GEP Olga Pons per tot el *feedback* i els consells que m'han ajudat a millorar i perfeccionar el meu treball.

En tercera instància donar gràcies a l'empresa Lanaccess i al seu equip directiu Fernando Guillem i Daniel di Checcho per oferir-me la possibilitat de fer les pràctiques amb ells, així com a la Laia Espuny i Víctor Bautista per haver-me recolzat i facilitat la meva entrada al món laboral.

Per últim, però no menys important, m'encantaria felicitar al Rangel Chirino i Thomas Vidal per haver tingut la paciència i l'esforç d'estar dia rere dia ajudant-me, aconsellant-me i solucionant tots els meus dubtes al llarg del desenvolupament de l'analítica així com per compartir amb mi part de la seva experiència en aquest món.

Gràcies a totes aquestes persones la meva incorporació en el món professional no només ha estat possible, sinó que a més, ha estat una experiència amena, agradable i, sobretot, molt fructífera.

# Bibliografia

---

- [1] Lanaccess. Lanaccess telecom homepage. <https://www.lanaccess.com/en/>.
- [2] SAP. What is machine learning? <https://insights.sap.com/what-is-machine-learning/>.
- [3] KDnuggets. Neural networks - an intuition. <https://www.kdnuggets.com/2019/02/neural-networks-intuition.html>.
- [4] Possibility. Imaging to perfection. <https://possibility.teledyneimaging.com/imaging-to-perfection-artificial-intelligence/>.
- [5] Milestone. Bosch forensic search plug-in. <https://www.milestonesys.com/marketplace/bosch-security-systems-b.v/bosch-forensic-search-in-centralized-search/>.
- [6] Theory of Computation. Edge computing diagram. <https://i2.wp.com/www.theoryofcomputation.co/wp-content/uploads/2018/11/diagram-edge-computing-1024x576.png?resize=1024>.
- [7] Infinity. Kanban methodology — a visual system for managing work. <https://startinfinity.com/project-management-methodologies/kanban>.
- [8] Kaggle. Gender classification dataset: Male and female image dataset. <https://www.kaggle.com/cashutosh/gender-classification-dataset>.
- [9] Github. Fairface: Face attribute dataset for balanced race, gender, and age. <https://github.com/joojs/fairface>.
- [10] Scutmash. Non maximum supression example. [https://www.scutmash.com/non\\_maximum\\_suppression/nms.png](https://www.scutmash.com/non_maximum_suppression/nms.png).

- [11] Lanaccess. Recorder by lanaccess. <https://www.lanaccess.com/en/solutions/recorder/>.
- [12] Lanaccess. Vms by lanaccess. <https://www.lanaccess.com/en/solutions/vms/>.
- [13] Lanaccess. Analytics (lava) by lanaccess. <https://www.lanaccess.com/en/solutions/analytics/>.
- [14] Britannica (B.J. Copeland). Artificial intelligence. <https://www.britannica.com/technology/artificial-intelligence>.
- [15] IBM Cloud Education. Artificial intelligence. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.
- [16] IBM Cloud Education. Machine learning. <https://www.ibm.com/cloud/learn/machine-learning>.
- [17] IBM Cloud Education. Deep learning. <https://www.ibm.com/cloud/learn/deep-learning>.
- [18] Microsoft. Deep learning vs machine learning. <https://docs.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning>.
- [19] Wikipedia. Artificial neural network. [https://en.wikipedia.org/w/index.php?title=Artificial\\_neural\\_network&oldid=1009433557](https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=1009433557).
- [20] Diego Calvo. Definición de red neuronal artificial. <https://www.diegocalvo.es/definicion-de-red-neuronal/>.
- [21] IBM Cloud Education. Neural networks. <https://www.ibm.com/cloud/learn/neural-networks>.
- [22] Educba. Types of neural networks. <https://www.educba.com/types-of-neural-networks/>.
- [23] Digital Vidya. What are the different types of neural networks? <https://www.digitalvidya.com/blog/types-of-neural-networks/>.
- [24] IBM Cloud Education. Convolutional neural networks. <https://www.ibm.com/cloud/learn/convolutional-neural-networks>.
- [25] DeepAI (Thomas Wood). Convolutional neural network. <https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network>.
- [26] Analytics India Magazine (Yugesh Verma). Comprehensive guide to different pooling layers in deep learning. <https://analyticsindiamag.com/comprehensive-guide-to-different-pooling-layers-in-deep-learning/>.

- [27] Diego Calvo. Función de activación – redes neuronales. <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>.
- [28] Great Learning. Activation functions in neural networks explained. <https://www.mygreatlearning.com/blog/activation-functions/>.
- [29] Machine Learning Mastery (Jason Brownlee). How to choose an activation function for deep learning. <https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>.
- [30] Microsoft. What is a machine learning model? <https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model>.
- [31] IBM Cloud Education. What is a data set? <https://www.ibm.com/docs/en/zos-basic-skills?topic=more-what-is-data-set>.
- [32] Towards Data Science (Adam Ross Nelson). What is a data set? <https://towardsdatascience.com/what-is-a-data-set-9c6e38d33198>.
- [33] Wikipedia. Video analytics. [https://es.wikipedia.org/wiki/Video\\_analytics](https://es.wikipedia.org/wiki/Video_analytics).
- [34] Axis. ¿qué es la analítica de vídeo? <https://www.axis.com/es-cl/learning/web-articles/video-analytics/what-are-video-analytics>.
- [35] Axis. Analítica centralizada y analítica distribuida: pros y contras. <https://www.axis.com/es-cl/learning/web-articles/video-analytics/centralized-vs-distributed-analytics>.
- [36] Wikipedia. Metadatos. <https://es.wikipedia.org/wiki/Metadatos>.
- [37] Axxonsoft. Búsqueda forense: análisis de escenas inteligente para archivos de video. [https://www.axxonsoft.com/sp/products/axxon\\_intellect\\_enterprise/video\\_analytics/forensic\\_search.php](https://www.axxonsoft.com/sp/products/axxon_intellect_enterprise/video_analytics/forensic_search.php).
- [38] IBM Cloud Education. What is edge computing? <https://www.ibm.com/cloud/what-is-edge-computing>.
- [39] Cisco. What is edge computing? <https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html>.
- [40] Axxonsoft. Forensic search in recorded video. [https://www.axxonsoft.com/intelligent\\_search.php](https://www.axxonsoft.com/intelligent_search.php).
- [41] Avigilon. Artificial intelligence and video analytics. <https://www.avigilon.com/products/ai-video-analytics>.
- [42] Programa Minerva. Metodologías de trabajo para mejorar tu productividad y tu organización. <https://www.programaminerva.es/blog/2015/10/metodologias-trabajo/>.

- [43] Teamwork. Which project management methodologies should you use? <https://www.teamwork.com/project-management-guide/project-management-methodologies/>.
- [44] Ben Aston. 9 of the most popular project management methodologies made simple. <https://thedigitalprojectmanager.com/project-management-methodologies-made-simple/>.
- [45] Amazon. Lenovo legion y540-15 gaming laptop. <https://www.amazon.com/Lenovo-i7-9750H-Processor-GTX1660Ti-81SX00NNUS/dp/B07YLJZV2G>.
- [46] Nvidia. Nvidia geforce gtx 1660 ti. <https://www.nvidia.com/es-es/geforce/graphics-cards/gtx-1660-ti/>.
- [47] Nvidia. Nvidia jetson nano. <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/>.
- [48] Python. Welcome to python. <https://www.python.org/>.
- [49] Pandas. Python data analysis library. <https://pandas.pydata.org/>.
- [50] Matplotlib. Visualization with python. <https://matplotlib.org/>.
- [51] NumPy. The fundamental package for scientific computing with python. <https://numpy.org/>.
- [52] Tensorflow. An end-to-end open source machine learning platform. <https://www.tensorflow.org/>.
- [53] Pip. The python package index. <https://pypi.org/project/pip/>.
- [54] Anaconda. The world's most popular data science platform. <https://www.anaconda.com/>.
- [55] Keras. The python deep learning api. <https://keras.io/>.
- [56] Scikit-learn. Machine learning in python. <https://scikit-learn.org/stable/index.html>.
- [57] Git. Version control system. <https://git-scm.com/>.
- [58] GitLab. What is gitlab? <https://about.gitlab.com/what-is-gitlab/>.
- [59] Slack. Slack is where the future works. <https://slack.com/intl/en-es/>.
- [60] Overleaf. The easy to use, online, collaborative latex editor. <https://www.overleaf.com/>.
- [61] Tuempleo. ¿qué sueldos cobran los informáticos en nuestro país? <https://blog.infoempleo.com/a/sueldos-de-los-informaticos-en-espana/>.

- [62] Hack a boss. Salario de programador en españa 2019, 2020 y 2021. <https://www.hackaboss.com/blog/salario-programador-espana>.
- [63] EDINSOST. Programa de sostenibilitat d'edisost. <https://is.upc.edu/es/noticias/edinsost>.
- [64] Zhenxing Niu. Ordinal regression with multiple output cnn for age estimation. [https://openaccess.thecvf.com/content\\_cvpr\\_2016/papers/Niu\\_Ordinal\\_Regression\\_With\\_CVPR\\_2016\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2016/papers/Niu_Ordinal_Regression_With_CVPR_2016_paper.pdf).
- [65] Cplusplus. Welcome to c++. <http://www.cplusplus.org/>.
- [66] Wikipedia. What is websocket. <https://en.wikipedia.org/wiki/WebSocket>.
- [67] gRPC. A high performance, open source universal rpc framework. <https://grpc.io/>.
- [68] Wikipedia. Transmission control protocol. [https://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol).
- [69] Wikipedia. Hypertext transfer protocol. [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol).
- [70] Protocol Buffers. Protocol buffers are a language-neutral, platform-neutral extensible mechanism for serializing structured data. <https://developers.google.com/protocol-buffers/>.
- [71] GeedForGeeks. Residual networks (resnet) – deep learning. <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/>.
- [72] VISO.AI. Deep residual networks (resnet, resnet50). <https://viso.ai/deep-learning/resnet-residual-neural-network/>.
- [73] Open Genus. Residual neural network (resnet). <https://iq.opengenus.org/residual-neural-networks/>.
- [74] Towards Data Science. Non-maximum suppression: Remove redundant bounding boxes in object detection. <https://towardsdatascience.com/non-maxima-suppression-139f7e00f0b5>.
- [75] Towards Data Science. How does a face detection program work? (using neural networks). <https://towardsdatascience.com/how-does-a-face-detection-program-work-using-neural-networks-17896df8e6ff>.
- [76] OpenCV. Opencv. <https://opencv.org/>.
- [77] Rust. A language empowering everyone to build reliable and efficient software. <https://www.rust-lang.org/>.
- [78] TypeScript. Typescript is javascript with syntax for types. <https://www.typescriptlang.org/>.