

Reconeixement automàtic de models de cotxe

Treball de Fi de Grau

Autor: **Jordi Izquierdo Herrera**
Director: **Joan Climent Vilaró**
Especialització: **Computació**
Curs Q2 2020-2021



Agraïments

Vull donar gràcies al meu director de projecte, Joan Climent Vilaró, per haver-me ajudat a planificar i a completar el projecte. També a la meua família per haver-me donat suport al llarg de tot el procés.

Resum

El projecte descrit en aquest document té com a objectiu fer una implementació d'un reconeixedor automàtic de models de vehicles. Es realitzarà una recerca previa, per seguidament implementar un model basat en Xarxes Neuronals amb la millor qualitat possible. Finalment es crearà una eina visual per mostrar els resultats. Aquest document detalla la recerca, el desenvolupament i els resultats d'aquesta implementació.

Resumen

El proyecto descrito en este documento tiene como objetivo hacer una implementación de un reconocedor automático de modelos de vehículos. Se realizará una investigación previa, y seguidamente se implementará un modelo basado en Redes Neuronales con la mejor calidad posible. Finalmente se creará una herramienta visual para mostrar los resultados. Este documento detalla la investigación, el desarrollo y los resultados de esta implementación.

Abstract

The project described in this document aims to implement an automatic vehicle model recognizer. A previous extensive research will be done, then an implementation of a Neural Networks-based model will be carried out with the best possible quality. Finally, a visual tool will be created to display the results. This document details the research, development, and results of this implementation.

Índex

1. Definició de l'abast i contextualització	5
1.1 Context	5
1.2 Justificació	7
1.3 Abast	8
1.4 Metodologia i rigor.....	11
2. Planificació Temporal	12
2.1 Desglossament de tasques	12
2.2 Recursos	16
2.3 Gestió de risc.....	16
3. Recerca i preparació	17
3.1 Recerca i anàlisi	17
3.2 Estratègia	18
3.3 Preparació entorn de treball.....	21
4. Implementació Principal	22
4.1 Construcció Dataset.....	22
4.2 Construcció Model Principal.....	25
4.3 Resultats i Anàlisi del Model Principal	27
5. Implementació Completa	33
5.1 Model Side i Model Rear.....	33
5.2 Combinació d'Angles	35
6. Implementació Eina Visual	39
7. Gestió econòmica i sostenibilitat	42
7.1 Gestió econòmica	42
7.2 Sostenibilitat	45
8. Conclusions	48

1. Definició de l'abast i contextualització

Durant les primeres etapes del projecte, la nostra primera tasca en aquest Treball de Fi de Grau va ser fer una definició de l'abast del projecte, definint els objectius principals i fins a on es volia arribar, fent una contextualització dels temes i els conceptes que es treballarien durant la realització del TFG.

1.1 Context

La visió artificial és un camp de la intel·ligència artificial que intenta emular i automatitzar el sistema visual humà. L'objectiu principal és que un ordinador pugui entendre el seu voltant a través d'una imatge o un vídeo. En l'actualitat, la visió artificial és un camp que avança ràpidament gràcies a la seva utilitat en la robòtica i l'automatització, com ara el vehicle autònom. La relació entre la visió artificial i la intel·ligència artificial, crea un camp de recerca extens idoni per aquest projecte.

Aquest Treball de Fi de Grau busca fer una recerca sobre el camp i aplicar-los per resoldre un problema basat en reconeixement d'objectes en una imatge. En el marc del Grau d'Enginyeria Informàtica de la Facultat d'Informàtica de Barcelona, aquest projecte està contingut en l'especialització de Computació, extraient informació d'assignatures d'Intel·ligència Artificial i Visió per Computadors.

El principal objectiu d'aquest problema no és fer un producte competitiu, sinó fer una recerca sobre els camps especificats, i aplicar-los per crear una aplicació útil, demostrant així les competències obtingudes durant els estudis del Grau d'Enginyeria Informàtica.

Definició del problema

Un cop definit els camps que volem estudiar formularem el nostre problema. El problema buscat ha de tenir prou contingut.

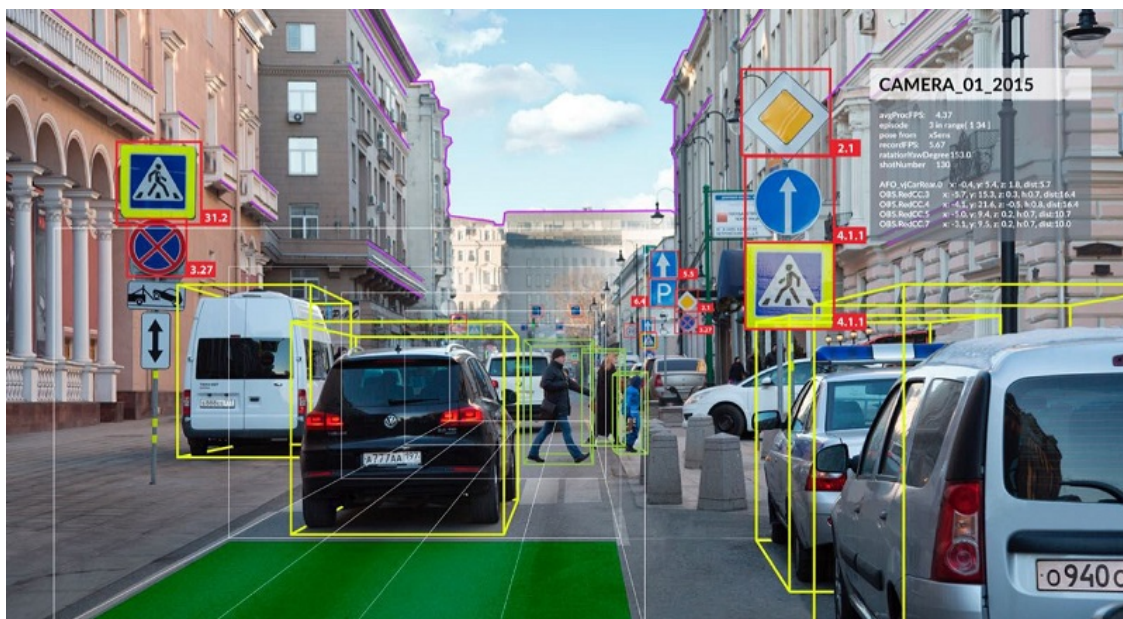
L'objectiu ha sigut implementar una aplicació de visió artificial que pugui reconèixer models de cotxe a partir d'una imatge. S'ha fet una anàlisi de diferents tècniques d'intel·ligència artificial (basades en algorismes de Xarxes Neuronals de l'aprenentatge profund) i s'ha realitzat una implementació d'una eina visual per mostrar els resultats.

Conceptes

Ja tenint definit el nostre problema, podem fer una simple introducció als conceptes que farem servir durant la realització del Treball de Fi de Grau.

Visió artificial:

La visió artificial (o visió per computadors) és una disciplina de la informàtica que tracta d'extreure i analitzar la informació d'interès continguda en una imatge o seqüència d'imatges. Tracta d'automatitzar tasques que el sistema visual humà pugui realitzar.



Imatge 1.1: Sistema de Visió artificial reconeixent elements d'un carrer

Aprentatge profund:

L'aprenentatge profund (deep learning en anglès) és una branca de l'aprenentatge automàtic que basa els seus mètodes en Xarxes Neuronals Artificials per realitzar l'aprenentatge. Aquest aprenentatge pot ser supervisat semi-supervisat o no supervisat.

Actors implicats

El resultat d'aquest projecte serà un producte que serveix d'eina a controladors de tràfic i aplicacions de recomanació de models de cotxe. L'objectiu principal d'aquest projecte és demostrar competències i fer una recerca dels conceptes implicats. El resultat no aspira a ser un producte competitiu en el mercat.

1.2 Justificació

Seguidament, justificarem si és convenient aprofitar i adaptar una solució existent o si s'ha de dissenyar-ne una de nova. Fent recerca, es van trobar aplicacions existents que resolen un problema similar eficientment que es podien adaptar al nostre problema. Tot i això, es va creure que dissenyar una aplicació des de zero era beneficiós per al nostre projecte per les següents raons:

- Amb el ràpid avanç de la intel·ligència artificial i la visió artificial una nova recerca i una anàlisi propi és beneficiós pel camp. Noves implementacions poden tindre nous resultats diferents als obtinguts a altres projectes amb característiques similars.
- L'objectiu principal d'aquest projecte és fer una recerca sobre els conceptes implicats i demostrar competències. Crear una aplicació des del principi és més beneficiós pel nostre objectiu que començar des d'un punt intermedi.

Per entrenar el nostre programa necessitem un dataset en condicions. Aquest dataset sí que ha sigut creat a partir de datasets ja existents i no començat des de zero, ja que crear datasets no aporta a la nostra recerca.

1.3 Abast

L'objectiu principal del projecte ha sigut realitzar amb èxit la implementació del reconeixement automàtic de models de cotxes. Han sigut implementades dues versions de la implementació.

Implementació principal: Imatge vista des d'un únic angle

La primera versió consisteix a fer una implementació d'un reconeixedor automàtic de models de cotxe on el cotxe és el principal element de la imatge i estigui vist des de un angle específic. S'ha decidit que aquest els vehicles de la Implementació principal estigui vist des de davant.

Implementació Completa: Imatge vista des de qualsevol angle

La segona versió implementa el reconeixedor d'una imatge on el vehicle pot ser vist des de qualsevol angle. Per tant, s'ha de tenir en consideració quina part del vehicle s'està veient per poder fer un reconeixedor satisfactori. Per implementar aquesta versió, cal primer completar la Implementació Principal, i utilitzar-lo com a base de la segona Implementació.

A part de les implementacions, els objectius secundaris del projecte han sigut fer una anàlisi del temps, eficiència i correctesa de les tècniques d'intel·ligència artificial usades, tant per la creació del model com pel reconeixement de la imatge a temps real. Finalment, s'implementarà una eina visual per mostrar els resultats.

La eina visual permetrà a un usuari introduir una imatge de un vehicle i que es retorni el model del vehicle introduït, juntament amb informació que aporti dades del procés d'identificació.

Requeriments

Perquè la nostra aplicació es consideri acabada i, per tant, el projecte realitzat amb èxit, els següents requeriments es van tenir en compte durant la estapa inicial del projecte:

- Les implementacions tenen un grau d'èxit satisfactori. Evidentment, no busquem un grau d'èxit del 100%, ja que les imatges poden variar en qualitat, focus i lluminositat.
- L'eina visual permet introduir la imatge i mostrar el resultat.

En quin grau esperavem que l'aplicació sigui satisfactòria no es podem saber en aquest punt del projecte. Durant la tasca d'aprenentatge i implementació es decidirà com de precís volem que el nostre reconeixedor sigui.

Obstacles

Es van identificar els següents obstacles per a la realització del projecte:

- El primer obstacle és la inexperiència de l'autor en treballar en projectes d'aquesta escala autònomament.
- Un altre obstacle serà el temps d'entrenament de les xarxes neuronals. Un entrenament extens de les xarxes neuronals pot durar moltes hores, si no dies, en completar-se. Una combinació eficient entre treballar en les implementacions i deixar l'ordinador processar l'entrenament és necessària.

Riscos considerats

Durant la Gestió del Projecte, es va fer una anàlisi dels possibles riscos i inconvenients que podien sorgir durant la realització del projecte. Se'n van identificar els següents riscos:

- El principal risc (i el més probable) és que l'autor no pugui completar algunes de les implementacions en el marc de temps establert. Això pot ser degut a una mala estimació del temps d'implementació o per uns mals resultats obtinguts que obliguin a tornar a fer alguna implementació.
- Un altre risc és, com ja s'ha comentat durant en els obstacles, no poder combinar el temps d'implementació amb el temps d'entrenament del model amb xarxes neuronals. Una descoordinació entre les dues tasques pot fer que el temps d'implementació sigui més llarg del previst.

Cap d'aquets riscos considerats a l'inici del projecte han sigut un problema durant la realització del projecte.

1.4 Metodologia i rigor

Aquest projecte està realitzat per una sola persona que desenvoluparà el rol de cap de projecte i programador. El temps de treball que s'esperen per aquest projecte són unes 450 hores, dictades pels 18 crèdits del TFG(25 hores/ECTS per 18 ECTS). Aquest temps és major al planificat inicialment (vegeu apartat 2: Planificació), així que hi ha hagut marge de temps per resoldre imprevistos i treballar més profundament en certes àrees del projecte segons s'ha consideri necessàries.

Per realitzar la recerca del temes tractats durant el projecte, s'ha requerit d'un aprenentatge previ. Part d'aquest aprenentatge es va realitzar durant els estudis a la Facultat d'Informàtica de Barcelona i durant un curs d'Aprenentatge Profund de JEDI durant l'estiu de 2020. Una segona part de recerca s'ha realitzat abans de començar a desenvolupar el projecte per aprendre més profundament les relacions entre la visió artificial i l'aprenentatge profund.

Per construir el nostre dataset s'ha utilitzat un dataset ja existent i s'ha expandit perquè sigui més útils per als nostres objectius. Per implementar les xarxes neuronals s'ha utilitzat el llenguatge Python amb varies llibreries, com TensorFlow. Majoritàriament, la implementació dels models ha estat realitzada a través de arxius ipynb en línia, Google Collabs sent el espai principal de treball. Altres espais i llenguatges s'utilitzaran per diferents tasques durant la implementació, com el tractament de la imatge prèvia a l'entrenament per Xarxes, o la implementació de la eina visual.

La implementació del projecte estarà dividida en 3 parts: les dues implementacions de reconeixement i la eina visual. Reunions de seguiment entre autor i director han sigut realitzades per tal d'analitzar el progrés, i resoldre imprevistos o dubtes que puguin haver sorgit durant les implementacions.

2. Planificació temporal

Per tal d'organitzar i finalitzar el projecte en el marc de temps establert, durant la Gestió inicial del Projecte es va realitzar una planificació temporal del projecte. A continuació es farà un desglossament de les tasques que s'han hagut de realitzar, la estimació temporal i de recursos comparant-los amb la realitat, i finalment, analitzarem els riscos que podrien haver aparegut durant la realització.

2.1 Desglossament de tasques

Primerament, farem una descripció de les tasques relacionades amb la gestió del projecte:

- **T1: Abast i context**
La primera tasca serà definir l'abast del nostre projecte, explicant el seu context per definir els principals objectius.
- **T2: Planificació**
La segona tasca serà fer una planificació temporal del projecte, per saber com i quan es realitzarà cada tasca.
- **T3: Pressupost i sostenibilitat**
La última tasca durant la Gestió del Projecte serà fer una anàlisi econòmic del projecte i explicar la seva sostenibilitat.

Seguidament, descriurem les tasques a fer abans de començar amb la realització tècnica del projecte:

- **T4: Recerca i Anàlisi**
Es realitzarà un aprenentatge profund dels camps d'estudi, facilitant la recerca.
- **T5: Construcció Dataset**
Es farà una construcció del dataset de Models de cotxes per entrenar les nostres xarxes. No podem començar a entrenar-les fins que el dataset estigui finalitzat.

Un cop acabat els passos previs, podem començar a treballar en el desenvolupament de l'aplicació:

- **T6: Implementació Principal**

El nostre primer objectiu és implementar un reconeixedor de models de cotxe amb el vehicle com a objecte principal de la imatge des de un sol angle de vista. Tot i que aquesta és la versió més senzilla que la Implementació Completa, és la versió en la qual l'altre implementació està basades i, per tant, requereix el mateix nombre d'hores.

- **T7: Implementació Completa**

El segon objectiu és que el reconeixedor detecti vehicles des de qualsevol punt de vista. És una versió més complexa de la primera implementació.

- **T8: Implementació Eina Visual**

Es realitzarà una Eina Visual per poder mostrar els resultats d'una forma clara i ordenada. Com que la realització d'aquesta eina no és un objectiu principal, i no necessitem que el producte final sigui competitiu en el mercat, aquesta eina serà simple i no requerirà tant de temps com les implementacions anteriors.

Finalment, analitzem els resultats del projecte, realitzant la Memòria i la Defensa.

- **T9: Memòria**

La memòria es realitzarà paral·lelament a la resta del projecte, tot i que la major dedicació a aquesta tasca serà realitzada després d'acabar les tres implementacions principals.

- **T10: Defensa**

Una vegada acabada la Memòria, es treballarà en la preparació per la defensa del TFG davant del tribunal.

Durant la gestió del projecte, es va realitzar una estimació temporal de cada tasca, relacionant-les amb les dependències per preveure la finalització del treball.

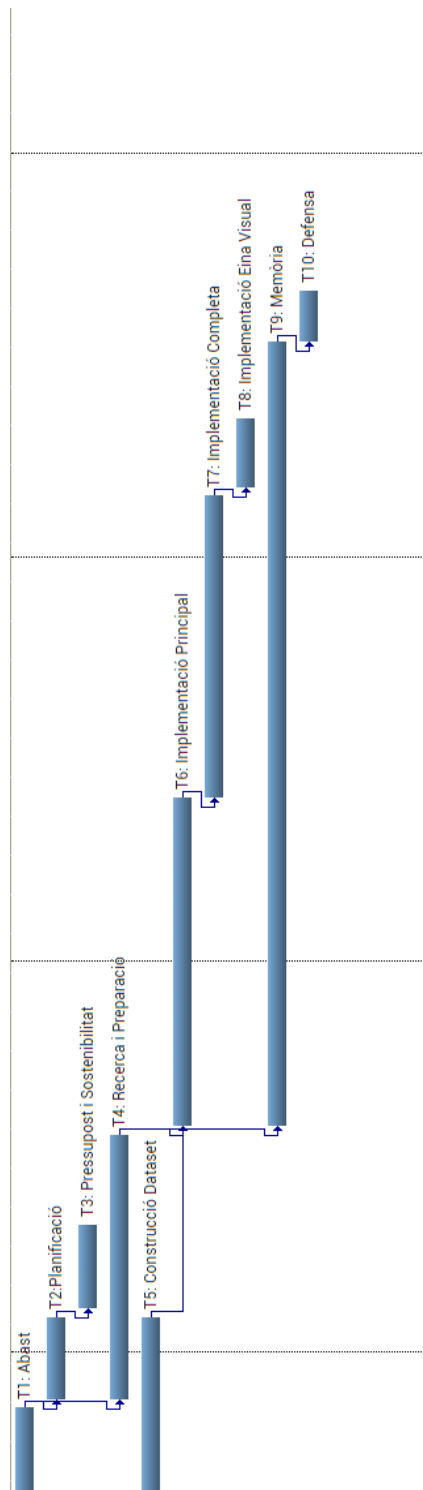
Tasca	Hores Previstes	Dependències
T1: Abast	20	{}
T2: Planificació	20	{T1}
T3: Pressupost i Sostenibilitat	20	{T2}
T4: Recerca	30	{T1}
T5: Construcció	15	{}
T6: Implementació Principal	70	{T4,T5}
T7: Implementació Completa	70	{T6}
T8: Implementació Eina Visual	15	{T7}
T9: Memòria	50	{T4}
T10: Defensa	15	{T9}

Taula 2.1: Desglossament de tasques i dependències, amb estimació de hores inicials

Les tasques i les hores planejades han sigut prou consistents amb el treball realitzat. Han requerit menys hores per la construcció del dataset del previst, ja que s'ha trobat una eina on fàcilment es pot expandir un dataset ja existent. Les hores estalviades a la construcció del dataset s'han dedicat a la implementació principal.

La planificació inicial mostrava que la feina pràctica es faria entre el Setembre i Desembre de 2020, però per raons personals i professionals, la feina pràctica s'ha fet majoritàriament entre Maig i Setembre de 2021, amb la memòria i preparació de la defensa realitzades entre la segona meitat de Setembre i Octubre de 2021.

Amb aquesta informació, podem crear el nostre Diagrama de Gantt:



Imatge 2.1: Diagrama de Gantt

2.2 Recursos

Primerament analitzem els recursos humans necessaris. Dos tipus de personal han sigut necessaris pel desenvolupament del projecte. El director del projecte, i el programador.

- El **cap de projecte** realitza les tasques relacionades amb l'organització i gestió del projecte.
- El **programador** ha de ser un enginyer informàtic capaç de resoldre els reptes tècnics que presenta el projecte.

Durant la realització del TFG, l'autor ha realitzat els dos rols, treballant les 330 hores estimades entre els dos rols.

Els recursos materials necessaris serà únicament l'ordinador que s'utilitzarà per fer les implementacions, que també s'utilitzarà per realitzar la memòria. Deguda a la pandèmia, gran part del projecte no es pot realitzar en algun espai de la universitat i s'haurà de fer a casa de l'autor amb el seu propi ordinador. L'ordinador és de mitja gamma i el seu rendiment és satisfactori per realitzar el projecte. (Processador i5-7500, GPU NVIDIA GeForce GTX 1060 3GB, RAM 8GB).

2.3 Gestió de risc

L'únic risc que podria afectar a la planificació era no poder acabar alguna de les implementacions en el marc de temps especificat. Es calculaven que cada una de les dues implementacions tenia un ~30% de probabilitats de no poder ser acabades en el temps marcat. Fins a 20 hores extres podrien ser dedicades a resoldre els problemes que poguessin sorgir per cada implementació.

Com que el temps planificat per tasques era menor al temps que l'autor dedicaria al TFG per crèdits (330 hores enfront a 450), va haver molt marge per resoldre els inconvenients que podien sorgir o inclús profunditzar en temes concrets.

3. Recerca i Preparació

3.1 Recerca i Anàlisi

Com s'ha explicat a l'apartat de Metodologia, s'ha decidit utilitzar tècniques d'Aprenentatge Profund per resoldre el nostre problema. Deguda a la falta de coneixements que es tenia abans de començar el projecte, no es va poder identificar quina seria exactament la Xarxa Neuronal Artificial que es faria servir durant el projecte.

Gràcies a l'Aprenentatge fet a l'inici del Projecte, amb la combinació del Curs Jedi de l'Estiu 2020 i una recerca pròpia, es va identificar que la Xarxa Neuronal Convolutiva (Convolutional neural network o CNN) seria la millor classe de Xarxa per utilitzar en el nostre model de Aprenentatge Profund.

Les Xarxes Neuronals Convolutives són normalment usades per analitzar imatges visuals. Constan de dues capes neuronals principals: les neurones convolutives, i les neurones de Pooling:

Durant la capa de convolució es produeix una extracció de característiques a partir de la matriu que descriu la imatge. Es convoluciona fraccions de la imatge amb filtres proporcionats com a paràmetre per intentar buscar vores, formes i característiques pròpies de la imatge.

Durant la capa de Pooling es redueix el nombre de neurones creades a la capa de convolució, combinant un clúster de neurones a una sola neurona. És necessari aquest procés per evitar un gran nombre de neurones i crear overfitting

Després de repetir les capes de convolució i Pooling diverses vegades, s'arriba a una única capa de classificació, on s'intenta classificar totes les característiques trobades en les capes anteriors a una etiqueta, seguit un Conjunt de Dades de entrenament.

3.2 Estratègia

Ara que ja tenim la Xarxa Neuronal decidida, podem passar a definir l'Estratègia d'Implementació. Com s'ha discutit a l'apartat d'Objectius, primerament volem entrenar un Model d'Aprenentatge que pugui reconèixer el model d'un Vehicle vist de front. El nostre objectiu secundari seria fer una implementació completa que pogués conèixer el model de un vehicle vist des de qualsevol angle.

En el nostre dataset elegit (Stanford Cars dataset), tenim les imatges classificades en l'angle de punt de vista: Front (Davant), Side (Costat), Rear (Darrera), Side-Front (Costat-Davant), Side-Rear (Costat Darrera):



Front (1)



Rear (2)



Side (3)



Side-Front (4)



Side-Rear (5)

Figura 3.1: Angles de visió de les imatges

Gràcies a aquesta classificació, podem dividir les imatges segons els nostres objectius. El nostre model principal intentarà classificar imatges vistes desde Front i Side-Front. Per aquest motiu, durant la fase d'entrenament solament serà entrenat amb les imatges d'aquestes perspectives.

El nostre objectiu secundari és crear una implementació que pugui reconèixer el vehicle desde qualsevol punt de vista. Hem de recordar que el Model d'aprenentatge sol pot veure una imatge en dus dimensions, però ha de poder reconèixer una imatge de tres dimensions. Per tant, ha de reconèixer que dos imatges totalment diferents com les següents formen part del mateix vehicle:



Figura 3.2: Mateix vehicle vist des de diferents punts de vista

Mentre que duess imatges de dos vehicles diferents poden resultar molt semblants si són vistos des del mateix punt de vista:



Figura 3.3: Diferents vehicles vistos des del mateix punt de vista

Per resoldre aquest problema tenim duess opcions:

- Crear un Model d'Aprenentatge amb un conjunt de dades d'entrenament molt gran que pugui distingir automàticament i classificar correctament.
- Classificar prèviament les imatges en diferents punts de vista, i crear un model d'aprenentatge diferent per cada angle.

Com que el nostre conjunt de dades no és prou gran, es decideix realitzar la segona opció.

Per realitzar aquesta estratègia, crearem dos Models d'Aprenentatge (Side i Rear) similars al Model Principal (Front), canviant el conjunt de dades d'entrenament. Les dades d'entrenament per Side seràn les imatges de Side (3), Side-Front(4) i Side-Rear(5) mentre que les imatges de Rear tindrà les imatges de Rear(2) i Side-Rear (5).

Per predir una imatge, primer caldria esbrinar el punt de vist de la imatge. Per això, crearem un quart Model d'aprenentatge que pugui reconèixer l'angle de visió. Depenent de la resposta d'aquest model, s'utilitzarà el Model Front, Model Side o Model Rear per, finalment, identificar el vehicle:

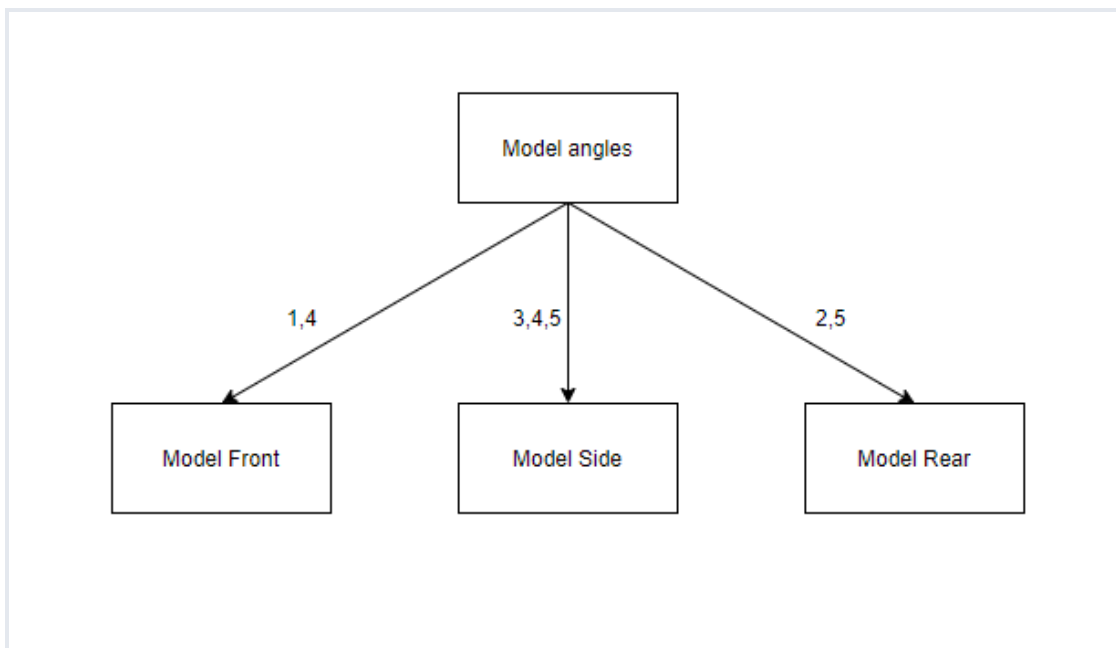


Figura 3.4: Estructura Model Complet

Com que la resposta no és binària, sino un vector de puntuacions, es farà una suma dels vectors de puntuacions de els tres models amb pesos diferents per cada un. Si el model d'angles reconeix que un vehicle està vist desde el costat amb un 75% de possibilitats i des del front amb un 25%, la resposta al problema serà una combinació del 75% / 25% del model front/model side.

3.3 Preparació Entorn de Treball

Com s'ha explicat en l'apartat de Metodologia, utilitzarem Python per programar el nostre model d'aprenentatge. Utilitzarem fitxers en format ipynb (Interactive PYthon NoteBook) per crear la implementació.

S'ha decidit treballar en l'espai Google Collabs, ja que ofereix crear fitxers ipynb amb una fàcil connexió amb Google Drive, on guardarem el nostre conjunt de dades. Per a que no hi hagi interferències amb altres projectes amb la mateixa compte de Google, es crearà un nou. Per mantenir l'ordre, es crearàn 4 fitxers de Google Collabs, 1 per cada Angle i un per la combinació d'angles i connexió amb Anvil.

Necessitarem un bon dataset per tindre un procés d'entrenament de qualitat. L'origen del nostre conjunt de dades es del Stanford Cars dataset. Aquest dataset conté imatges de multitud de marques i models de cotxes. Estan dividides en la carpeta superior per marques, després per model i finalment per any del model. Guardarem aquestes dades en el Google Drive del mateix compte de Google creat.

4. Implementació Principal

4.1 Construcció Dataset

El nostre conjunt de dades de Toyota té un total de 5132 imatges, dividides en 58 models. Aquest gràfic mostra el total d'imatges que té cada model.

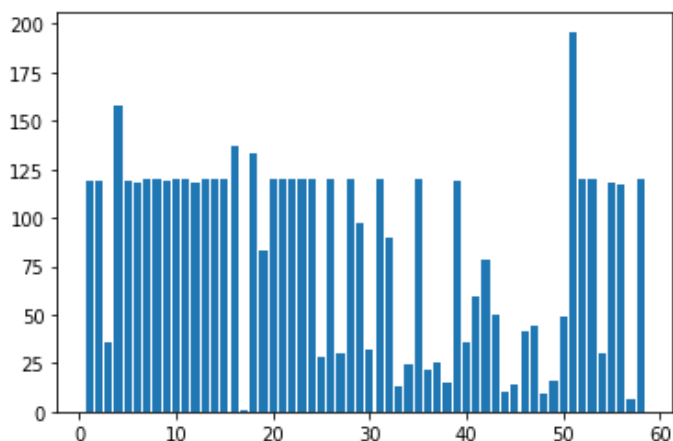


Figura 4.1: Nombre inicial d'imatges per model

Es pot observar que molts models tenen molt poques imatges. S'han revisat aquests models i s'ha observat que són models experimentals o poc comuns al carrer.

Per millorar la qualitat del nostre resultat, s'han eliminat aquests models del nostre conjunt de dades. El nostre dataset, per tant, compta amb 43 models amb una mitja de 108 imatges per model.

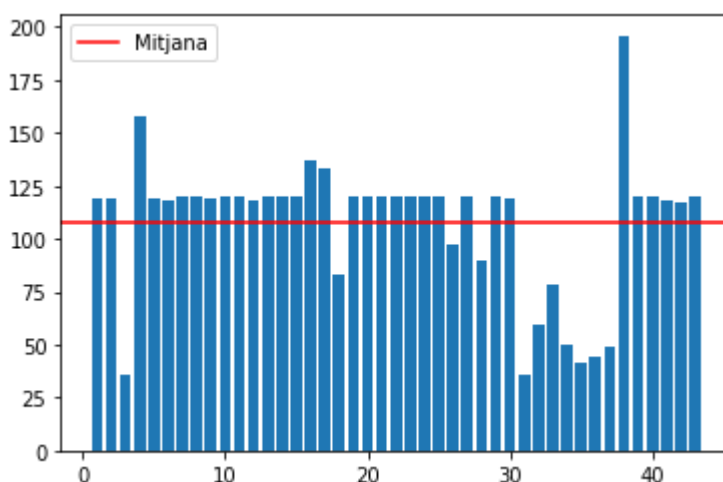


Figura 4.2: Nombre final d'imatges per model

No totes les imatges del conjunt de dades tenen les mateixes dimensions. Per facilitar l'entrenament, volem estandarditzar totes les dimensions de les imatges. Com que totes les imatges tenen forma rectangular (amb l'axis horitzontal sent el més llarg), es decideix que les imatges tinguin un format de 192x128 per tal de preservar al màxim les formes i distorsionar-les el menys possible.

S'ha elegit les dimensions 192x128 Per tal de reduir el tamany per imatges, fent més ràpid l'entrenament i ajudant a generalitzar més les característiques del vehicle i evitant l'overfitting. També donarem les imatges en blanc i negre per reduir la confusió del mateix vehicle amb diferents colors.

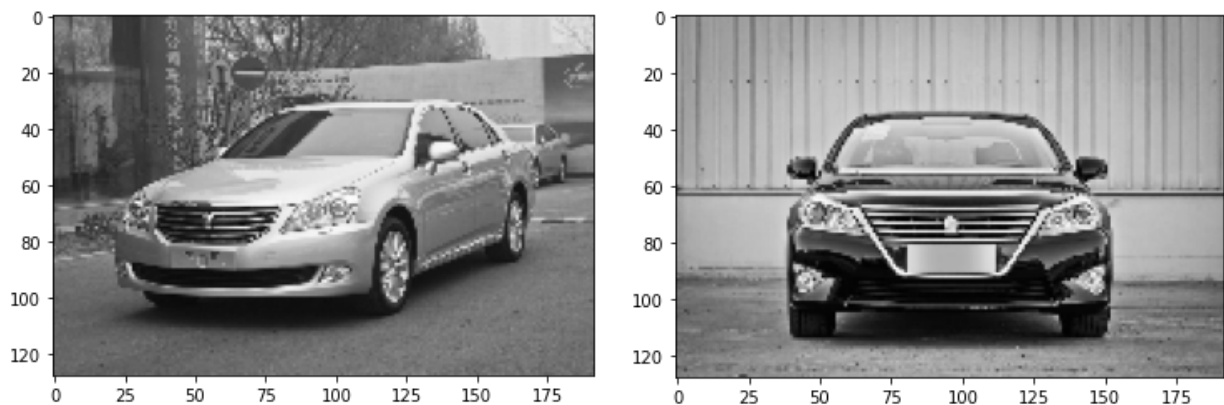


Figura 4.3: Imatges després del preprocessament

Les 95 imatges per model encara és un nombre menor al necessari per poder obtenir un resultat de qualitat. Per això, utilitzarem un Generador de Datasets per ampliar el nostre conjunt de dades. Concretament, utilitzarem el ImageDataGenerator de la llibreria keras.

El ImageDataGenerator fa transformacions sobre cada imatge, com rotacions girs horitzontals, canvis de centre... Aquest generador no crea un nou Dataset, sinó que a cada pas de l'entrenament fa una transformació aleatòria a cada imatge, obligant al model a entrenar amb imatges lleugerament diferents per cada epoch.

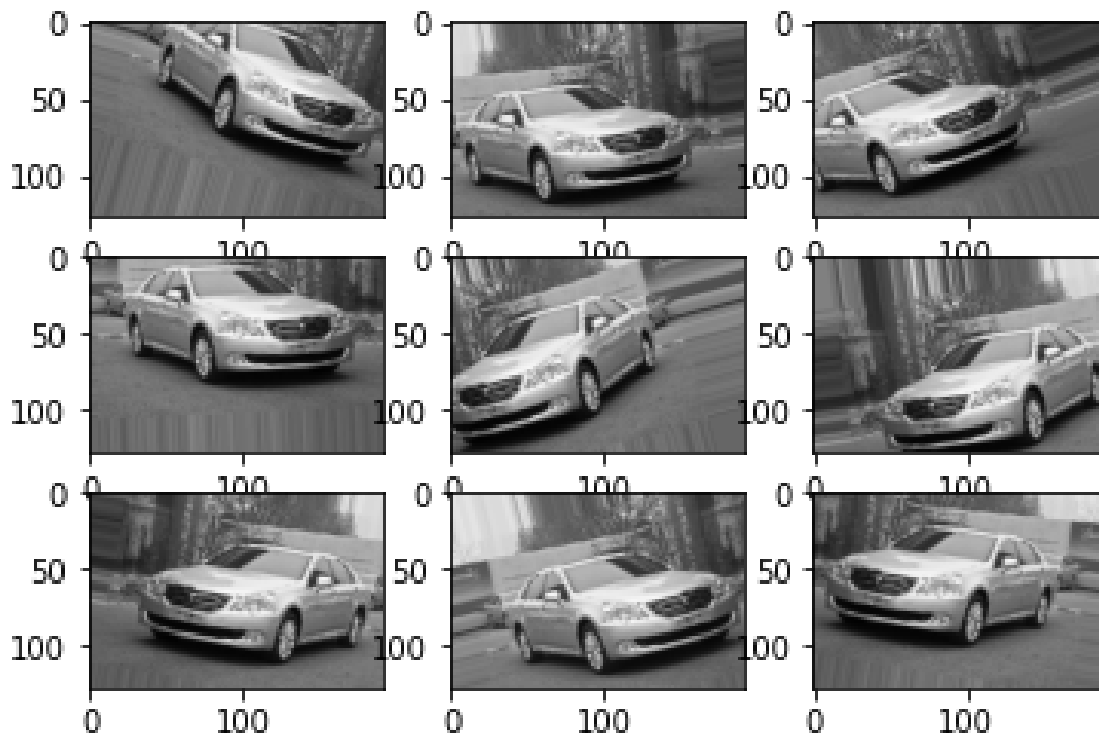


Figura 4.3: La mateixa imatge transformada pel ImageDataGenerator

En el nostre cas, les transformacions que apliquem són:

- Rotació de fins a 20 graus als dos costats
- Canvi del centre de la imatge, fins a un 20% horitzontalment i verticalment.
- Flip horitzontal de la imatge.

Amb el nostre conjunt de dades creat, podem passar a crear el Model Inicial.

4.2 Construcció Model Principal

Com s'ha comentat durant la planificació, primerament construirem el nostre model d'aprenentatge que pugui reconèixer models de vehicles vist des de davant. Primerament, farem una extracció del dataset de les imatges vistes desde davant (Angle Front i Side-Front). Utilitzarem el 90% de les imatges

Construirem la nostra xarxa neuronal. Com s'ha explicat en el capítol anterior, utilitzarem capes de Convolució i capes de Pooling. Finalment, es farà una capa de classificació i les 47 neurones que donaran els resultats. Podem utilitzar la funció `model.summary()` de tensorflow per veure el Model:

Layer (type)	Output Shape	Param #
conv2d_52 (Conv2D)	(None, 189, 125, 32)	544
max_pooling2d_22 (MaxPooling)	(None, 94, 62, 32)	0
conv2d_53 (Conv2D)	(None, 91, 59, 64)	32832
max_pooling2d_23 (MaxPooling)	(None, 45, 29, 64)	0
conv2d_54 (Conv2D)	(None, 38, 22, 128)	524416
flatten_8 (Flatten)	(None, 107008)	0
dropout_8 (Dropout)	(None, 107008)	0
dense_14 (Dense)	(None, 43)	4601387

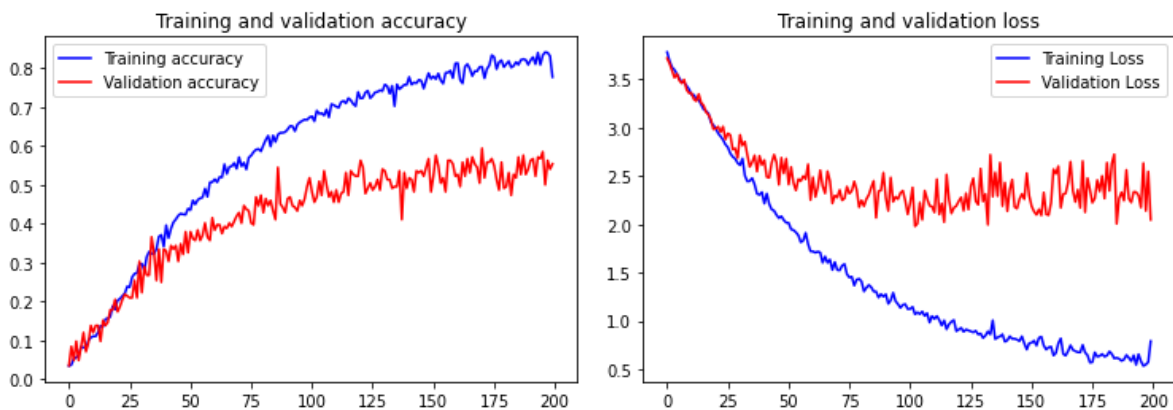
=====
Total params: 5,159,179
Trainable params: 5,159,179
Non-trainable params: 0
=====

Figura 4.4: Capes neuronals del Model principal

En total, el nostre model té 5,159,179 neurones. Una xifra prou alta per obtenir un model de qualitat, però no excessivament gran, ja que augmentaria exponencialment el temps d'entrenament.

Podem inicialitzar l'entrenament amb la funció `model.fit()` passant les imatges d'entrenament amb el `ImageDataGenerator` com a paràmetre. També passarem les imatges de test amb el `Generator`.

El nostre entrenament consistirà en 200 epochs, ja que després de varies proves es conclou que no hi ha una millora significativa al fer més. Es divideixen les imatges en "batches" de 16 imatges, creant 139 steps per epoch. Cada epoch tarda de mitja uns 15 segons per completar-se, creant un entrenament d'aproximadament 50 minuts.



4.5 Historials d'aprenentatge del model principal

En aquests gràfics podem observar l'evolució de la precisió (esquerra) i de la pèrdua (dreta). Es poden observar dues mètriques durant l'evolució de la precisió, la precisió d'entrenament i la precisió de test. La precisió d'entrenament és la precisió a la qual arriba el model amb les imatges que està entrenant. Aquestes imatges són les que influeixen a la capa de classificació per fer decisions. La precisió de test, en canvi, és la precisió sobre imatges que el model no ha vist mai. La precisió de test, com és d'esperar, és bastant més baixa que la d'entrenament, un 57% respecte el 89%.

La mètrica que ens interessa a nosaltres és la precisió de test, ja que l'objectiu és que el model reconegui imatges que no ha vist mai. Per tant, passem a fer una anàlisi

4.3 Resultats i Anàlisi del Model Principal

Al tractar-se d'un model de CNN amb 43, el nostre model d'aprenentatge no retorna una sola solució, sinó un vector de puntuació, amb una posició per model de vehicle . Cada posició indica la probabilitat de que la imatge donada al model sigui la indicada. Llavors, calcularíem la precisió com:

$$Precision = \frac{p_{correcta}}{\sum_{i=1}^{43} p_i}$$

on p és la puntuació de cada model de vehicle. És a dir, la precisió serà la puntuació del model correcte respecte a la puntuació total.

Com s'ha vist en la secció anterior, la precisió del nostre model amb imatges que no s'han utilitzat per entrenar-se és del 56%. Això vol dir que el model correcte té, com a mitja, un 56% de confiança de que és la correcta.

Una segona mètrica que podem utilitzar per analitzar és la taxa d'encert. Calculem la taxa d'encert com la fracció de probes on el model correcte té la puntuació màxima.

Podem esperar una taxa d'encert major a la precisió, ja que quan el model correcte té més d'un 50% de probabilitats (és a dir, una precisió del >50%), hi ha un 100% de probabilitats de que sigui el model correcte. Quan el model correcte té menys d'un 50% de probabilitats, però majors de 35%, també podríem esperar amb alts graus de probabilitat de que segueixi sent el model amb més probabilitats, ja que en un vector de 43 models, seria poc probable que un altre model tingui més de 35%.

Tot i aquesta hipòtesi, s'ha observat que la taxa d'encert és solament del 59%, solament un 3% major a la precisió. Per entendre aquest fet, podem observar els vectors que retorna el model.

La majoria dels vectors que retorna el model, quan el resultat amb major puntuació és correcte són similars a aquests:

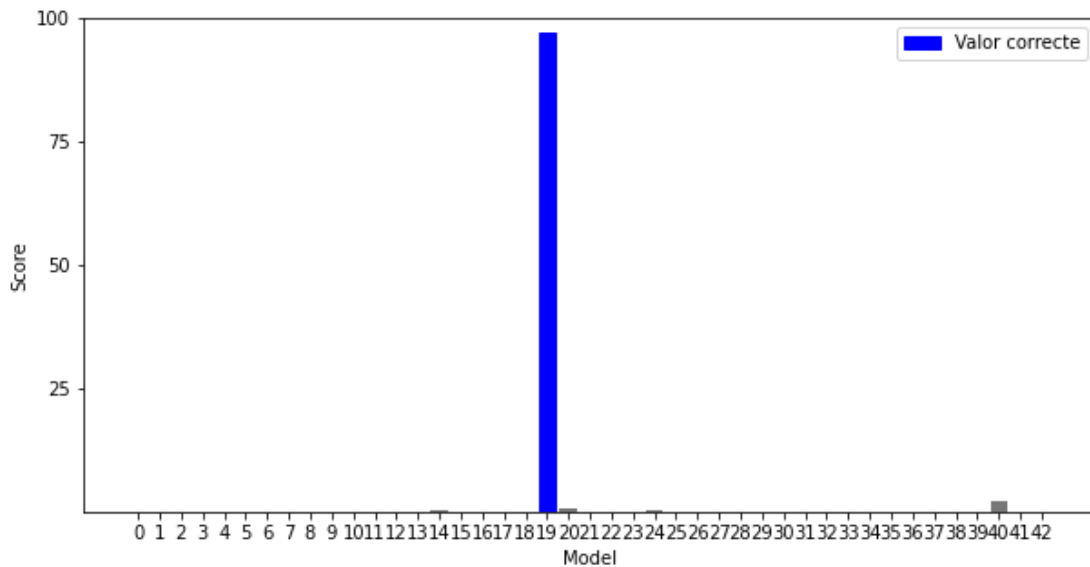


Figura 4.6: Vector de puntuació d'una predicció correcte

I la majoria dels vectors que retorna el model, quan el resultat amb major puntuació és incorrecte són similars a aquests:

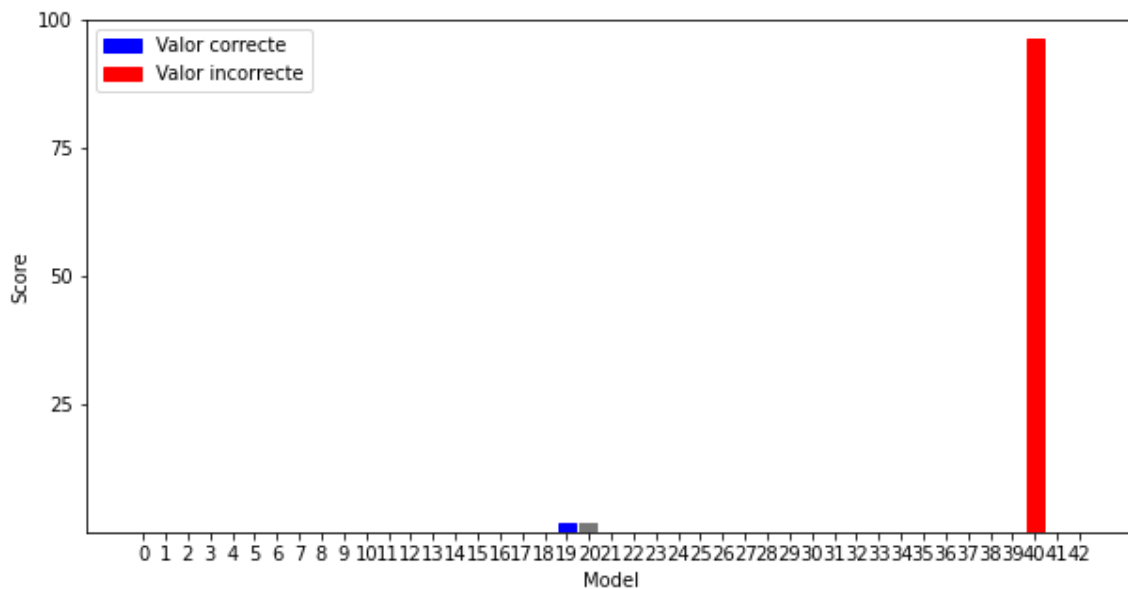


Figura 4.7: Vector de puntuació de una predicció incorrecte

Com podem observar, en els casos que són correctes la puntuació és del 90-100% mentre en les que no són correctes són del 0-10%. Encara que la mitja de puntuació en el correcte sigui del 56%, la majoria dels casos són puntuacions extremes, causant que la diferència entre la precisió i la taxa d'encert sigui menor de l'esperat.

Per reduir aquesta discrepància, podem utilitzar el dataset generator de la mateixa forma que l'hem utilitzat durant l'entrenament, i generar la mateixa imatge diversos cops analitzant-la amb el model. Aquest canvi fa que les puntuacions es vegin de la següent forma:

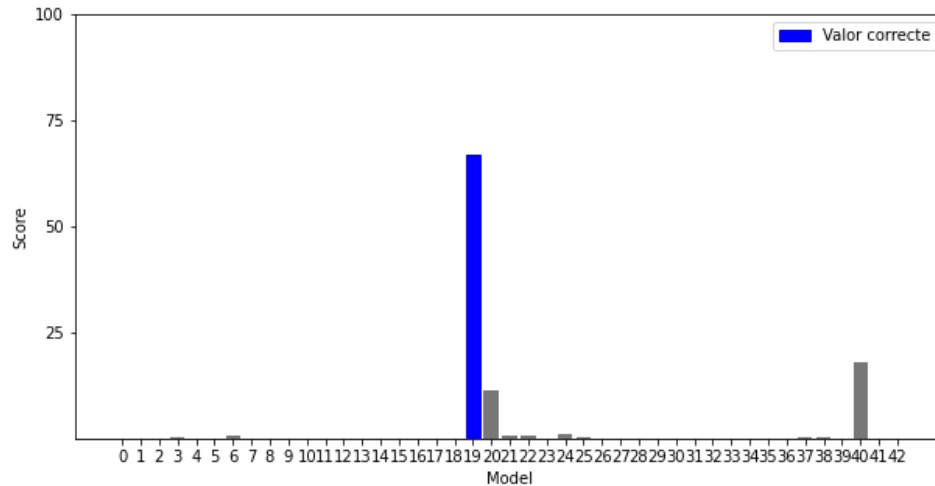


Figura 4.8: Vector de puntuació corregit

Al tenir més perspectives, i per tant més opinions, observem que els hi han més models que passen del 10% de puntuació. Això no fa que pugui la precisió, però sí la taxa d'encert, ja que, de mitja, el model correcte serà el que més puntuació tingui.

Podem veure l'increment de la taxa d'encert a mesura que augmentem el nombre d'imatges generades pel Generator:

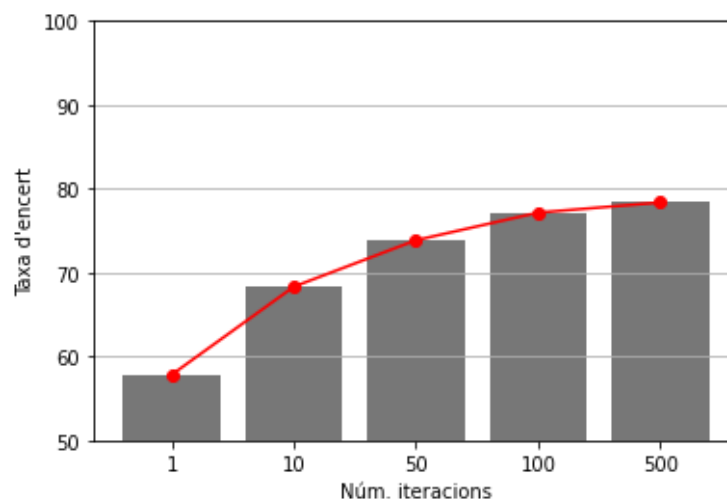


Figura 4.9: Taxa d'encert amb diferent nombre d'iteracions

Però també podem veure que el temps de computació augmenta linealment:

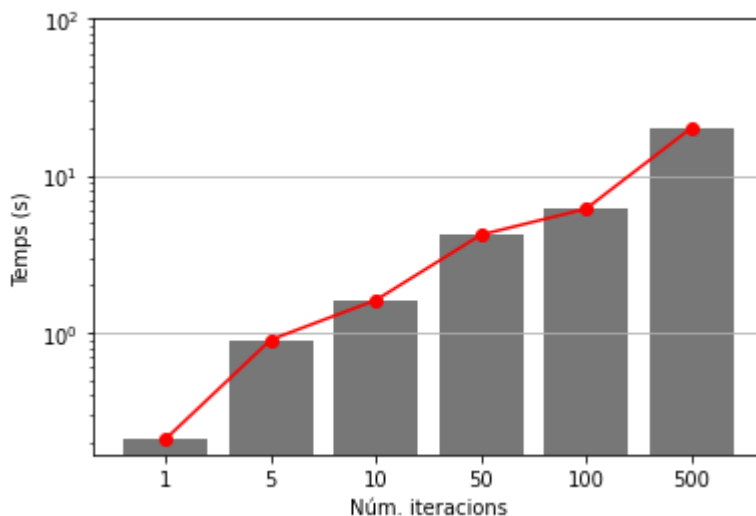


Figura 4.10: Temps de computació amb diferent nombre d'iteracions

Amb 100 imatges és prou correcte per tindre un resultat satisfactori amb una taxa d'encert del 77.15%. Afegir més imatges augmenta molt el temps de computació, però en canvi, puja poc la taxa d'encert.

Com que ara el nostre vector de respostes és més variat, donant més de un model per sobre del 10%, podem intentar veure la taxa d'encert si es consideren els 3 models amb més puntuació:

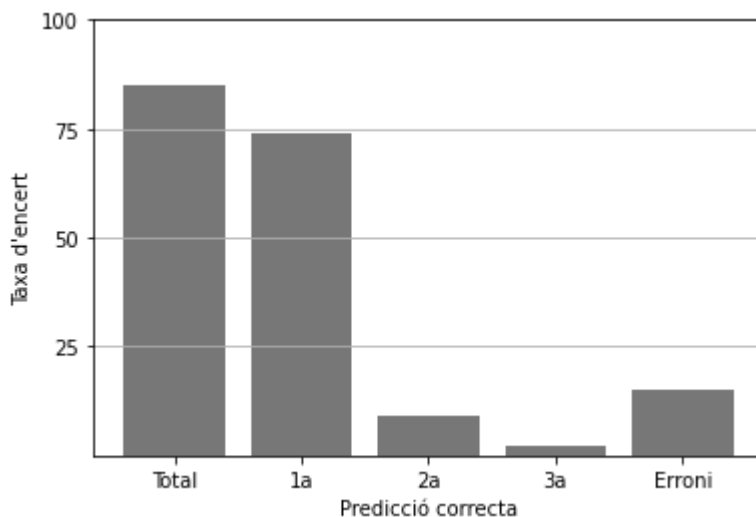


Figura 4.11: Taxa d'encert considerant els 3 models amb més puntuació

Observem que si considerem un encert que la imatge estigui entre els 3 primers models, la taxa d'encert puja fins un 86%.

Per seguir utilitzant el model, podem intentar separar els dos tipus d'angles (front, i side-front) per veure les diferències.

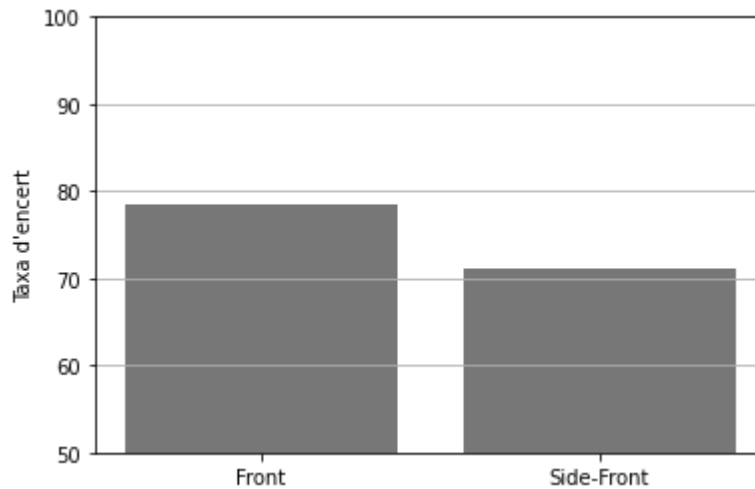


Figura 4.12: Comparació Front i Side-Front

Es pot observar que side-front és pitjor amb una diferència important del 8%, (un 78.47% contra un 72.22%). El nostre dataset té exactament el mateix nombre de imatges en front que desde side-front, així que no hauria d'haver una diferència basada en el mètode d'aprenentatge. La hipòtesi de les causes són les següents:

- Les característiques que fan distingible un model vist desde davant, tenen menys qualitat en una imatge vista desde side-front que desde front. (imatge comparant front i side-front)
- Les imatges side-front tenen més variació de angle que les imatges front, fent-les més distintes entre elles. (imatge comparant dos side-fronts)

Per arreglar aquest problema podem utilitzar el nostre segon objectiu, combinar diferents models de angles per poder reconèixer el vehicle desde qualsevol punt de vista. En aquest cas, podem combinar el model de front i de side, per tenir dos "opinions" de quin és el vehicle de la imatge, fent un 50% de puntuacions dels resultats dels dos models. Això dóna un segon objectiu a la nostra implementació completa.

Amb les dades que tenim, també podem crear una matriu de confusió entre els models. Una matriu de confusió compara els models reals del vehicle d'una imatge (True label) amb els models predit per el Model d'aprenentatge (Predicted label).

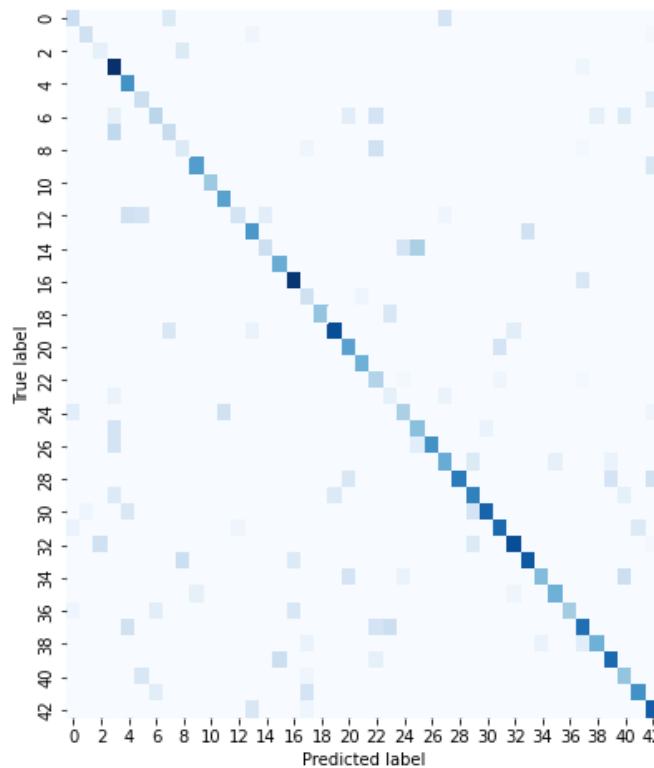


Figura 4.13: Matriu de confusió entre models

Es pot observar que per la majoria de vehicles, el model real és el més predit per la Xarxa. Hi han algunes excepcions, que coincideixen amb models que tenen poques imatges en el conjunt de dades de l'entrenament. Tot i això, es pot considerar que el model principal és satisfactori i es pot passar a implementar el Model Complet.

5. Implementació Completa

5.1 Model Side i Model Rear

El procés per crear el Model de Side i el Model de Rear serà equivalent al procés de crear el Model de Front. Al ser el mateix nombre de models, la xarxa neuronal serà la mateixa:

Layer (type)	Output Shape	Param #
conv2d_52 (Conv2D)	(None, 189, 125, 32)	544
max_pooling2d_22 (MaxPooling)	(None, 94, 62, 32)	0
conv2d_53 (Conv2D)	(None, 91, 59, 64)	32832
max_pooling2d_23 (MaxPooling)	(None, 45, 29, 64)	0
conv2d_54 (Conv2D)	(None, 38, 22, 128)	524416
flatten_8 (Flatten)	(None, 107008)	0
dropout_8 (Dropout)	(None, 107008)	0
dense_14 (Dense)	(None, 43)	4601387
Total params: 5,159,179		
Trainable params: 5,159,179		
Non-trainable params: 0		

Figura 5.1: Capes neuronals del Model Side i Model Front

Amb el model ja construït podem realitzar la fase d'entrenament de la mateixa forma que el Model Principal, introduint les imatges pertinents.

Els paràmetres de la fase d'entrenament seràn equivalents als utilitzats al model principal consistint en 200 epochs. Les "batches" de 16 imatges, amb 139 steps per epoch. Igual que abans, cada model trigarà uns 50 minuts en entrenarse, tardant un total d'aproximadament 1h i 40 minuts per entrenar els dos models.

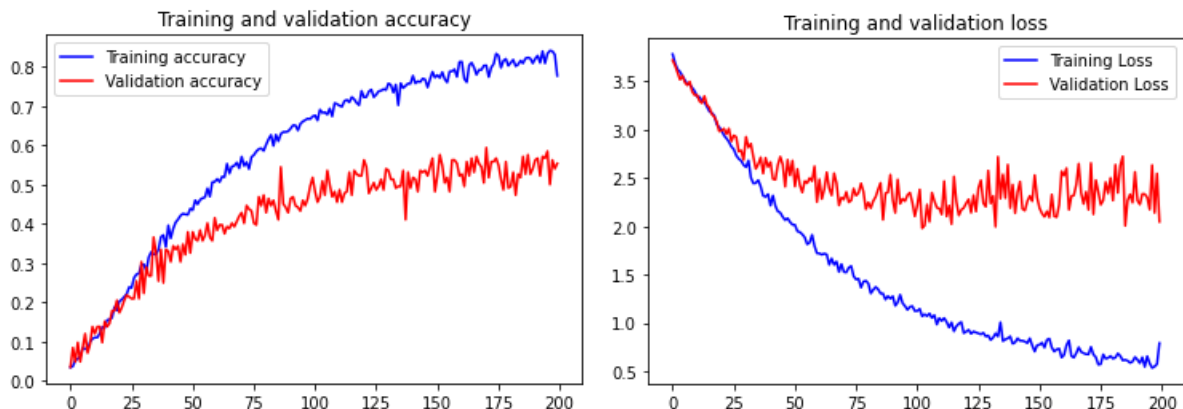


Figura 5.2: Historial d'aprenentatge del model Side

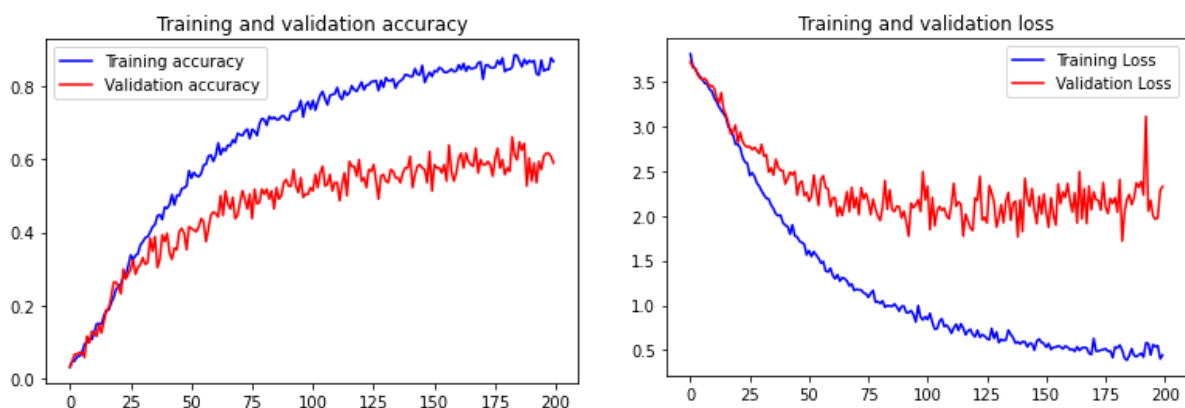


Figura 5.3: Historial d'aprenentatge del model Rear

El Model Side i Model Rear tenen una precisió menor a la del Model Front (52% i 54% respectivament), possiblement causat per un conjunt de dades menor. Tot i així, es considera un resultat satisfactori.

Amb els models ja construïts, podem intentar veure si s'ha resolt el problema creat en la secció anterior, on les imatges side-front tenen pitjor taxa d'encert que les imatges front. La prova que farem serà la següent: comparar la taxa d'encert dels angles front solament utilitzant el model front, la taxa d'encert del angle side utilitzant el model side, i la taxa d'encert del angle side-front utilitzant el model side i el model front i fent una mitja de les puntuacions:

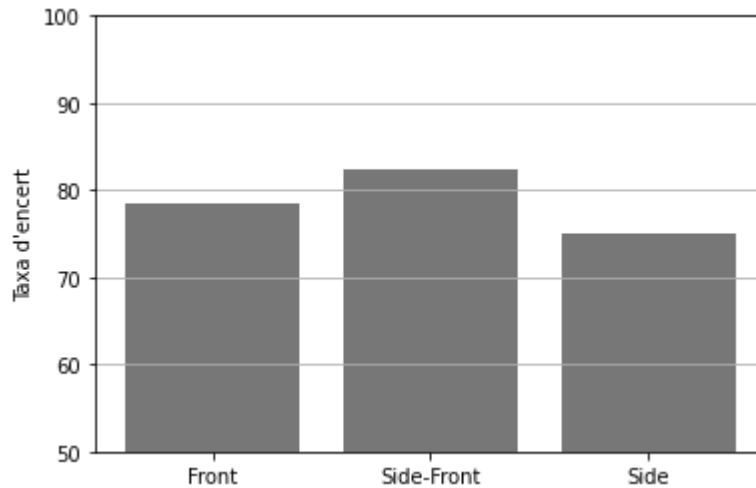


Figura 5.4: Comparació Taxa d'encert combinant Front i Side

Com podem observar, no solament s'ha resolt el problema, sinó que side-front té la millor taxa d'encert. Podem també observar si el mateix efecte es produeix entre les imatges rear, side-rear i side.

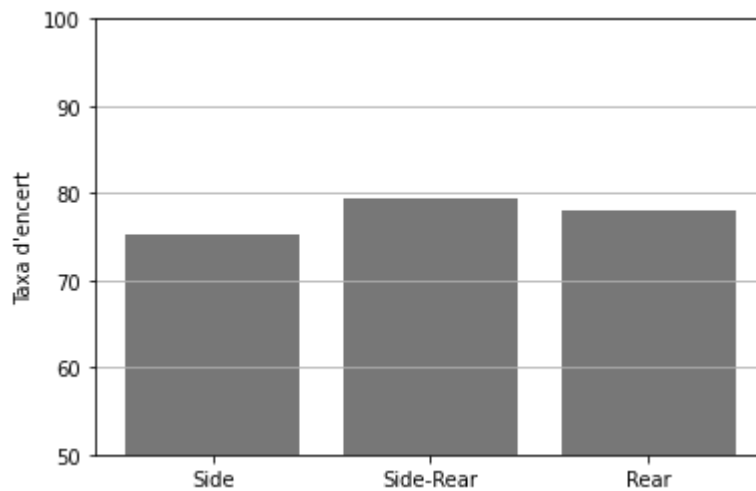


Figura 5.5: Comparació Taxa d'encert combinant Side i Rear

Efectivament, es produeix el mateix efecte, realitzant un resultat satisfactori. Podem passar a la segona part del Model Complet: crear el reconeixedor d'angles per combinar tots els models implementats.

5.2 Combinació d'Angles

Ara que ja tenim tres models cobrint tots els punts de vista, podem passar a crear un model que pugui reconèixer el vehicle sense haver de donar la informació de quin és l'angle de visió de la imatge, creant així un producte més versàtil i útil.

Tal com s'havia planificat al capítol 3, implementarem un sistema que pugui identificar un model de vehicle desde qualsevol angle, sense especificar desde quin angle s'està enfocant la imatge. Per aquest objectiu, crearem un quart model que tingui com a objectiu classificar les imatges en els angles de visió: Front, Side, Rear, Side-Front i Side-Rear.

L'estructura de la xarxa neuronal serà similar a la dels altres models, solament canviant la capa de classificació per que tingui 5 etiquetes, 1 per cada angle de visió:

Layer (type)	Output Shape	Param #
conv2d_27 (Conv2D)	(None, 125, 189, 128)	2176
max_pooling2d_18 (MaxPooling)	(None, 62, 94, 128)	0
dropout_26 (Dropout)	(None, 62, 94, 128)	0
conv2d_28 (Conv2D)	(None, 59, 91, 128)	262272
max_pooling2d_19 (MaxPooling)	(None, 29, 45, 128)	0
dropout_27 (Dropout)	(None, 29, 45, 128)	0
conv2d_29 (Conv2D)	(None, 22, 38, 128)	1048704
flatten_9 (Flatten)	(None, 107008)	0
dropout_28 (Dropout)	(None, 107008)	0
dense_9 (Dense)	(None, 5)	535045
Total params: 1,848,197		
Trainable params: 1,848,197		
Non-trainable params: 0		

Figura 5.6: Capes neuronals del Model Angles

Podem esperar que el nostre model tingui resultats molt més precisos que els anteriors tres, per les dues raons següents:

- Solament hem de classificar les imatges en 5 resultats, minimitzant les possibilitats de confusió entre resultats
- Podem usar un ampli conjunt de dades, ja que no ens hem de limitar a les imatges de Toyota, obtenint un conjunt de dades de 10000 imatges.

Es seguirà fent ús de del DataSetGenerator ja que, tot i que no seria necessari per l'entrenament pel gran tamany del conjunt de dades, ajudarà a tenir consistència respecte la resta de models, ja que les mateixes imatges es passaran d'un model al altre. Amb el nostre conjunt de dades, podem començar el entrenament.

La precisió d'aquest model és molt elevada, amb un 92%. Com que hi ha solament 5 etiquetes, podem investigar la taxa d'encert amb una matriu de confusió amb una prova de 200 imatges:

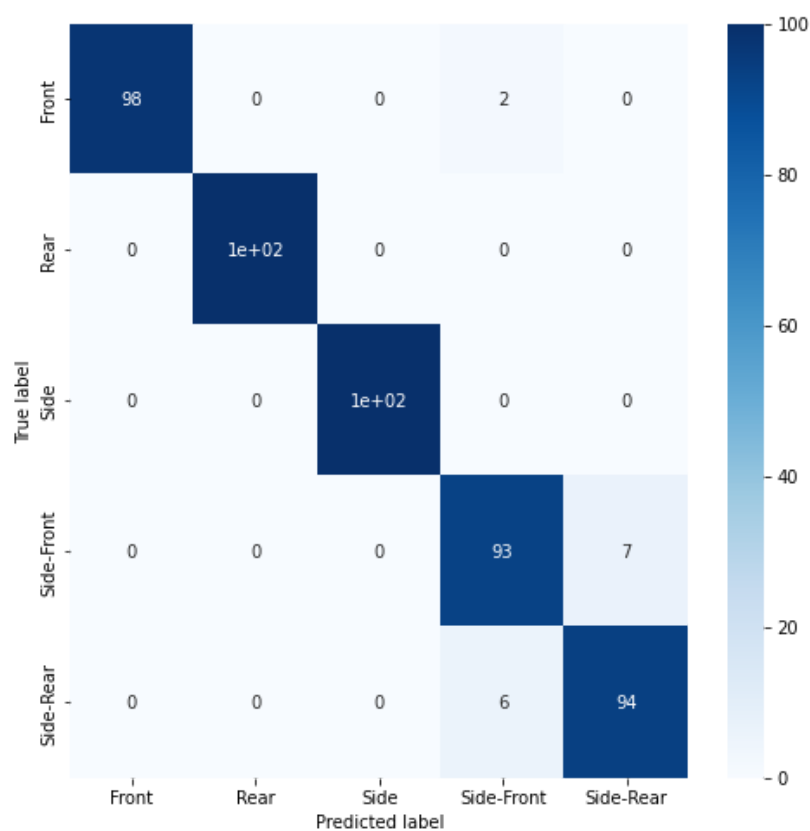


Figura 5.7: Matriu de Confusió Model Angles

Podem observar que la taxa d'encert amb la prova donada és molt alta, arribant al 100% en Rear i Side. També es pot apreciar, però, certa confusió entre Side-Front i Side-Rear.

Per tant, el podem utilitzar de forma fiable per substituir la informació del angle. Amb aquest model podem començar a crear el nostre model complet.

Computarem la imatge per a cada model de angle i utilitzarem la següent fórmula per calcular la puntuació final:

$$\frac{((pF \times F) + (pSF \times 0.5 \times (F + S)) + (pS \times S) + (pSR \times 0.5 \times (S + R)) + (pR \times R))}{5}$$

On pF,pSF,pS,pSR,pR són les puntuacions donades per model d'angles, i F,S,R són els vectors de puntuació de Front, Side i Rear. Aplicant la taxa d'encert i fent les mateixes proves que amb la implementació principal obtenim el resultat final:

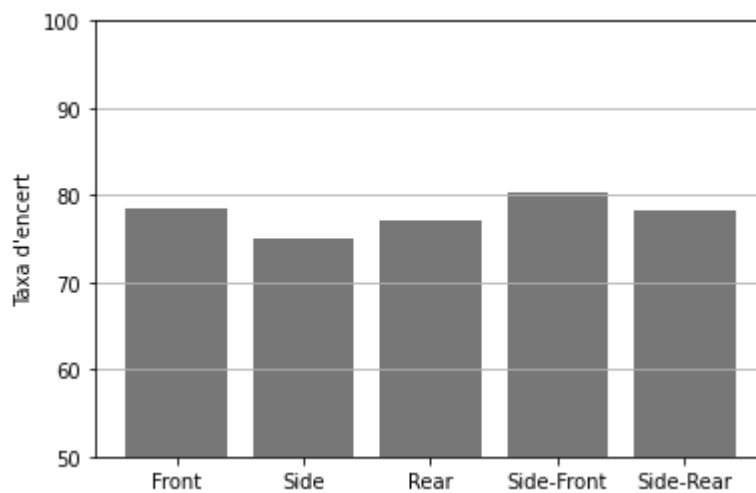


Figura 5.8: Taxa d'encert del Model Complet

Amb una taxa d'encert d'entre un 75% i 80% en tots els angles de visió, podem donar per satisfactoris els resultats i donat per acabat el Model Complet.

6. Implementació Eina Visual

Com s'ha explicat a l'apartat de preparació, utilitzarem Anvil per construir la nostra eina visual en forma d'aplicació. Anvil ens permet fàcilment connectar amb Google Collabs i per tant amb Google Drive, on tindrem els nostres models guardats. L'estructura de usuari-aplicació seria la següent:

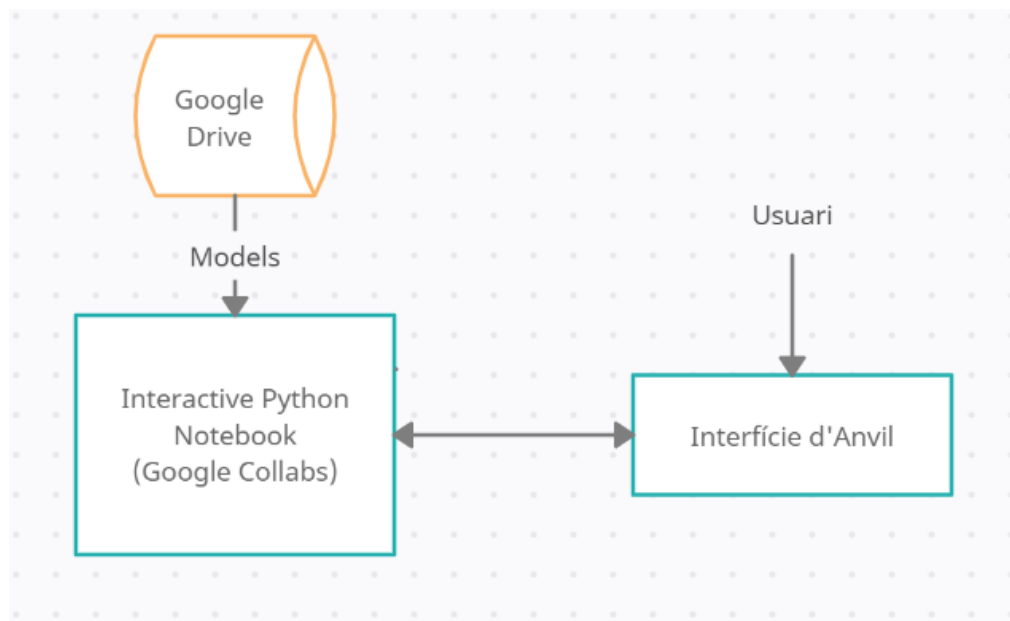


Figura 6.1: Estructura Usuari-Aplicació

Primerament, l'usuari introdueix la imatge a Anvil. Anvil crida a les funcions de Google Collabs amb la imatge com a paràmetre. Google Collabs càrrega de Google Drive els models i fa la computació, retornant les dades de predicció a Anvil. Anvil, finalment, ensenyarà el resultat al usuari.

Desde Anvil, primerament solament s'ensenya un botó per penjar una imatge. Això obrirà el navegador dels discs locals de l'usuari.

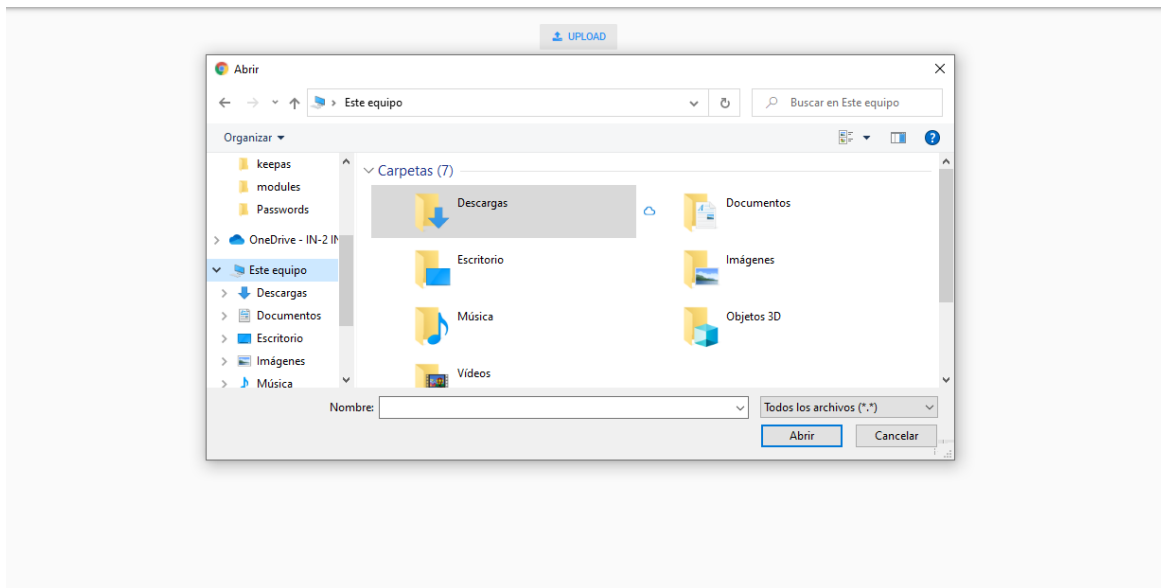


Figura 6.2: Carga d'arxius a l'eina visual

Al penjar la imatge, es cridarà a Google Collabs amb la imatge com a paràmetre i retornarà:

1 vector amb les prediccions dels angles, 1 vector amb amb les prediccions del model i un vector amb els noms dels 3 vehicles amb més puntuació.

Amb la informació retornada de Google Collabs, en Anvil s'ensenyarà aquesta informació:

- La imatge original (per claredat del que s'està computant)
- Un gràfic, ensenyant la puntuació de angles.
- Un gràfic, ensenyant la puntuació de models.
- Un text, escrivint els noms dels models amb més puntuació.

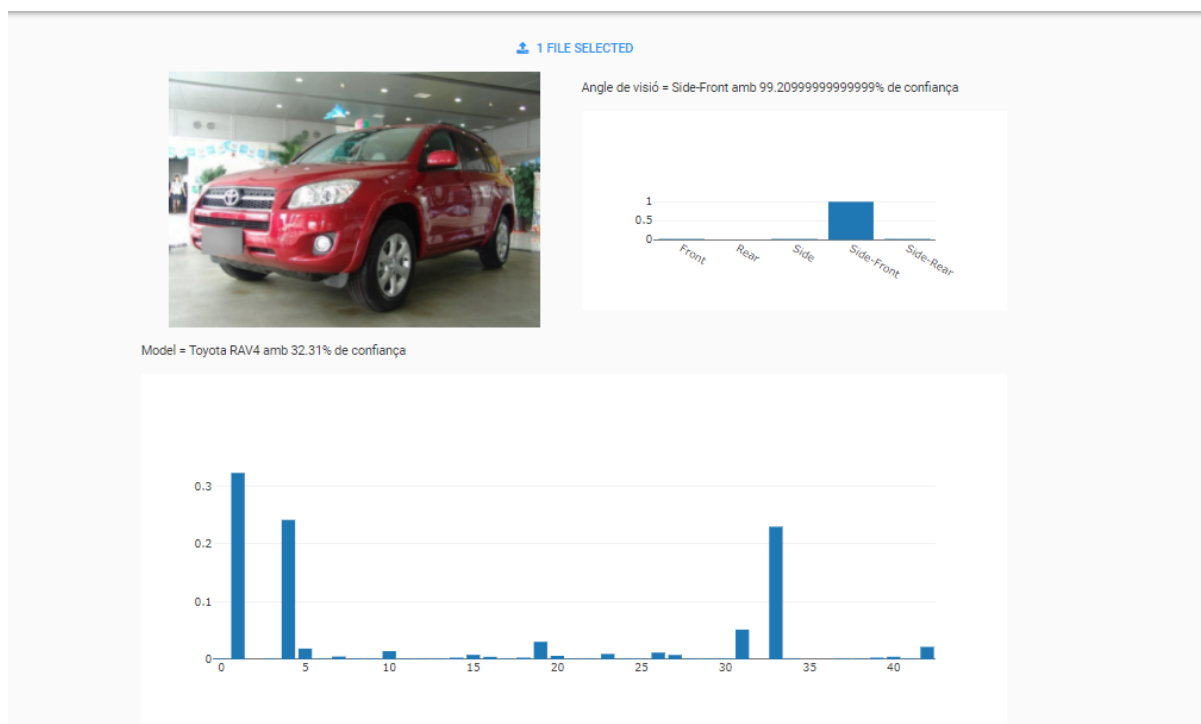


Figura 6.3: Resultats a la eina visual

El botó de penjar imatge seguiran sent present per si es vol computar una altre imatge, reiniciant el procés.

7. Gestió econòmica i sostenibilitat

7.1 Gestió econòmica

En aquesta secció es realitzarà un estudi de la gestió econòmica de la realització del projecte per tal de calcular els pressupostos i gastos del nostre treball, i així saber la seva viabilitat econòmica. Tindrem en compte els següents factors per calcular els nostres pressupostos:

- **Personal:** treballadors del projecte. En el nostre cas tenim de dos tipus: director de projecte i programador.
- **Desenvolupament:** Costos per desenvolupar el projecte. L'espai per treballar, ordinadors, consum elèctric....

Finalment haurem de fer una estimació dels costos pels inconvenients i els riscos.

Cost de personal

Dos tipus de personal són necessaris pel desenvolupament del projecte. El director del projecte, i el programador.

- El **cap de projecte** realitza les tasques relacionades amb l'organització i gestió del projecte.
- El **programador** ha de ser un enginyer informàtic capaç de resoldre els reptes tècnics que presenta el projecte.

Per fer el càlcul total dels costos de personal, cal primer fer un desglossament de les tasques que ha fet cada persona. Per com està plantejat el projecte, no tenim costos de personal en recursos humans o anàlisi de mercat.

Calculant el total d'hores que ha realitzat cada tipus de personal, i fent una recerca sobre els salaris de cada un, podem crear la següent taula:

Posició	Sou brut/hora	Hores	Cost
Cap de Projecte	16€	125	2000€
Programador	11.3€	200	2260€

Taula 7.1: Cost per treballador

Això ens dóna un total de 4316.5€ en costos de personal. A continuació, calculen els costos de cada tasca especificada en l'apartat de Planificació temporal. Per calcular-ho, podem fer un desglossament per cada una, assignant les hores necessàries per a cada treballador:

Tasca	Hores Cap de projecte	Hores programador	Cost per Tasca
Abast	20	0	320€
Planificació	20	0	320€
Pressupost i Sostenibilitat	20	0	320€
Recerca	0	30	339€
Construcció Dataset	0	15	169.50€
Implementació Im. Princ.	0	70	791€
Implementació Im. Completa.	0	70	791€
Implementació Eina Visual	0	15	169.50€
Memòria	50	0	800€
Defensa	15	0	240€

Taula 7.2: Desglossament de tasques final

La tasca amb cost més alt és la creació de la memòria, ja que és realitzat per el cap de projectes i té moltes hores de treball. Com era d'esperar, el cost de les tasques d'implementació són les segones més altes.

Costos de desenvolupament

Les activitats del programador s'han realitzat en un ordinador de mitja gamma, amb un cost aproximadament 600€ i els perifèrics i monitor uns 200€. Deguda a la pandèmia actual, tot el projecte s'ha realitzat localment i, per tant, no hi ha cap cost de lloguer d'espai. Totes les reunions s'han fet telemàticament. Tampoc no hi ha hagut cap despesa en ús de software.

El cost energètic ve donat principalment per la despesa energètica de l'ordinador utilitzat pel programador. Estimem que mentre entrenem el model es gastin uns 120 kWh . Com que l'entrenament per CNN del model és extens, el temps d'ús d'electricitat energètica ha sigut major al de temps de treball del programador (unes 300). El monitor sí sol es farà ús quan el programador estigui treballant. El preu calculat d'electricitat durant la realització pràctica del treball és de 0.121€ / kWh, per tant:

	Hores estimades	Despesa	Cost total
Ordinador	300	120 kWh	14.52€
Monitor	145	3.62 kWh	0.43€

Taula 7.3: Cost energètic

Balanç total i control de gestió

Un cop capturat tots els costos podem fer un balanç total de les despeses per fer una anàlisi dels pressupostos:

Despesa	Cost
Personal	4316.50€
Desenvolupament	811.37€
Riscos	0€ ~ 678€
Total	5.127,87€ ~5.805,87€

Taula 7.3: Balanç total

El cost total del projecte és d'entre 5.127,87€ i 5.805,87€.

Un control de gestió ha sigut necessari per controlar que els costos reals segueixen els costos estimats. Com que la majoria dels costos són dependents del temps que s'ha trigat a realitzar una tasca, l'únic que s'ha de calcular és quantes hores s'ha treballat a cada tasca respecte a les que s'havien planificat inicialment.

7.2 Sostenibilitat

En aquest apartat es farà una anàlisi de la sostenibilitat del projecte, tant de la sostenibilitat econòmica com de la sostenibilitat social i mediambiental. Es dividirà l'anàlisi de Sostenibilitat en tres seccions: Producció del Projecte, Vida Útil del Projecte i Riscos associats:

Producció del projecte

La dimensió econòmica durant la realització projecte és relativament petita, ja que l'actual projecte està orientat a mostrar competències i no a ser un producte competitiu en el mercat. Si el producte final fos destinat al mercat, una major inversió hauria de ser destinada a l'estudi del mercat i més recursos haurien de ser destinats a la creació del model i, sobretot, a l'eina visual. Pels objectius del projecte, la sostenibilitat econòmica és correcte i sostenible.

L'únic aspecte social ha tenir en compte durant la realització del projecte ha sigut la invasió de privacitat en les imatges usades en el conjunt de dades d'entrenament. Com s'ha explicat en els capítols anteriors, el conjut de dades utilitzat solament inclou imatges proporcionades per les marques associades. Per tant, ninguna matrícula ni ninguna persona és inclosa a les imatges sense autorització.

L'impacte que el projecte té en el medi ambient és relativament petit. No s'utilitzen recursos naturals ni s'emet contaminació. L'únic impacte mediambiental rellevant és l'ús d'energia elèctrica que fem servir amb els ordinadors. La despesa elèctrica no solament és causat per l'ús durant les hores de treball del programador, sinó per les hores que l'ordinador ha de passar entrenant el model a través de Xarxes neuronals. Com s'ha esmentat anteriorment, la despesa encara seria més gran si el projecte estigués enfocat en ser competitiu en el mercat.

Vida Útil

Econòmicament serà viable mantindre el projecte, ja que requereix molt poc manteniment. Actualitzacions sobre el model de reconeixement es poden fer fàcilment solament modificant les implementacions i carregant els nous models al servidor.

Es considera positiu l'impacte social que té el projecte sobre la societat. El projecte està enfocat a ser una eina útil per a control de tràfic, i a altres objectius com un sistema de recomanacions de cotxes. Aquesta eina també pot ser utilitzada amb fins autoritaris i invasió de privacitat, però en el marc de la Unió Europea, no es creu que el projecte sigui perjudicial en la societat.

L'impacte mediambiental durant la vida útil del projecte és relativament petita. Solament s'ha de tenir en consideració l'ús d'energia elèctrica a l'hora de reconèixer una imatge, i l'ús de energia per mantenir encès el servidor d'Anvil i de Google Colabs.

Riscos associats

L'únic risc associat durant la vida útil del projecte podria ser el tancament o les modificacions que podrien haver sobre els servidors de Anvil o de Google Collabs. En aquest cas , una nova implementació per a la eina visual hauria de ser realitzada. Inclús considerant aquesta possibilitat, el projecte seguiria sent viable econòmicament.

No es preveu cap risc durant la vida útil del projecte que pugui fer-lo insostenible en el marc Social ni Ambiental. Com s'ha descrit abans, no es preveu que el projecte es pugui fer servir per envair la privacitat de les persones en cap cas, i ningú factor pot augmentar l'impacte ambiental en tal quantitat que el faci insostenible.

8. Conclusions

Els objectius establerts a l'inici del projecte del Treball de Fi de Grau han sigut complets satisfactòriament. S'ha realitzat una recerca sobre tècniques d'Aprenentatge profund i, amb els coneixements obtinguts, s'ha construït un sistema de reconeixement de models de vehicles de Toyota prou fiable, amb una taxa d'encert prop del 80% per poder reconèixer 43 models de vehicles.

Aquest reconeixedor està acompanyat d'una eina visual per facilitar la visualització dels resultats amb qualsevol imatge.

El nostre projecte solament reconeix vehicles de Toyota, però és fàcilment adaptable a qualsevol marca. Solament faria falta canviar el conjunt de dades inicials a la marca desitjada, i canviar la capa de classificació de la xarxa neuronal per tindre un output equivalent al nombre de vehicles de la marca. La Xarxa, llavors, pot ser entrenada per aquesta nova marca.

Es poden implementar futures millores per millorar la qualitat del producte obtingut. La principal millora seria augmentar el conjunt de dades per millorar la precisió. Això també comportaria una extensió del temps d'entrenament. Una nova anàlisi de la Xarxa Neuronal per optimitzar-la amb els coneixements obtinguts durant la creació del projecte també podria millorar la qualitat de la solució.

En perspectiva personal, la realització d'aquest projecte ha contribuït a profunditzar els meus coneixements sobre la intel·ligència artificial, aprenent sobre l'Aprenentatge Profund des de zero.

Referències

[1] **What is Deep Learning and how does it works**

Artem Oppermann

<https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac>

[2] **Principales aportaciones de la visión artificial**

Ignacio G.R. Gavilan

<https://ignaciogavilan.com/principales-aportaciones-de-la-vision-artificial/>

[3] **Deep Deep Learning for Computer Vision: A Brief Review** per Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis i Eftychios Protopapadakis

<https://www.hindawi.com/journals/cin/2018/7068349/>

[4] **3D Object Representations for Fine-Grained Categorization** per Jonathan Krause, Michael Stark, Jia Deng i Li Fei-Fei

<https://ai.stanford.edu/~jkrause/papers/3drr13.pdf>

[5] **Car Dataset** per Stanford University

https://ai.stanford.edu/~jkrause/cars/car_dataset.html

[6] **Cursos Jedi**

<https://cursos.jediupc.com/>

[7] **Eina Gantter**

<https://www.gantter.com/>

[8] **Application of the residue number system to reduce hardware costs of the convolutional neural network implementation**

Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. (2020).

[9] **Eina Google Collabs**

<https://colab.research.google.com/>

[10] **Eina Anvil**

<https://anvil.works/>

[11] **Llibreria Tensorflow**

https://www.tensorflow.org/api_docs/python/tf