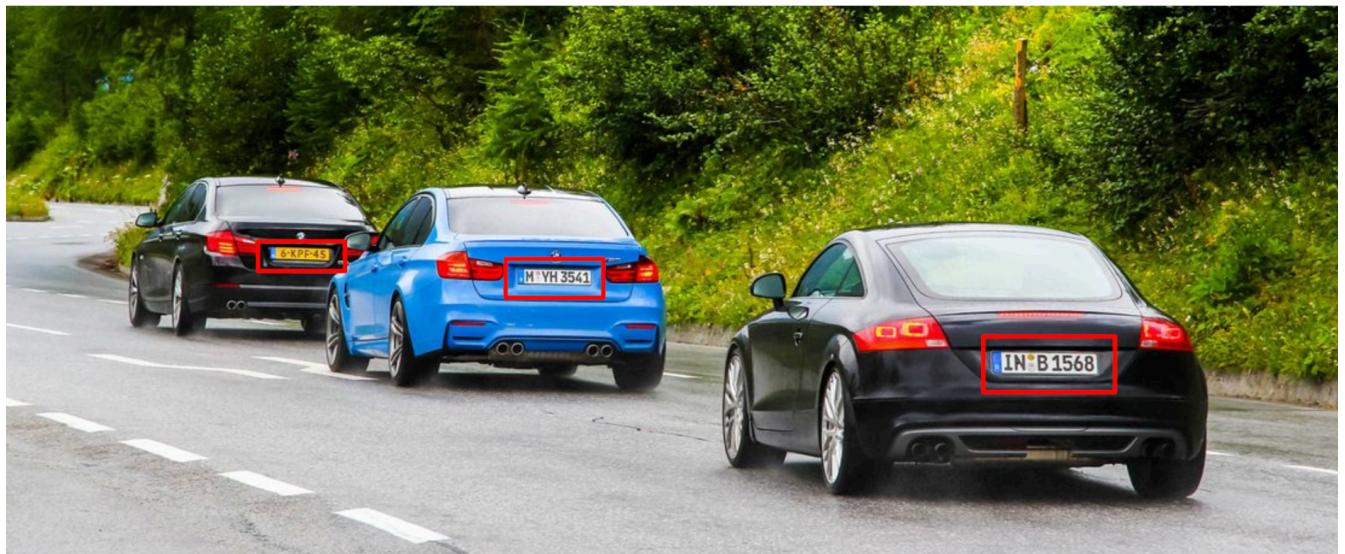


# **UAB**

**Universitat Autònoma  
de Barcelona**

## **Reconeixement automàtic de matrícules**



Mètodes Avançats de Processament de Senyal, Imatge i Vídeo

Octubre 2024

# **1. Introducció**

Aquest projecte se centra en el desenvolupament d'un sistema automàtic de reconeixement de matrícules utilitzant tècniques de segmentació d'imatges i models d'aprenentatge automàtic com xarxes neuronals convolucionals (CNN), SVM i KNN. L'objectiu principal és identificar les matrícules dels automòbils a partir d'imatges reals i reconèixer els seus caràcters amb precisió. En aquest informe es detalla la metodologia seguida, des de la captura d'imatges fins a l'entrenament dels models, i presenta una comparativa del rendiment dels diferents enfocaments emprats.

## **1.1 Objectius**

1. Desenvolupar una tècnica de segmentació d'imatges que permeti aïllar la matrícula del vehicle del seu entorn per facilitar el procés de reconeixement.
2. Entrenar models d'aprenentatge automàtic específics per a lletres i números, per tal de proporcionar una alta precisió elevada en el reconeixement dels caràcters.
3. Avaluar la precisió del model de xarxa neuronal utilitzat i aplicar els canvis necessaris per millorar la precisió i el rendiment global del model.

# **2. Metodologia**

En aquest apartat descrivim detalladament cada fase del procés utilitzat per resoldre el repte del reconeixement de matrícules. La nostra metodologia inclou diverses etapes: adquisició de fotos, localització de la matrícula, segmentació de caràcters i, finalment, el reconeixement de cadascun d'aquests caràcters.

## **2.1 Adquisició de fotos**

El primer pas en el reconeixement de matrícules és l'adquisició de fotos per tal d'avaluar el nostre model i realitzar els tests necessaris. Hem utilitzat tant les

fotos proporcionades pel professor com les que hem pres nosaltres mateixos pel carrer. Aquestes imatges cobreixen una varietat de situacions reals, ja que hem capturat fotografies de vehicles en diferents angles, distàncies i moments del dia, incloent condicions de baixa lluminositat i llum directa per assegurar que el model pugui gestionar situacions possibles que ens podem trobar en el dia a dia.

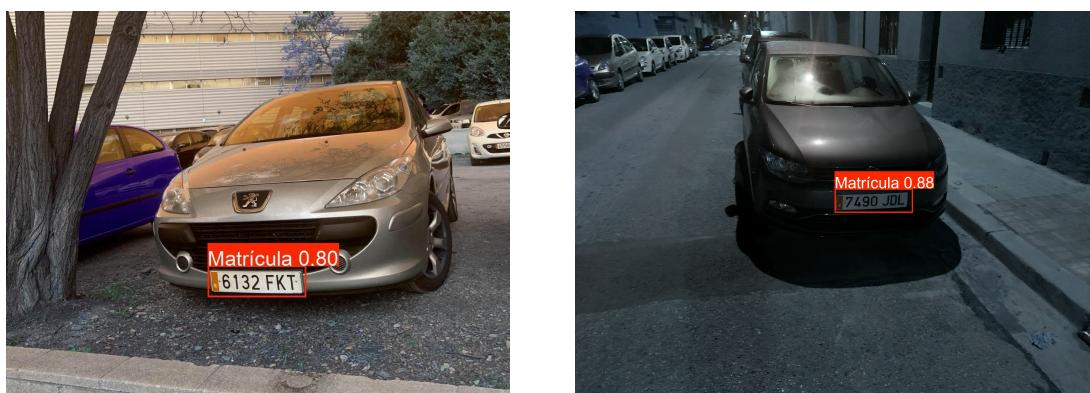


*Figura 1: Fotografies de matrícules preses per nosaltres*

## 2.2 Localització

Per tal d'identificar la matrícula primer necessitem identificar quines són les coordenades on es troba la matrícula. Per la localització hem utilitzat YOLO<sup>[1]</sup>, un algorisme de detecció d'objectes en temps real que és capaç de localitzar diversos elements dins d'una imatge.

Després d'haver entrenat l'algorisme amb 335 fotos de matrícules, hem aconseguit trobar el 100% de les matrícules de les fotos amb una confiança superior al 0.8, tal i com podem veure en les figures 3 i 4.



*Figura 2: Detecció de la matrícula amb YOLO*

## 2.3 Segmentació

Per la detecció e matrícules hem de trobar els caràcters d'aquesta. Primer, hem retallat la regió de la matrícula basant-nos en els contorns detectats amb YOLO, seleccionant aquells que s'ajusten a les proporcions típiques d'una matrícula. A continuació, hem convertit la imatge a blanc i negre per facilitar la detecció dels contorns que delimiten tant la matrícula com els caràcters que conté. Un cop identificats, hem filtrat els contorns que no són rellevants, com aquells que es troben massa a prop de les vores o que no compleixen els criteris de mida i proporció.

Quan hem detectat més de 7 contorns, hem considerat que probablement la imatge incloïa la lletra "E" d'Espanya i la bandera de la Unió Europea, que no formen part dels caràcters essencials de la matrícula. Per això, hem calculat la lluminositat de cada contorn i hem eliminat el que tenia la menor lluminositat, que sol correspondre a la "E". Així, ens hem assegurat que només utilitzem els caràcters necessaris, com podem veure en la figura 5.

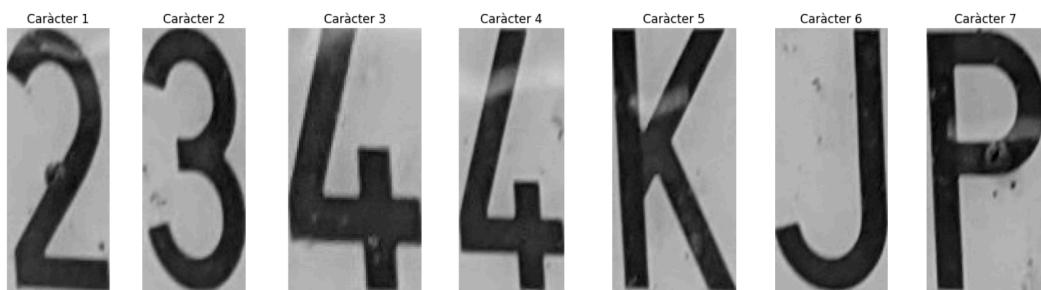


Figura 3: Exemple de matrícula segmentada

## 2.4 Reconeixement

Per al reconeixement dels caràcters de la matrícula, hem provat diferents models d'aprenentatge automàtic. Concretament, hem experimentat amb xarxes neuronals convolucionals (CNN), així com amb altres algorismes com el SVM (Support Vector Machine) i el KNN (K-Nearest Neighbors). Cada model ha estat entrenat i evaluat per determinar quin oferia el millor rendiment en termes de precisió.

En els 3 casos hem entrenat un model conjunt de lletres i números, i dos altres models, un de números i un de lletres, que hem combinat per predir la matrícula al complet. Hem decidit provar de les dues formes per veure si n'hi ha prou amb entrenar el model conjunt on si és necessari separar-los per evitar confusions entre números i lletres. L'entrenament està fet amb el 80% del dataset d'imatges i el 20% restant com a test.

### **3. Disseny experimental**

En aquest apartat s'explicaran amb més detalls certs aspectes mencionats en la metodologia com podria ser descriure els datasets, els models que hem emprat pel reconeixement, els hiperparàmetres dins d'aquests mètodes i les mètriques per poder mesurar la qualitat del model.

#### **3.1 Dataset**

En aquest projecte han calgut dos datasets diferents dependent de la tasca requerida:

- Reconeixement automàtic de matrícules: El primer és el conjunt d'imatges de matrícules de cotxes, el pilar principal que forma tot aquest projecte, per poder reconèixer la matrícula a partir de la imatge. Aquest dataset consta de 60 imatges de matrícules de diferents angles, distàncies i moments del dia, incloent condicions de baixa lluminositat i llum directa.

La font d'on hem obtingut aquest dataset ha sigut tant les imatges proporcionades pel professor com les que hem pres personalment amb cotxes estacionats a la via pública.

- Entrenament del model: Per la fase reconeixement d'alfanumèrics hem explicat que s'ha entrenat diversos models, tots ells amb el mateix dataset format per 30 directoris, un per cada lletra que pugui aparèixer en una matrícula espanyola (A, CH, E, I, LL, Ñ, O, Q i U no hi son) i els 10 possibles nombres (0 - 9).

En cada directori s'hi pot trobar al voltant de 1000 imatges de l'alfanumèric corresponent per aquell directori, creades de manera sintètica, amb diferents angles, distàncies, rotacions i il·luminació.

La font d'on hem obtingut aquest dataset ha estat en la plataforma Kaggle<sup>[2]</sup>.

La partició d'aquest dataset per l'entrenament ha estat de 80% train i 20% test.

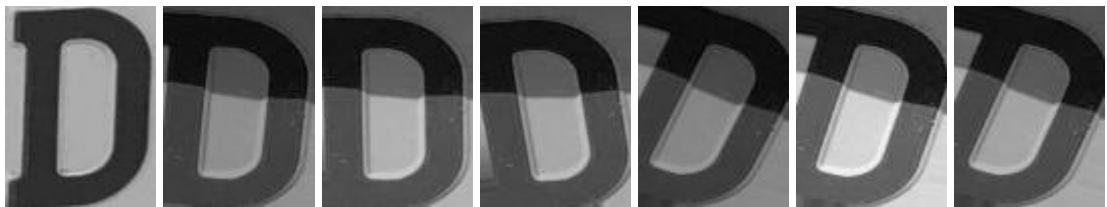


Figura 4: Exemples de lletres D amb diferents característiques

### 3.2 Models

Hem utilitzat tres mètodes per a la classificació: CNN, SVM i KNN.

- CNN: En aquest model lii entrem les imatges amb les característiques d'una alçada de 128 pixels, una amplada de 64 pixels, i 3 canals de color (RGB). A partir d'aquí la nostra xarxa és ascendent on comença amb les capes de convolució de 32 filtres, seguit de 64 i finalment 128filtres, on en aquesta última capa capture les característiques més detallades de les imatges. En cada capa de convolució li afegim una capa de Max Pooling2D per reduir la resolució de la imatge a la meitat, però mantenint les característiques més importants, d'aquesta forma simplifiquem la informació.

Un cop finalitzen les capes de convolució apliquem una capa Flatten, que ens permet aplanar la sortida de les característiques en un vector 1D i així la informació de la imatge és converteix en una llista de nombres.

Finalment, acabem aplicant dues capes de Dense, que ajuden a l'aprenentatge de patrons més abstractes que obtenim de la llista anterior. En la segona capa Dense introduim el número de neurones pels caràcters que volem identificar (10 pel cas dels números, 20 per les lletres), per així utilitzar l'activació Softmax, que converteix la sortida en una distribució de probabilitats per cada classe (lletres o números). Això vol dir que la xarxa retorna la probabilitat que la imatge pertany a cadascuna de les 10 o 20 classes.

- SVM: Aquest model s'ha configurat amb un kernel lineal, ideal per a problemes de classificació i també per la seva simplicitat que en casos com aquest amb volums de dades molt grans ens aporta més rapidesa. A més, en aquesta base de dades ens trobem que són linealment separables.
- KNN: Finalment, el KNN s'ha configurat amb 3 veïns, de manera que el model busca els tres veïns més propers per realitzar les prediccions. Una quantitat estàndards que com hem pogut veure ens dona uns resultats força bons, ja que no és molt genèrica com seria amb menys veïns ni s'hi crea sobreajustament com seria amb més veïns.

Per cada mètode hem creat tres models diferents: un entrenat amb lletres i números conjuntament, un només amb lletres, i un altre només amb números, amb l'objectiu d'avaluar si entrenar un únic model combinat és suficient o si és més eficient entrenar dos models separats per minimitzar errors.

### 3.3 Mètriques

Les mètriques que hem decidit utilitzar per mesurar la qualitat dels diferents models han sigut 2.

En primer lloc hem utilitzat la precisió per mesurar quantes de les prediccions positives són correctes, hem escollit la precisió perquè no volem que hi hagin falsos positius.

Per altre banda, hem utilitzat un hit or miss, és a dir si es detecten tots els caràcter de la matrícula es prediuen bé es considera un hit. Per contra, si falla com a mínim un caràcter es considera miss.

## 4. Resultats

Els models utilitzats per analitzar per a la detecció de matrícules inclouen CNN, KNN i SVM, tant en configuracions separades per lletres i números com conjunts, per comprovar l'eficàcia entre les diferents configuracions. Les mètriques utilitzades per avaluar el rendiment són la precisió, el percentatge de hit or miss.

### 4.1 Experiment 1

L'anàlisi dels resultats mostra que el model SVM Separat és el més eficient per a la detecció de matrícules com veurem a continuació en la *Table 1*, amb una precisió del 90% i un percentatge de hit del 52.46%. Això indica que el SVM té la capacitat de fer prediccions correctes sobre la majoria de les matrícules que analitzem. En canvi, el KNN Separat i el CNN Separat mostren precisions de 86% i 84% respectivament, amb percentatges de hit que disminueixen considerablement, especialment en el cas del CNN, amb només un 29.51%.

En comparar els models conjunts, el rendiment es redueix minoritàriament. El SVM Conjunt presenta una precisió del 87% i un percentatge de hit del 39.34%, la qual cosa demostra que, tot i que continua sent el millor entre els models conjunt, la seva eficàcia es veu variada en comparació amb la seva versió per separat. Els models KNN Conjunt i CNN Conjunt són els menys efectius, amb precisions del 74% en el cas de KNN on mostra el valor més baix amb clara diferència, entre la mètrica de Precision, i els percentatges de hit del 18.03% i 34.43% respectivament també tenen una clara reducció respecte els valors dels models separats.

<b>Model</b>	<b>Precision</b>	<b>Percentatge Hit</b>	<b>Percentatge Miss</b>
<b>SVM Separat</b>	0.90	52.46%	47.54%
<b>KNN Separat</b>	0.86	40.98%	59.02%
<b>CNN Separat</b>	0.84	29.51%	70.49%
<b>SVM Conjunt</b>	0.87	39.34%	60.66%
<b>CNN Conjunt</b>	0.84	34.43%	65.57%
<b>KNN Conjunt</b>	0.74	18.03%	81.97%

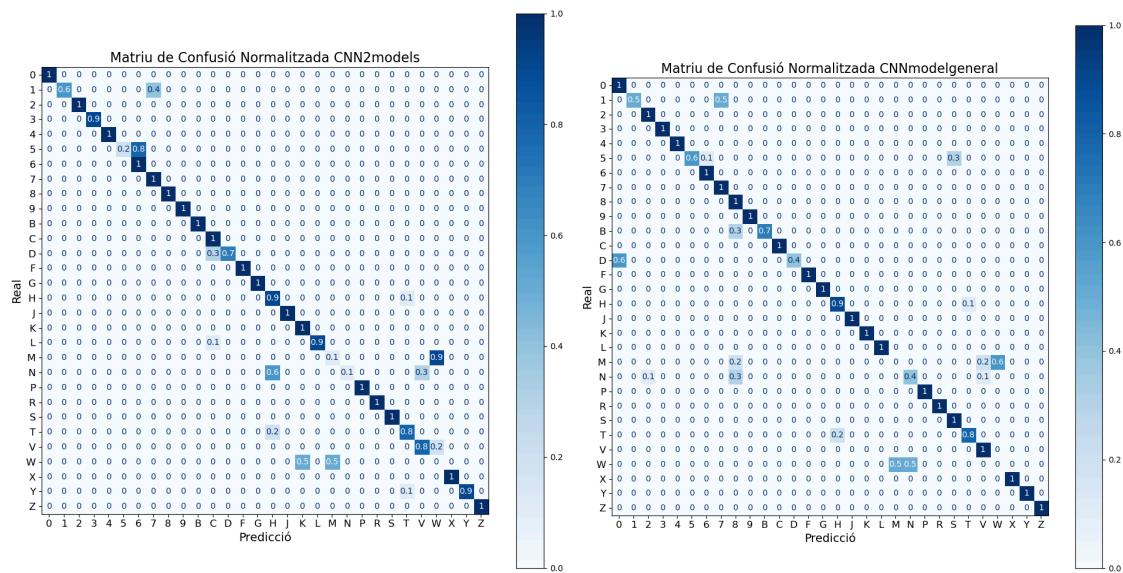
La conclusió dels resultats, mostren que els models separats superen clarament els models conjunt en termes de precisió i capacitat de detecció. El SVM Separat destaca com el model més efectiu (en els dos models es mostra molt constant), mentre que el KNN Conjunt és el menys fiable. Aquests resultats indiquen que per a la detecció de matrícules, és preferible utilitzar models separats, ja que ofereixen un rendiment superior.

### 4.3 Experiment 2

En aquest estudi hem comparat tres models de machine learning CNN, KNN i SVM per a la detecció de caràcters en matrícules, utilitzant dues configuracions, un model conjunt per a números i lletres, i un altre model separats els caràcters. Les matrius de confusió mostren que els models conjunts presenten més confusions entre caràcters similars, com ara entre la "M" i la "W", o el número "1" i la lletra "I". Això suggereix que entrenar models separats és més efectiu per evitar errors com els dels números amb lletres, però no soluciona del tot el problema de confusió entre lletres.

Pel que fa als errors específics, un patró recurrent en tots els models és la confusió de la lletra "M", que sovint és identificada com altres caràcters similars com la "W", la "N", la "V" o la "H". Aquest error es manifesta especialment en el model KNN, mentre que el SVM sembla gestionar millor aquestes confusions, però encara presenta problemes amb la "M".

A l'hora de comparar models veiem que KNN és el que més diferència té entre els models diferenciats per caràcters, amb el contrast de la CNN que manté la similitud entre els seus models. Per acabar, el SVM és el que més estable és ja que, obté els millors resultats en qualsevol dels casos, però tots ells menten el patró comentat en el que tenen la mateixa confusió en certes lletres.



*Figura 5. Matrius de Confusió model CNN*

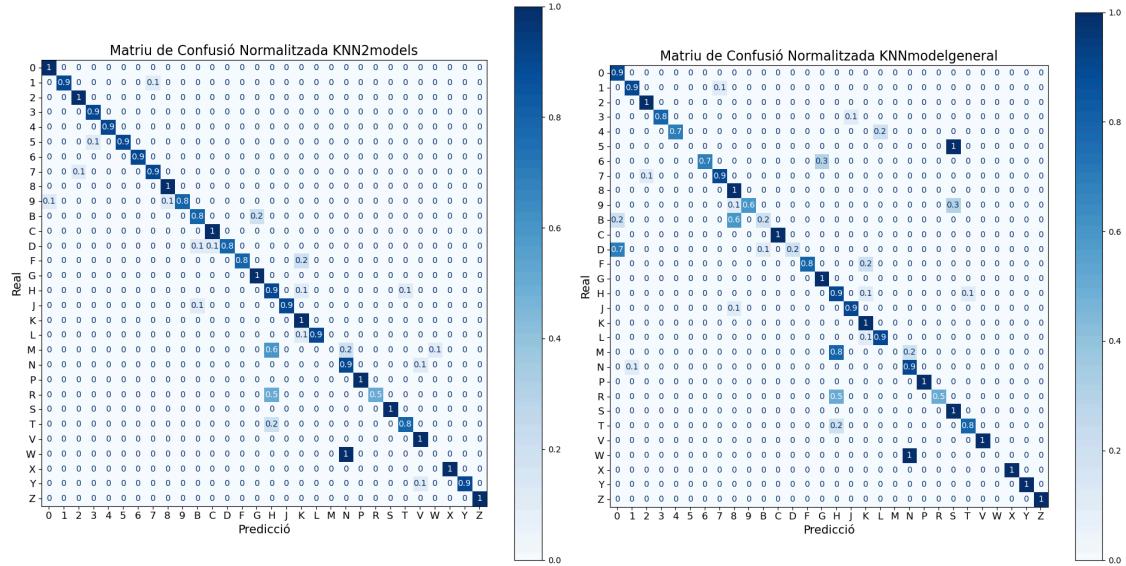


Figura 6. Matrius de Confusió model KNN

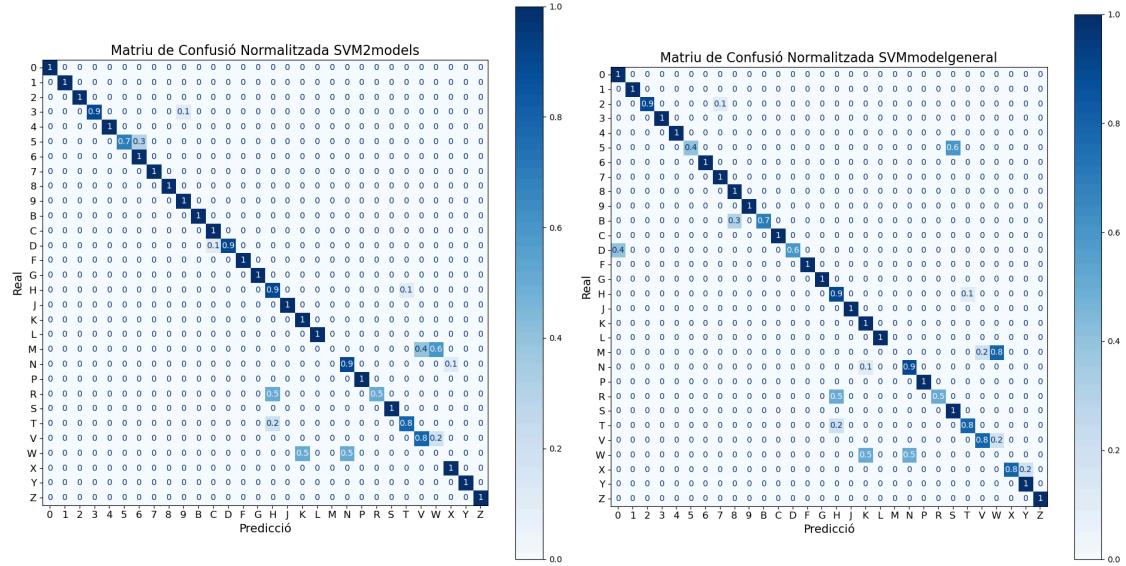


Figura 7. Matrius de Confusió model SVM

En conclusió, entrenar models separats per a lletres i números és la millor opció per reduir errors de classificació. Tots els models tenen dificultats amb la lletra "M", però el SVM amb models separats mostra el millor rendiment general, amb el KNN quedant com el model menys precís per el nostre context de detecció de matrícules.

## 5. Conclusions i millores

Estructurem les millores en dos punts principals:

- La nostra valoració és que les males prediccions en les matrius de confusió que hem comentat anteriorment amb el patró que va succeiren en tots els models, part de la correcció d'aquest pot ser causada per una millora de la base de dades de cada caràcter. El nostre dataset és molt complet i per aquí no s'ha de canviar res, però la tipografia dels números i lletres són de matrícules belgues, tenen molta similitud a l'Espanyola, però no són idèntiques, per tant, això pot ser un dels principals motius de no predir certs caràcters amb exactitud.
- Per últim, en els punts de millora, els nostres models són funcionals i donen uns resultats prou bons, però són molt genèrics, per tant, aquests resultats entrenats en altres contextos també mostrarien resultats acceptables. En conseqüència, gran millora del nostre treball pots ser enfocar els tres models de manera més precisa amb el nostre cas, és a dir detectar matrícules. Les idees podrien ser, comparar més paràmetres de cada model per determinar el més òptim, treballar més al fil del overfitting per crear un model més complex i robust, entra altres pensaments.

Finalment, les conclusions que traiem en aquest treball són bastant clares, ja que la nostra valoració és que gran part del treball està en funció de començar amb una bona base, per tant, obtenir un molt bon dataset tant de matrícules com de caràcters amb totes les característiques que després posem a prova. D'aquesta manera tenir una base de dades de qualitat et permetrà poder abordar els problemes del tractament de dades de forma més fàcil i eficient, fent que ens puguem centrar a generar i entrenar, els models més precisos i òptims de cara al context del treball.

## 6. References

[1] Entrenar Yolov5:

<https://colab.research.google.com/github/roboflow-ai/yolov5-custom-training-tutorial/blob/main/yolov5-custom-training.ipynb#scrollTo=X7yAi9hd-T4B>, Date accessed: Octubre 27, 2024

[2] Kaggle lletres,

<https://www.kaggle.com/datasets/aladdinss/license-plate-digits-classification-dataset/data>, Date accessed: Octubre 27, 2024