UAB
Universitat Autònoma de Barcelona

Repte 1: Reconeixement de matrícules: ANPR

Entregat per:

1639080 - Carlota Cortés

1636358 - Paula Macías

1636671 - Marc Puigbó

1636486 - Pol Riubrogent

Mètodes Avançats de Processament de Senyal, Imatge i Vídeo

18/10/2024

1. Introducció

En aquest primer repte hem treballat en la implementació d'un sistema de Reconeixement Automàtic de Matrícules (ANPR), una tecnologia bàsica però essencial en molts sistemes. Ens permet detectar i reconèixer automàticament les matrícules de vehicles en imatges o vídeos, convertint aquesta informació en text que pot ser processat posteriorment. Aquests sistemes són àmpliament utilitzats en context de gestió d'aparcaments, vigilància del trànsit, etc.

En aquest informe proposem la nostra implementació, formada per 4 fases. Hem provat i utilitzat una mescla de mètodes clàssics i xarxes neuronals.

El codi del treball es pot trobar al repositori GitHub: https://github.com/NIU1636486/PSIV2_Repte1

1.1 Objectiu

L'objectiu principal del projecte és, a partir d'una imatge, poder detectar i reconèixer correctament la matrícula d'un cotxe matriculat a Espanya (amb la versió de matriculació més nova).

2. Metodologia

Per tal de dur a terme aquest repte, l'hem dividit en 4 parts:

1- Adquisició de les imatges
2- Localització
3- Segmentació
4- Reconeixement

Al acabar totes les fases, hem generat el següent diagrama per mostrar el flux del nostre sistema:

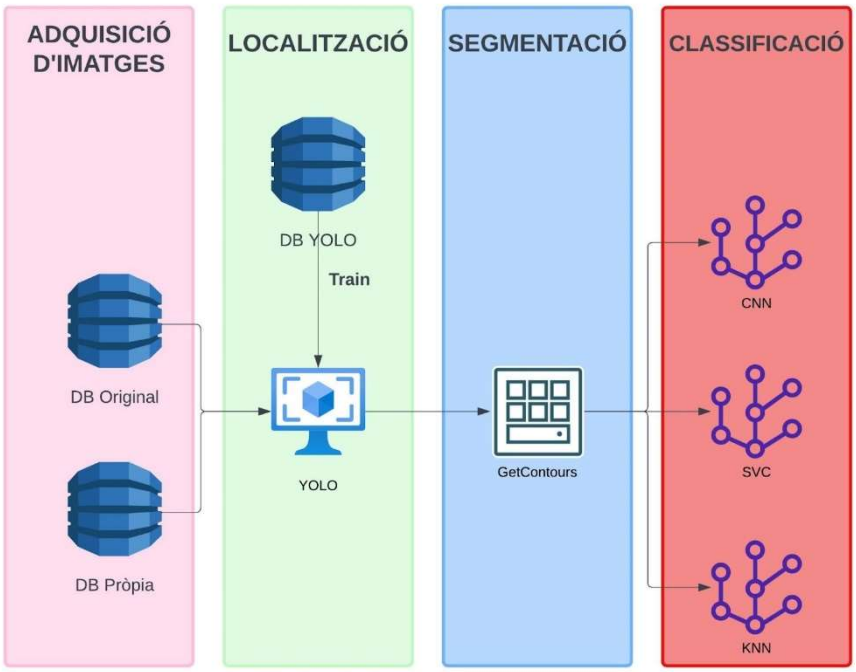

Figura.1. Diagrama del repte

2.1 Adquisició de les imatges

En aquest projecte hem tingut 3 bases de dades d'imatges: una proporcionada pel professor, una de collita pròpia i una tercera utilitzada només per entrenar el sistema YOLO, utilitzat en l'apartat de localització. A més hem aplicat Data Curation per seleccionar les imatges que estan fetes des de la dreta de la matrícula, per estandarditzar el sistema.

2.1 Localització

Un cop adquirides les imatges que contenen un vehicle, el primer pas va ser localitzar la regió de la imatge que conté la matrícula. En un principi, vam aprofitar les combinacions tècniques bàsiques de processament d'imatges per revelar la regió. No obstant, al final vam fer servir un model entrenat de YOLO degut a la seva precisió.

Mètode 1 – Tècniques de processament d'imatge

El primer mètode que vam implementar va consistir en combinar diverses tècniques de processament d'imatges [3], com ara operacions morfològiques, detecció de gradients, aplicació de llindars (thresholding) [4] i anàlisi de contorns.

Vam començar llegint la imatge i convertint-la a escala de grisos per als processos següents. Tot seguit, vam aplicar un filtre blackhat amb un kernel rectangular. Aquest pas és un bon punt de partida perquè permet ressaltar les regions clares en un fons més fosc, com és el cas de la matrícula. A més, el kernel rectangular amb proporcions similars a una matrícula (una amplada de 13 i una alçada de 5) ajuda a ressaltar les àrees que tenen una estructura similar

Figura 2. Imatge amb blackhat

A continuació, vam aplicar el gradient de Scharr per detectar vores i línies a la imatge, tenint en compte les variacions en la intensitat dels pixels. Per després normalitzar els valors a una escala de [0,255] per millorar la visualització i futur processament.

Per reduir el soroll i suavitzar les vores, vam aplicar un desenfocament Gaussià, seguit d'una operació de tancament amb un kernel rectangular per omplir petits buits i identificar millor les estructures de la imatge. Posteriorment vam aplicar un llindar per separar les regions d'interès del fons.

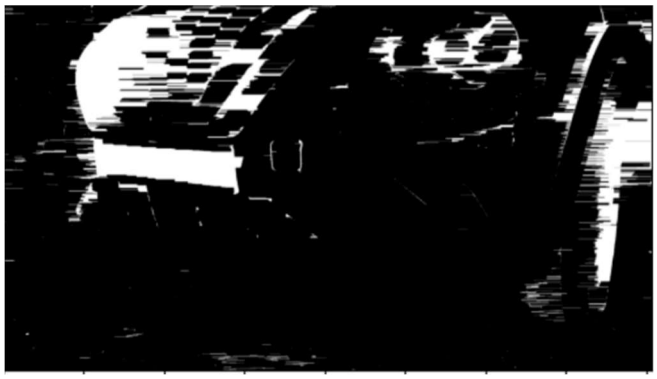

Figura 3. Imatge amb desfoc i tancament

Com que es continuaven detectant àrees petites de soroll, vam fer una operació d'obertura per eliminar aquest soroll i definir millor els contorns, reduint així els falsos positius.

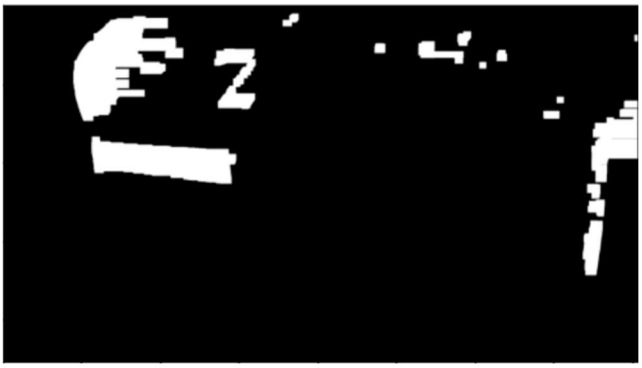

Figura 4. Imatge amb operació d'obertura

Finalment, vam detectar els contorns amb la funció findContours(). Aquesta funció els detecta tots, en si totes les regions candidates a matrícules. Per aquest motiu, vam aplicar un filtratge basat en l'àrea i les proporcions per trobar el que millor s'adapta a les característiques d'una matrícula, proporcions similars a 5:1. D'aquesta manera, vam poder eliminar els falsos positius restants.

Figura 5. Imatge amb contorns i matrícula detectada

Tot i que, aquest mètode va detectar correctament la regió de la matrícula en alguns casos **(Figura 5)**, els resultats no eren consistents. Només va encertar tres de deu imatges de vehicles. Per això, vam decidir realitzar un altre mètode, el YOLO.

Mètode 2 – YOLO

YOLO (You Only Look Once) és un mètode de detecció d'objectes en temps real que processa la imatge en una sola mirada, com diu el nom. YOLO tracta tota la imatge alhora, predint directament les

coordenades de les “bounding boxes” i les classificacions d’objectes. Cosa que el fa extremadament ràpid i eficaç.

Per entrenar el model YOLO, vam utilitzar un conjunt de dades de Roboflow [7] format per 171 imatges per a entrenament i 16 imatges per a validació, totes elles ja anotades en el format YOLO.

Vam descarregar l’última versió (11) del model YOLO i vam configurar el fitxer “config.yaml” amb els camins d’accés a les dades i especificacions de la classe a detectar.

L’entrenament el vam realitzar a Google Collab, aprofitant la GPU per accelerar el procés, ja que es van executar 23 èpoques. Al final de l’entrenament, vam obtenir resultats de mètriques d’avaluació, corbes de pèrdua, imatges de validació amb les prediccions del model **(Figura 6)**, així com una carpeta amb els pesos finals del model per a futures prediccions.

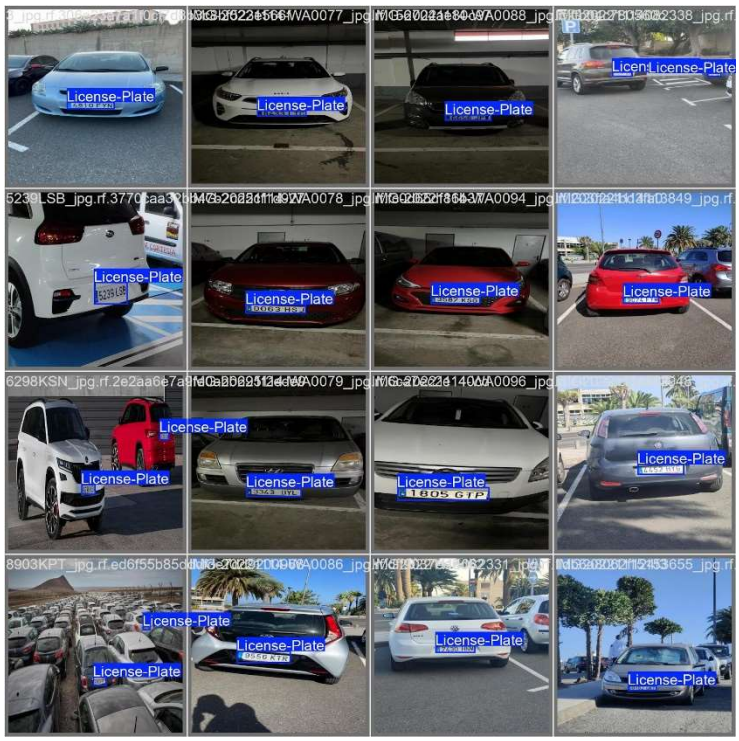

Figura 6. Imatges de validació amb les prediccions del model

Els models YOLO proporcionen com a sortida les coordenades de les “bounding boxes” i els valors de confiança associats a cada detecció. Llavors per treballar amb els resultats vam iterar sobre cada “bounding box” per extreure les coordenades. Per així, dibuixar el rectangle delimitador a la imatge original, i extreure la regió d’interès (ROI) per a la segmentació posterior.

Figura 7. Imatge amb la regió de la matrícula detectada YOLO

En les proves amb les imatges proporcionades pel professor, el model va detectar correctament 16 imatges de 17, assolint així una taxa d'encert del 94%. L'error va ser en un cas en què el model va identificar el frontal del cotxe en lloc de la matrícula, com es pot veure en la **(Figura 8).**

Figura 8. Imatge amb la regió de la matrícula detectada incorrectament

Llavors a causa de la seva alta precisió i rapidesa, vam optar pel model YOLO. A diferència de les tècniques de processament d'imatges, que en la nostre opinió requereixen un procés llarg amb ajustos constants de paràmetres per obtenir els resultats desitjats.

2.2 Segmentació

Després de detectar la regió de la imatge que conté la matrícula, el següent pas va ser segmentar cadascun dels caràcters. Per aconseguir-ho, vam aplicar un procés de processament d'imatge.

Mètode 1 - Processament d'imatge i detecció de contorns

Com que totes les imatges tenien una perspectiva lateral, primer vam dur a terme una transformació de perspectiva per alinear la imatge i facilitar l'extracció de caràcters.

En primer lloc, vam identificar les coordenades de les cantonades de la matrícula detectada i les vam guardar en un array ordenat. A continuació, vam definir les dimensions de sortida desitjades i vam crear un array amb els punts de destinació segons les dimensions.

Per realitzar la transformació de perspectiva [5], vam calcular la matriu de transformació amb "cv2.getPerspectiveTransform" i la vam aplicar a la imatge amb "cv2.warpPerspective".

Tenint la imatge alineada, la vam convertir a l'espai de color HSV per extreure el canal V (value) i així normalitzar la lluminositat.

Després, vam aplicar un threshold d'OTSU i vam invertir la imatge per separar el fons dels caràcters, de manera que aquests queden en blanc sobre un fons negre, com es pot veure en la **(Figura 9)**.

Figura 9. Matrícula després del processament d'imatge

Per a la detecció de contorns [6], vam utilitzar findContours(), que identifica tots els contorns presents. Llavors vam aplicar un filtratge per caracteritzar els contorns que representaven els caràcters segons les seves proporcions.

Un cop aplicat el filtratge, vam definir les "bounding boxes" per a cada contorn que complia les condicions especificades. On a partir d'aquestes vam extreure les coordenades i la regió d'interès per a cada caràcter detectat, preparant-los així per a la fase de reconeixement.

Figura 10. Caràcters després de la segmentació

2.3 Classificació

Un cop feta la segmentació de la matrícula, s'ha de classificar els caràcters segmentats. A continuació, es plantegen tres mètodes per fer-ho:

Mètode 1 – SVC:

El mètode SVC (Support Vector Classifier) [1], es una tècnica de classificació supervisada. El seu objectiu és trobar el millor hiperplà que separa les dades en diferents classes, maximitzant el marge entre les classes, es a dir, la distància entre els punts de dades més propers de les classes, anomenats vectors de suport. És molt efectiu en problemes de classificació amb alta dimensió

Mètode 2 – KNN:

El mètode KNN (K-Nearest Neighbours) [2], és una tècnica de classificació supervisada. Emmagatzema totes les dades amb la seva corresponent etiqueta i per classificar un nou punt de dades desconegut busca els seus k veïns més propers en el espai de característiques; segons el paràmetre k definit. És molt simple i fàcil d'implementar, i s'adapta fàcilment a noves dades d'entrenament sense necessitat de tornar a entrenar el model.

Mètode 3 – CNN:

Un altre mètode per la classificació és una CNN (Convolutional Neural Network), formada per dues capes convolucionals, amb filtres de mida 3x3 i amb funcions d'activació ReLU. A continuació de cada

capa convolucional hi ha una capa de max-pooling, i finalment hi ha dues capes (fully-connected), que actuen com a classificador. És una tècnica que dona molt bons resultats classificant, però necessita una gran quantitat de dades per entrenar-se.

3. Disseny Experimental

3.1 Descripció dels datasets

En aquest projecte, en total s'han fet servir 5 datasets diferents. 2 d'ells han sigut imatges de cotxes matriculats per provar el nostre sistema. 1 s'ha utilitzat per entrenar la xarxa YOLO, i la resta s'ha utilitzat per entrenar els sistemes classificadors.

1. Dataset original (CV): Dataset proporcionat pel professorat. Consta de 17 imatges, totes fetes des de la dreta.

Figura 11. Imatge del dataset original

2 . Dataset propi: Dataset creat a partir d'imatges fetes per nosaltres. Consta de 22 imatges.

Figura 12. Imatge de la base de dades pròpia

3. Dataset YOLO: Dataset treballat a la pàgina roboflow per tal d'entrenar el sistema de detecció. Consta de 187 imatges.

Figura 13. Imatge de la pàgina Roboflow

4. Dataset sintètic: És tracta de dades que provenen de la font original de la matrícula espanyola. On s'ha aplicat un (data augmentation) rotant cada caràcter 5, 10, 15 i 20 graus des de la dreta. Format per un total de 170 imatges

Figura 14. Imatges del caràcter "1" del dataset sintètic

5. Dataset real segmentat: És tracta de dades que provenen dels resultats de la fase de segmentació del repte, s'ha seleccionat les que han sortit bé. Hi ha un total de 259 imatges

Figura 15. Imatges de caràcters segmentats del dataset

3.2 Experiments i mètriques

Experiment 1:

Per veure com funcionaven els classificadors, vam fer un petit experiment amb les dades sintètiques definides a l'apartat 3.1.

En aquest experiment únicament hem vist els resultats dels models SVC i KNN, degut a que com indicàvem la CNN necessita bastantes dades per poder donar bons resultats. Les dades sintètiques són bastant fàcils de predir, ja que, amb les que hem entrenat els models són bastant similars. Per tant, hauria de donar bons resultats. Per la comparació utilitzarem K-Folds per no dependre d'una única partició, sinó de 5 diferents. L'accuracy per mesurar en cada fold i la mitjana global dels 5 folds de cada classificador. I matrius de confusió, per observar quins caràcters confonen els mètodes.

Experiment 2:

Un cop fet el petit experiment, posarem a prova els tres models amb moltes més dades. Utilitzarem les dades sintètiques (170 imatges) juntament amb les dades segmentades (259) per comparar els tres classificadors i veure quin és el millor. En aquest cas també utilitzarem els KFolds, i, a més, farem servir mètriques com: (accuracy, intervals de confiança, diagrames de caixa i matrius de confusió). Abans de treballar conjuntament amb els dos conjunts de dades, hem de aplicar-li un processament per a que siguin iguals les imatges del dos datasets.

Experiment 3:

Després d'analitzar els diferents models, hem realitzat prediccions amb el model definitiu, que utilitza YOLO per a la localització de la matrícula, processament d'imatge per a la seva segmentació i un classificador SVC per a la predicció dels caràcters.

Per avaluar el rendiment del model, hem introduït cinc imatges al sistema per observar les prediccions generades i comparar-les mitjançant mètriques de text. Les mètriques utilitzades inclouen la distància de Levenshtein i la similitud de Levenshtein.

4. Resultats

Per dur a terme un anàlisi exhaustiu dels models implementats i determinar quin classificador s'adapta millor a les dades, s'han realitzat diversos experiments, els quals s'acompanyen dels seus resultats. Aquesta metodologia ens ha permès comparar el rendiment dels diferents classificadors i identificar quins ofereixen la millor precisió i eficiència en les tasques de localització i classificació de matrícules. A continuació, presentem els detalls dels experiments i els resultats obtinguts.

4.1 Experiment 1

	fold 1	fold 2	fold 3	fold 4	fold 5	mitja
KNN	0,53	0,37	0,27	0,43	0,34	0,39
SVC	1	0,83	0,97	0,87	0,93	0,92

Taula 1. Precisió de KNN i SVC

Veient els resultats de la **(Taula 1)**, podem observar com la precisió del classificador SVC es bastant alta, al contrari del KNN. Traiem la conclusió que al KNN potser li falten més dades per poder etiquetar correctament. Ho veurem en el pròxim experiment.

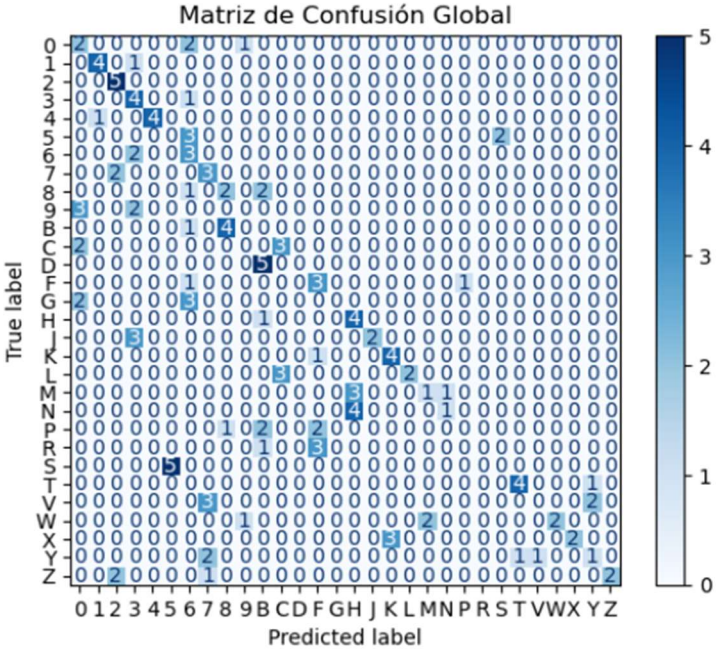

Figura 16. Matriu de confusió KNN

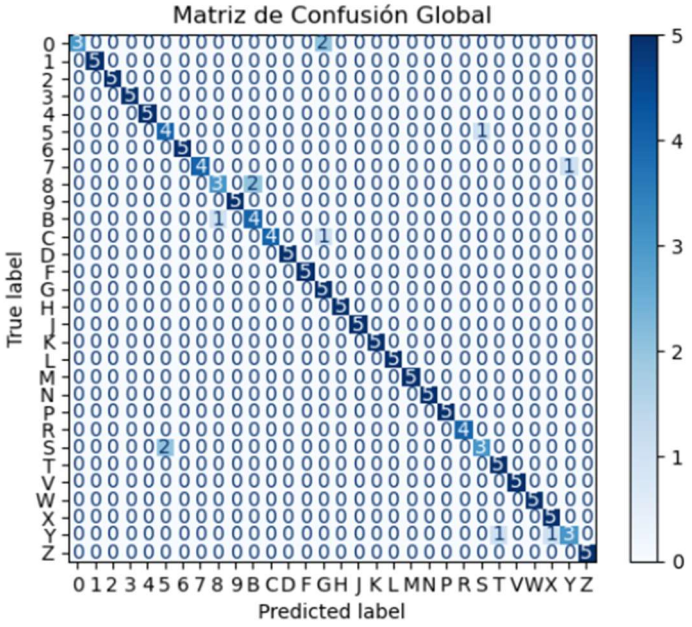

Figura 17. Matriu de confusió SVC

Comparant les dues matrius de confusió podem observar que la matriu de la **(Figura 16)** corresponent a la KNN té moltes confusions entre lletres i números, en canvi, la SVC **(Figura 17)**, no. Podem observar alguns patrons com per exemple el 0 amb la G, el 8 amb la B,…

4.2 Experiment 2

Per tal de comparar el rendiment de cada model, fem una taula amb la precisió obtinguda en cada fold pels tres classificadors.

	fold 1	fold 2	fold 3	fold 4	fold 5	mitja
KNN	0,62	0,74	0,79	0,73	0,8	0,74
SVC	0,96	0,91	1	0,95	0,94	0,95
CNN	0,9	0,93	0,93	0,91	0,9	0,91

Taula 2. Precisió de KNN, SVC i CNN

Després d'analitzar els tres classificadors, observem que, en comparar la precisió de cada model en cada fold, el classificador SVC destaca amb els millors resultats en la majoria dels folds, aconseguint una precisió mitjana del 95%.

Per la seva banda, la xarxa neuronal presenta una precisió adequadament elevada del 94%, tot i que una mica inferior al SVC.

En contrast, els resultats del classificador KNN són els més baixos, mostrant un rang de precisió més ampli, possiblement a causa de la seva sensibilitat a la distribució de les dades. La mitja de precisió per fold és 0,74.

Per visualitzar els resultats comentats anteriorment, presentem un boxplot amb els valors de precisió dels tres models. A més, els intervals de confiança ens proporcionen un rang estimat sobre la precisió esperada del model amb un nivell de confiança del 95%.

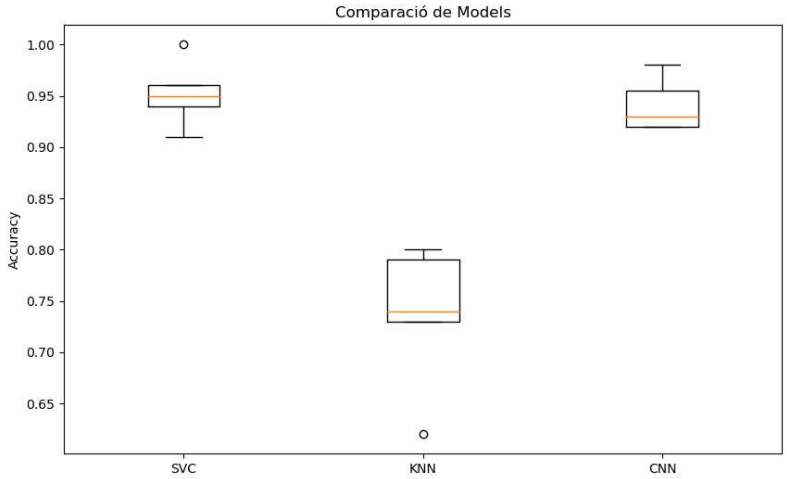

Figura 18. Diagrames de caixes

Classificador	Interval de confiança
SVC	91,41% - 99,27%
KNN	64,86% - 82,62%
CNN	92,05 - 96,34%

Taula 3. Intervals de confiança

A partir dels boxplots **(Figura 18)** i dels càlculs dels intervals de confiança **(Taula 3)** obtenim els següents resultats:

El classificador SVC presenta l'interval de confiança amb els valors de precisió més elevats, que oscil·len aproximadament entre el 91% i el 99%. El segon model amb millors resultats és la CNN, que té un interval de confiança de 92% a 96%; aquest és un rang més petit, la qual cosa indica menys variabilitat entre folds.

D'altra banda, com hem observat, el classificador KNN mostra una precisió millorable. Així, presenta un interval de confiança amb valors significativament més baixos, que arriba fins al 82,62%, amb un rang de confiança més ampli i valors més dispersos.

Una matriu de confusió és una eina que permet visualitzar el rendiment d'un model de classificació. En el cas d'un classificador amb 30 classes, la matriu de confusió serà una taula de 30x30, on cada fila representa les instàncies de la classe real, i cada columna representa les instàncies de la classe predita pel model. Això ens proporciona una representació clara de quines classes són més

propenses a ser mal classificades, informació crucial per ajustar el model o per millorar el procés de recopilació de dades.

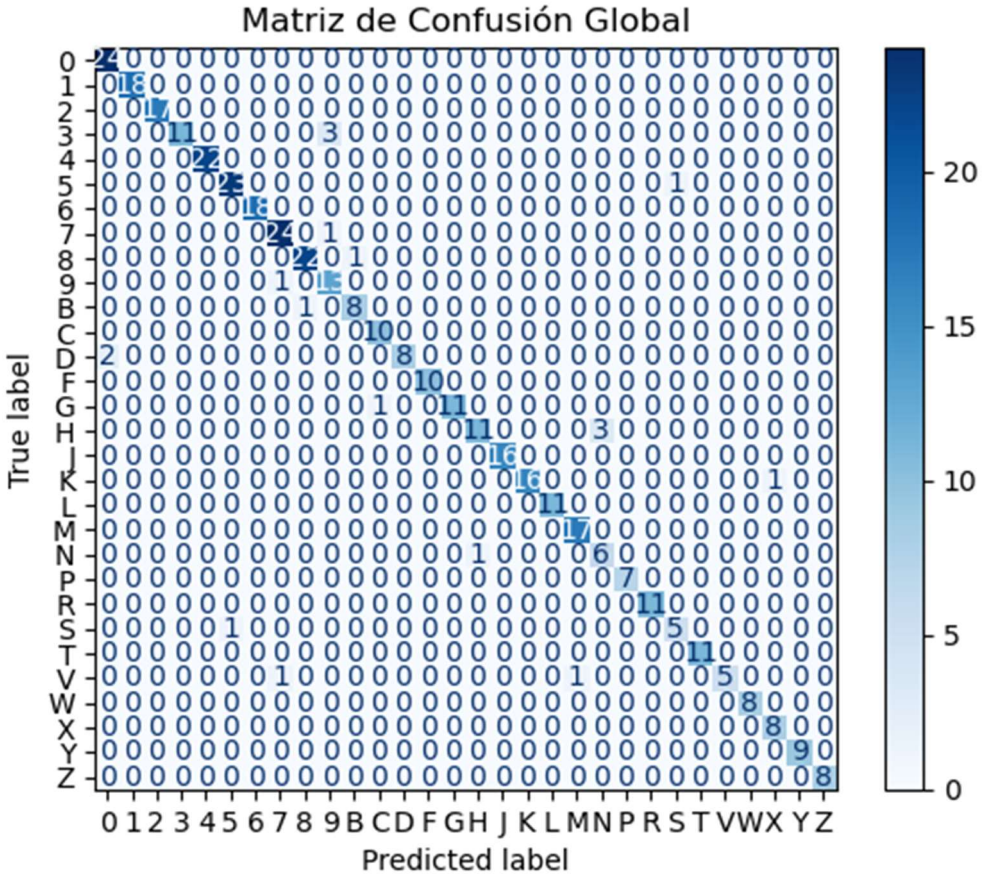

Figura 19. Matriu de confusió del model SVC

En la **(Figura 19)** s'observa que el model SVC no comet gaires errors ja que la majoria de valors es troben en la diagonal. Però si ens fixem en els errors puntuals, veiem que la lletra 'd' es confón amb el '0'. A més a més, el model prediu erroniament la 'n' en alguna ocasió quan es tracta d'una 'h'.

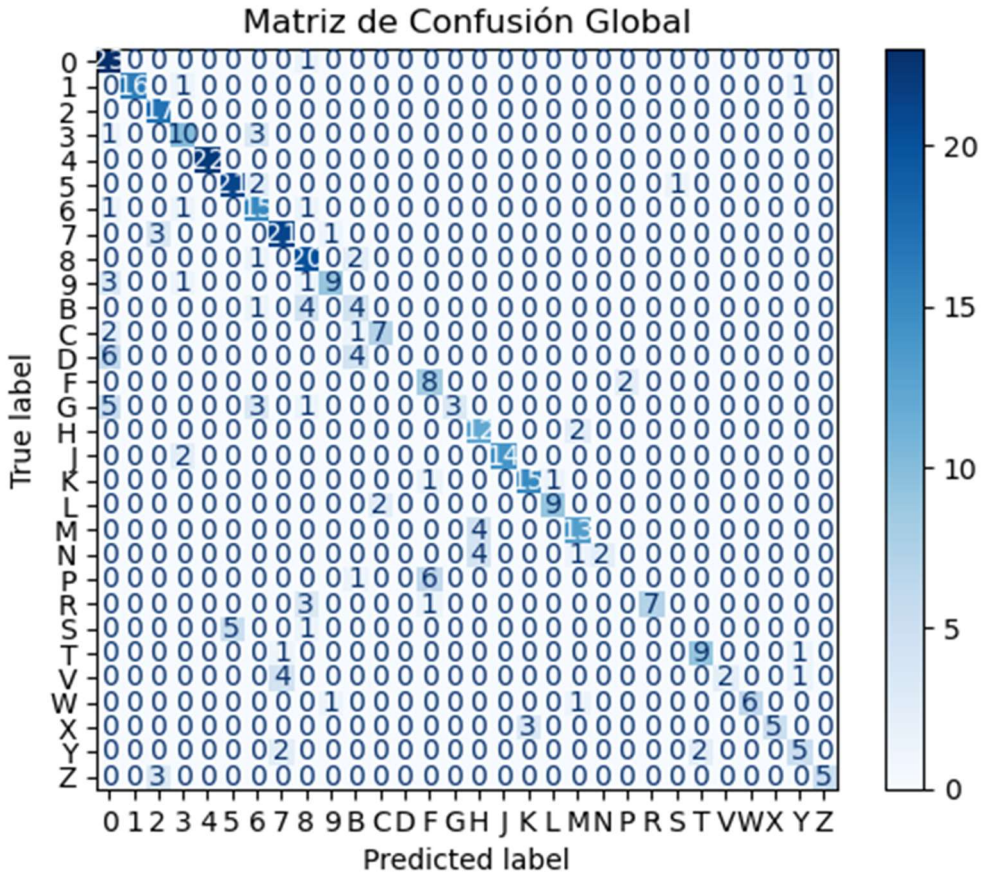

Figura 20. Matriu de confusió del model KNN

A la **(Figura 20)** podem observar que el classificador KNN presenta una major taxa d'errors en lletres amb similitud visual. Aquesta tendència és coherent amb els resultats de precisió obtinguts en l'experiment 1.

Els parells de lletres que resulta difícil de distingir per al model inclouen la 'm', la 'h' i la 'n', així com la 's' i el número '5', i la 'p' i la 'f'. A més, cal destacar la tendència del model a classificar erròniament el '0'.

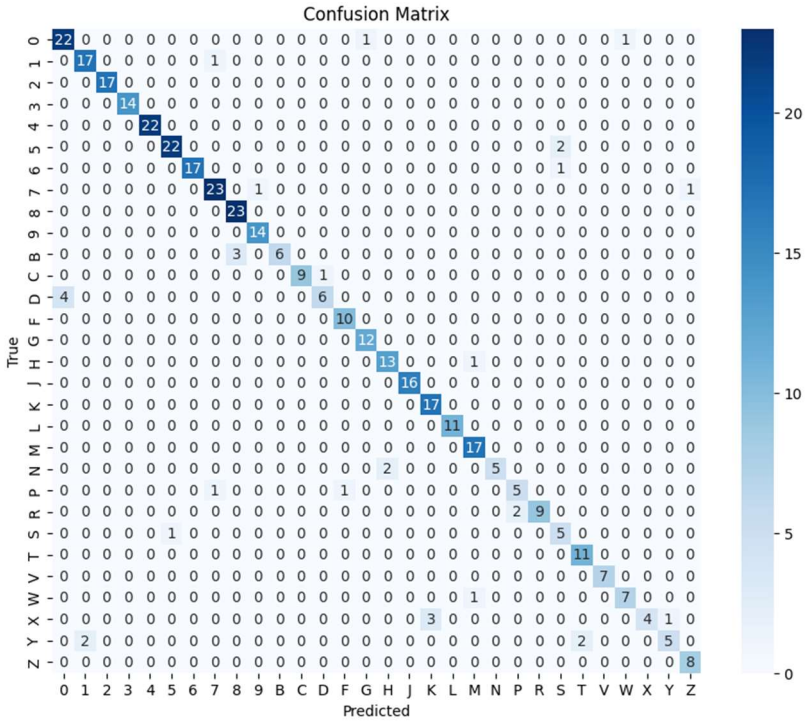

Figura 21. Matriu de confusió del model CNN

El model CNN, igual que el SVC, presenta molt bons resultats, on la majoria de prediccions son correctes. Podríem destacar alguns errors puntuals entre la ‘x’ i la ‘k’. En els tres models s’aprecia la tendència que tenen de predir la ‘d’ com un ‘0’.

4.3 Experiment 3

La distància de Levenshtein quantifica el nombre d'operacions (insercions, substitucions o eliminacions) necessàries per transformar l'etiqueta predita en l'etiqueta real. D'altra banda, la similitud de Levenshtein és una mesura similar, però normalitzada en funció de la longitud de la cadena, la qual permet obtenir una comparativa més ajustada entre les dues cadenes de text.

Amb aquestes mètriques, podem avaluar la precisió del model i identificar possibles àrees de millora en la seva capacitat de predicció.

Tot i que s’han realitzat cinc prediccions, que es troben en el fitxer ‘model_definitiu.ipynb’ en el nostre repositori de github, es mostraran només 2 exemples que representen totalment les 5 prediccions.

Figura 22. Exemple 1 de predicció del model definitiu

En la **(Figura 22)** es veu la imatge del cotxe a l'esquerra amb la matrícula a identificar, i a la dreta tenim la sortida generada pel nostre model amb la corresponent predicció.

Quatre de les cinc prediccions realitzades mostren un resultat similar a la **(Figura 22)**, totes obtenen un distància de Levenshtein de 0 i una similitud de 1. Això és perquè l'etiqueta predita és exacta a l'etiqueta real, per tant obtenim un resultat correcte.

Un altra exemple és el següent:

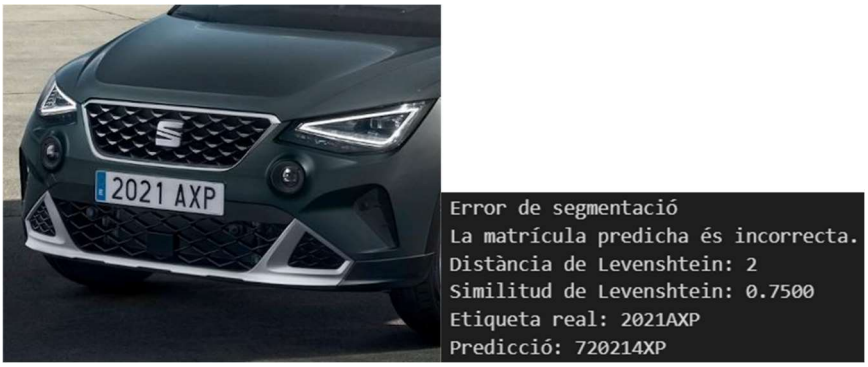

Figura 23. Exemple 2 de predicció del model definitiu

Aquest exemple mostra una distància de Levenshtein de 2, el que indica que hi ha hagut dos errors en la predicció de la matrícula. En primer lloc, el model ha predit un '4' en lloc d'una 'A'. A més, s'ha inserit un caràcter addicional al principi de la matrícula, en aquest cas un '7'.

Aquest error es classifica com un error de segmentació, ja que ha ocorregut perquè el model ha detectat el requadre blau de la lletra E europea com si fos un caràcter. Aquesta situació pot ser un indicatiu

que suggereix la necessitat de millorar el procés de segmentació del model per a evitar confusions similars en el futur.

5. Discussions i Conclusions

Creiem que el treball realitzat per a la localització i classificació de matrícules ha estat exitós, utilitzant YOLO per a la detecció i SVC com el millor classificador. No obstant això, la segmentació continua sent una àrea clau a millorar, ja que una segmentació precisa és essencial per a la posterior classificació.

Per millorar aquesta tasca, proposem implementar una xarxa neuronal convolucional (CNN) que podria augmentar la precisió del model. A més, augmentar el dataset amb més diversificació ajudaria a aconseguir un model que generalitzi millor. En el cas de la CNN per a la classificació, disposar de més dades podria ser especialment beneficiós.

A més, podríem considerar entrenar models separats per a nombres i lletres, amb l'objectiu de millorar la precisió general i reduir errors com la confusió entre la ‘d’ i el ‘0’, per exemple. Aquesta estratègia podria conduir a una classificació més precisa i fiable.

6. Referencies

[1] UMA. Accedido el 30 de septiembre de 2024. [En línea]. Disponible: https://lp2m.uma.ac.id/2024/01/20/support-vector-classifier-svc-a-comprehensive-guide/

[2] IBM. “¿Qué es KNN? | IBM”. IBM - United States. Accedido el 18 de octubre de 2024. [En línea]. Disponible: https://www.ibm.com/mx-es/topics/knn

[3] “OpenCV: Basic Operations on Images”. OpenCV documentation index. Accedido el 18 de octubre de 2024. [En línea]. Disponible: https://docs.opencv.org/4.1.0/d3/df2/tutorial_py_basic_ops.html

[4] “OpenCV: Image Thresholding”. OpenCV documentation index. Accedido el 18 de octubre de 2024. [En línea]. Disponible: https://docs.opencv.org/4.1.0/d7/d4d/tutorial_py_thresholding.html

[5] “OpenCV: Morphological Transformations”. OpenCV documentation index. Accedido el 18 de octubre de 2024. [En línea]. Disponible: https://docs.opencv.org/4.1.0/d9/d61/tutorial_py_morphological_ops.html

[6] Ultralytics. “Detect”. Home - Ultralytics YOLO Docs. Accedido el 18 de octubre de 2024. [En línea]. Disponible: https://docs.ultralytics.com/tasks/detect/

[7] “spanish-license-plates Object Detection Dataset (v3, 2022-11-15 6:57pm) by licenseplates”. Roboflow. Accedido el 18 de octubre de 2024. [En línea]. Disponible: https://universe.roboflow.com/licenseplates-h9qfr/spanish-license-plates/dataset/3