

Reconeixement de matrícules de vehicles

Pau Bofill, Lara Castillejo, Júlia Lipin Gener

Abstract—Aquest projecte presenta el desenvolupament d'un sistema automàtic de reconeixement de matrícules de vehicles estacionats a la via pública mitjançant tècniques de visió per computador i Deep Learning. L'arquitectura del sistema combina metodologies clàssiques de processament d'imatges amb xarxes neuronals avançades, integrant-se en un robot mòbil dissenyat per patricular zones d'estacionament. El sistema identifica vehicles, localitza la matrícula dins la imatge i reconeix els caràcters que la componen. Aquesta eina ofereix una solució eficient i escalable per a la gestió automatitzada d'espais d'aparcament i la verificació del pagament associat.

Keywords—Reconeixement de matrícules, Visió per computador, YOLOv8, OCR, Vehicles estacionats, Robòtica.

poder detectar canvis bruscos a la imatge, com ara les zones de contrast blanc-negre o les formes rectangulars. Per a aquesta tasca se sol utilitzar l'algorisme *Canny*^[2] al ser modern i robust. La transformada de *Hough*^[2] és ampliament utilitzada per poder reconèixer formes geomètriques, on en aquest cas, la matrícula té una forma rectangular i es pot delimitar la seva regió d'interès amb força precisió. Per últim, en quant als caràcters que formen la matrícula, s'aplica morfologia matemàtica per, donat un llindar i una binarització de la imatge, eliminar el soroll o bé unir trossos de caràcters que no han sigut captats del tot bé des d'un inici, aplicant dilatació, erosió, obertura o tancament.

2.3 Tècniques de Deep Learning

En quant a la fase de *Deep Learning*, les xarxes neuronals que presenten una eficàcia major són *YOLO*^[3] i *Faster R-CNN*. Per una banda, el model *YOLO* (*You Only Look Once*) processa la imatge un sol cop per detectar múltiples objectes i les seves ubicacions, divideix la imatge en cel·les i, donades aquestes ubicacions, fa una predicció dels objectes que hi poden haver. En canvi, el model *Faster R-CNN* consta d'una *CNN*^[4] (*Convolutional Neural Network*, xarxa base) que extreu les característiques de la imatge, i una segona xarxa *Region Proposal Network* que capta possibles regions d'interès. Tots dos models s'utilitzen per a la detecció de la matrícula del cotxe, tot i que la diferència radica en que *YOLO* és molt ràpid i actua gairebé en temps real, mentre que *Faster R-CNN* és més lent però més precís, al poder detectar matrícules en situacions més complexes. Pel que fa a la detecció de caràcters dins la matrícula, s'utilitzen combinacions de *CNN* i *RNN* (*Recurrent Neural Network*). Mentre que el model *CNN* capta el contingut general de la imatge, el model *RNN* tracta les dades seqüencials dins la imatge processada amb *CNN*, i entén l'ordre de la seqüència detectada. Aquesta combinació s'anomena *CRNN* i assegura una bona visió general i una bona comprensió de la seqüència.

1 INTRODUCCIÓ

El reconeixement de matrícules de vehicles (també denominat generalment com *Automatic Number Plate Recognition*)^[1] és una eina molt present avui dia en el món de la visió per computador, la qual podem trobar tant en sistemes de control de trànsit a les carreteres i autopistes com ara en gestió de zones d'estacionament, entre d'altres situacions quotidianes.

L'objectiu principal d'aquest projecte és implementar una solució capaç d'identificar matrícules de cotxes estacionats a la via pública de manera precisa mitjançant tècniques de visió per computador, tant clàssiques com de *Deep Learning*. La nostra motivació per desenvolupar aquesta eina ha estat tant tècnica com social. Per la part tècnica se'ns presenta un nou repte enfocat a la intel·ligència artificial, que haurem de solucionar tenint en compte les tècniques que se'ns han proporcionat a l'assignatura. Per altra banda, la motivació social consisteix en la viabilitat i utilitat que pot tindre al món real.

2 ESTAT DE L'ART

2.1 Tècniques més utilitzades

Els sistemes *ANPR* (*Automatic Number Plate Recognition*) actuals combinen tant tècniques clàssiques de processament d'imatges com mètodes moderns, com poden ser les xarxes neuronals. Per norma general, aquests sistemes segueixen els següents passos, en l'ordre especificat:

- Detecció del vehicle
- Localització de la matrícula
- Segmentació dels caràcters de la matrícula
- Reconeixement de text (*OCR*)

2.2 Tècniques clàssiques

Dins les tècniques clàssiques, hi trobem la detecció de vores per

3 EXPERIMENTS I RESULTATS

La proposta del sistema a desenvolupar es fonamenta en tres etapes que, de forma consecutiva, ens donen com a producte final reconèixer matrícules de cotxe de manera eficient i precisa.

1. Detecció de vehicles

Amb una primera xarxa neuronal *YOLOv8*^[5] entrenada per a la detecció d'objectes, s'identificaran els vehicles i la posició d'aquests per cada *frame*.

2. Localització de la matrícula

Un cop s'hagi detectat el vehicle, s'aplicaran diverses tècniques clàssiques de visió per computador, com ara detecció de contorns i morfologia matemàtica per poder localitzar la matrícula dins la regió del vehicle detectada a l'apartat anterior.

3. Reconeixement de text (*OCR*)

Finalment, s'aplicarà una última i tercera xarxa neuronal *CNN* entrenada per a la detecció de caràcters dins d'una matrícula, per poder extreure i identificar de forma ordenada els caràcters continguts a la matrícula.

3.1 Bases de dades

Per tal de poder avaluar l'eficàcia de l'execució completa del model, s'ha creat un conjunt d'imatges etiquetades amb les matrícules corresponents als cotxes que s'hi mostren.

Per a la validació del model de detecció de cotxes, s'ha utilitzat el conjunt de validació de la base de dades amb la que s'ha entrenat el model, de la mateixa forma que pel model de OCR.

L'algorisme de detecció de matrícules ha estat avaluat usant una part curada del conjunt de dades Car Plate Detection ^[6] el qual conté *labels* amb les *Bounding-Box* de les matrícules

3.2 Detecció de vehicles

La primera etapa del projecte consisteix en identificar els vehicles dins del context de la via pública. Per dur a terme aquesta tasca, s'ha escollit el model *YOLOv8*, una evolució de *YOLO (You Only Look Once)*, reconeguda per la seva capacitat de detecció d'objectes en temps real. Aquest model es caracteritza per la seva velocitat i precisió gràcies a una arquitectura optimitzada que permet processar imatges en temps real, dividint-les en *Bounding-Boxes* i proposant la ubicació i classificació dels objectes detectats.

Hem començat per entendre l'estructura dels directoris d'una base de dades que és entrenada amb *YOLOv8*. L'ordre dels directoris a *YOLOv8* és primordial per poder accedir a les imatges o labels correctament, ja sigui del conjunt d'entrenament o de validació. En el cas que el model vulgui accedir a les labels de train, haurà de anar primer al directori *labels* i, després, al directori *train*. Dins el conjunt d'entrenament es presenten totes les imatges que volem entrenar, mentre que dins el conjunt de validació s'hi troben els casos que coincideixen amb els resultats esperats. Tot i així, la nostra base de dades original constava de identificació de vehicles generals, pel que incloïa busos, motocicletes, camions i furgonetes, a més del turisme convencional. Com que per al nostre projecte només ens interessa saber si hi ha o no un turisme a l'escena proporcionada, s'ha fet un filtratge per treure tots els demés vehicles, tant de els directori d'imatges com de labels, per ajustar-ho al nostre problema.

Donat el filtratge de la base de dades sobre la que hem treballat, s'ha hagut d'editar el nombre de classes que descriuran les *Bounding-Boxes* del nostre model. *YOLOv8* compta amb un fitxer *config.yaml* que descriu al model com està estructurat el conjunt de dades durant l'entrenament o inferència, i el nombre de classes a identificar, per tant, aquest fitxer s'ha ajustat per poder garantir un bon funcionament del model.

Un cop ja s'ha preparat la base de dades que donarà pas a la primera fase del nostre detector de matrícules, entrenem el model. Amb la llibreria *ultralytics* s'ha importat el nostre model *YOLOv8n* per poder entrenar la nostra base de dades adaptada i poder crear una xarxa neuronal capaç de poder identificar cotxes de forma robusta. El model s'ha sigut entrenat amb 15 èpoques i un batch size de 8, el qual ha sigut decisiu per trobar un bon equilibri entre precisió i temps emprat alhora d'entrenar el nostre model. Al finalitzar, es genera un arxíu *best.pt*, que obté el model

que ha obtingut el millor rendiment durant el procés d'entrenament, pel que serà utilitzat alhora d'utilitzar de comprovar el seu funcionament.

Com a imatge de test hem agafat imatges on es veïés més d'un cotxe, per verificar quan precís és el nostre model entrenat ja no només per un cotxe en primer pla, sinó també per altres cotxes a l'escena que es troben a un pla secundari. Aquest ha sigut el resultat per al test realitzat:



Fig 1. Imatge original per al test d'aquesta fase.



Fig 2. Sortida generada pel model entrenat amb *YOLOv8n*.

podem veure com el nostre model detector de cotxes és altament fiable tant en objectes a primer pla com en objectes en segon pla. No obstant, per a la localització de la matrícula s'ha treballat amb el cotxe trobat a primer pla, és a dir, el cotxe amb una confiança major i una àrea de *Bounding-Box* major.

3.3 Localització de la matrícula

Un cop s'ha detectat el cotxe principal a l'escena, el següent pas és localitzar la matrícula dins la seva àrea. Aquesta fase s'ha dut a terme mitjançant tècniques clàssiques de processament d'imatge.

El primer pas consisteix a millorar el contrast i la nitidesa de les imatges per tal de facilitar l'extracció de contorns i vores. Es converteix la imatge a escala de grisos i s'aplica un *CLAHE* ^[7] (*Contrast Limited Adaptive Histogram Equalization*) per ressaltar els detalls. A continuació, s'aplica una correcció *gamma* per millorar la brillantor i es fa servir un filtre bilateral per eliminar soroll mantenint les vores.

Després de la fase de preprocessament, es procedeix a l'extracció de contorns per identificar possibles regions candidates a contenir matrícula. Inicialment, es va considerar l'ús de l'algorisme de *Canny* per a la detecció de vores, donat que és un mètode àmpliament utilitzat per la seva precisió i robustesa davant el soroll. No obstant això, tal i com s'observa a la Fig. 3, aquest enfocament no resulta òptim per a la nostra aplicació.



Fig 3. D'esquerra a dreta: Imatge original utilitzada. Aplicació filtre Canny per a detecció de vores.

Tot i que *Canny* genera contorns fins i ben definits, aquests tendeixen a fragmentar les vores de la matrícula i a ressaltar una gran quantitat de detalls innecessaris, com ara línies del vehicle, logotips, reflexos i textures del fons. Aquesta sobrecàrrega d'informació fa difícil d'identificar una regió rectangular clara i compacta corresponent a la matrícula.

Per aquesta raó, es va optar per un enfocament alternatiu basat en el filtre de *Sobel* en direcció horitzontal, combinat amb operacions morfològiques. El filtre *Sobel* calcula el gradient d'intensitat vertical, resultant especialment efectiu en la detecció de lletres negres sobre fons clar. A la Fig. 4 es pot apreciar com aquest mètode destaca de forma més neta l'àrea rectangular de la matrícula, generant vores més consistents i menys sorolloses.



Fig 4. D'esquerra a dreta: Imatge original utilitzada. Aplicació filtre Sobel i morfologia matemàtica per a detecció de vores.

Aplicant la operació morfològica *close* (dilatació seguida d'erosió), s'unifiquen les vores pròximes i es tanquen petits vuits, cosa que facilita l'extracció d'un contorn rectangular coherent.

Un cop s'han extret els contorns de la imatge, es calcula un *score* per a cada contorn, basant-se en *aspect ratio* (ha de ser rectangular), densitat (proporció de píxels dins la caixa) i àrea total. Només els contorns amb característiques similars a les de les etiquetes amb *One-Hot Encoder* per a la classificació multiclasse. S'ha dissenyat una arquitectura *CNN* amb les següents característiques:

- **Blocs convolucionals Conv2D** (per extreure característiques locals importants de la imatge)
- **MaxPooling** (disminueix càrrega computacional i fa el model més robust)
- **Flatten** (converteix la sortida de Conv2D a un vector unidimensional)
- **Capa densa ReLU^[11]** (no lineal i evita gradient desaparegut)
- **Dropout** (evita *overfitting* eliminant neurones)
- **Softmax** (transforma sortida en probabilitats sobre les classes possibles)



Fig 5. Sortida de la funció de test de l'algorisme de detecció de regions d'interès de la matrícula, mostrant màxim i mínim error de

l'algorisme.

S'observa com el nostre algorisme funciona correctament en la detecció de les regions de matrícula dins d'un cotxe, en els millor dels casos quan el cotxe es troba de front a la càmera, i en el pitjor quan la perspectiva entra en escena i, tot i així, és capaç de detectar la matrícula del cotxe exitosament. Per demostrar la seva validesa en un cas real, utilitzem la imatge de test validada en la primera fase de detecció de cotxes:



Fig 6. Sortida generada per l'algorisme de detecció de matrícules implementat sobre la imatge de test.

Es pot veure clarament que l'algorisme implementat funciona de la manera esperada en un cotxe amb matrícula espanyola.

3.4 Reconeixement de text (OCR)

El pas més important d'aquest projecte ve donat per la fase *OCR* (Reconeixement de caràcters), per poder extreure la informació dins la regió de la matrícula trobada a la fase anterior. Per a aquesta fase s'ha desenvolupat un sistema híbrid que combina tècniques clàssiques de processament d'imatge amb un model de *Deep Learning* basat en una *CNN* (xarxa neuronal convolucional)^[8]

Per entrenar el model de reconeixement de caràcters s'ha utilitzat el *dataset Chars74k*^[9], que conté imatges de caràcters individuals amb múltiples variacions. Tot i així, per ajustar al màxim el reconeixement de caràcters, s'ha ampliat la base de dades retallant caràcters de matrícules espanyoles i afegint-la a la base de dades, partint d'una pàgina en línia de generació de matrícules de vehicles d'ús gratuït.^[10]

Un cop s'ha definit la base de dades a entrenar el model *CNN*, s'ha realitzat un preprocesament enfocant-se en la redimensió de les imatges (32x32 píxels), normalitzar els valors de píxels i la decodificació de les etiquetes amb *One-Hot Encoder* per a la classificació multiclasse.

S'ha dissenyat una arquitectura *CNN* amb les següents característiques:

Per a l'entrenament, s'ha utilitzat l'optimitzador *Adam*^[12] per a un entrenament més ràpid i estable, i s'ha entrenat amb 20 èpoques i un *batch size* de 16 que, després de diverses proves, ha sigut el conjunt més eficient i amb millors resultats.

validació, per determinar en quines imatges s'ha localitzat correctament la matrícula.

2. La distància cartesiana entre el valor trobat i el valor real, per mesurar l'error de localització.

En quant als mètodes clàssics, s'ha implementat segmentació^[13] Amb aquestes mètriques, i en un conjunt de dades de 250 per cada caràcter de la matrícula detectada, aplicant filtre *Gaussià* i imatges, el sistema ha identificat correctament la matrícula en un per eliminar soroll, binarització *Otsu*, detectant contorns i 80% dels casos, amb un error mitjà de 15 píxels.

redimensionant la imatge a 32x32 píxels, sense perdre l'*aspect ratio* original dels caràcters, per evitar confusions. Com a afegit, El reconeixement de caràcters OCR, avaluat sobre una partició s'ha implementat una funció que busqui zones blaves a la del conjunt de dades amb el qual es va entrenar, ha obtingut un matrícula (regió del país de la matrícula) per aplicar un crop a la *accuracy* del 99% regió de la matrícula i detectar els caràcters amb major eficiència,

ja que el país en aquest cas ens és redundant, al treballar solament Finalment, utilitzant el nostre propi conjunt de dades de 150 amb matrícules espanyoles.

Cada caràcter segmentat s'introdueix en el model *CNN* per predir la seva classe, i la predicció per tota la matrícula es concatena en

una cadena de text. Per millorar la precisió del reconeixement, s'implementa una funció de correcció basada en el format vàlid d'una matrícula espanyola (quatre dígitos seguits de tres lletres).

Aquesta funció corregeix errors comuns entre caràcters, visualment similars en casos en que, durant els 4 primers dígitos, es classifiqués un caràcter com una lletra i no pogués ser possible (per exemple, 'L' per '4', 'S' per '5').

Com a resultat de la segmentació, podem veure a la Fig 7 com el nostre OCR actua de la manera esperada, detectant correctament els caràcters i assignant-los a la classe corresponent.



Fig 7. Segmentació de caràcters i assignació de classe sobre la imatge de test escollida.

Finalment, de l'implementació de les tres fases dutes a terme en aquest projecte, es mostra a la Fig 8 que el algoritmes i models entrenats satisfan els requisits proposats i donen els resultats esperats.



Fig 8. Output final de les tres fases de detecció concatenades.

3.5 Resultats d'avaluació de mètriques

Per a la detecció de cotxes, el nostre model ha estat avaluat amb un *set* de validació de 53 imatges, assolint una precisió del 87,9% i un *recall* de 85,7%.

Pel que fa al retall de les matrícules, s'han emprat dues mètriques principals:

1. La superposició entre el *Bounding-Box* trobat i el de

4 CONCLUSIONS

Com a resultat final de la implementació de les tres fases dutes a terme en aquest projecte, podem veure que el sistema desenvolupat és capaç de detectar cotxes, localitzar les seves matrícules i reconèixer correctament els caràcters que les componen. A més, s'ha demostrat que la combinació de tècniques de visió per computador amb models de *Deep Learning* pot oferir una solució robusta i eficient, fins i tot en condicions de perspectiva diversa.

Tot i això, cal destacar que el sistema pot veure's limitat en casos de matrícules molt deteriorades o condicions ambientals extremes, pel que són situacions que s'haurien d'abordar mitjançant models més avançats.

En definitiva, aquest projecte ha complert els seus objectius i representa una base sòlida per a sistemes de reconeixement de matrícules aplicables a escenaris reals, especialment en contextos de control d'aparcament.

BIBLIOGRAFIA

- [1] Navacerrada, Jorge. "Sistema de detección de matrículas con OpenCV". Universidad Politécnica de Madrid.
- [2] Muñoz Manso, Roberto . "Sistema de visión artificial para la detección y lectura de matrículas". Universidad de Valladolid.
- [3] Redmon, Joseph. Farhadi, Ali. "YOLOv3: An Incremental Improvement." University of Washington.
- [4] Zherzdev, Sergey. Gruzdev, Alexey. "LPRNet: License Plate Recognition Via Deep Neural Networks.". IOTG Computer Vision Group.
- [5] Muhammad Yaseen. "What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector" National University of Computer and Emerging Science.
- [6] Car plate detection :

<https://www.kaggle.com/datasets/andrewmvd/car-plate-detection/data>

- [7] Purnawarman Mus, Farid Al Rafi, Missa Lamsani. “A Review: Contrast-Limited Adaptive Histogram Equalization (CLAHE) methods to help the application of face recognition”. Universitas Gunadarma, Indonesia.
- [8] Keiron O’Shea, Ryan Nash. “An introduction to Convolutional Neural Networks”
- [9] Chars74K-Digital-English-Font,
(<https://www.kaggle.com/datasets/supreethrao/chars74kdigitalenglishfont>)
- [10] Creador de matrículas de coches personalizadas gratuito.
(<https://matriculasdelmundo.com/creador.php>)
- [11] Takio Kurita. “Improvement of learning for CNN with ReLU activation by sparse regularization”. Hiroshima University.
- [12] Diederik P.Kingma, Jimmy Lei Ba. “Adam: A method for stochastic optimization”. University of Amsterdam, University of Toronto.
- [13] Rojas Camacho, Oswaldo. “Segmentación paramétrica de células aplicada a imágenes de microscopía”. Universidad Politécnica de Madrid.

El codi implementat per a la realització d’aquest microprojecte es troba al següent enllaç:

<https://github.com/laranyeta/Reconeixement-Matricules>