

BusVigia 2.0: detección de infracciones en carril bus

Javier Martínez^{1,*}

¹Universidad Rey Juan Carlos, j.martinezma.2018@alumnos.urjc.es

12 de febrero de 2024

1. Introducción

Las ciudades de gran escala se ven continuamente afectadas por problemas de tráfico como atascos o accidentes, lo que provoca que la movilidad sufra fuertes demoras. Una posible solución radica en la correcta gestión del transporte público. Los autobuses urbanos son los más afectados por el tráfico de la ciudad, debido a que comparten la calzada con vehículos privados.

Los carriles exclusivos para autobuses se han diseñado con el objetivo de mejorar la eficiencia del transporte público y el flujo general del tráfico. Sin embargo, estos carriles suelen ser ocupados por vehículos privados y los autobuses se ven obligados a utilizar los carriles comunes. Para evitar este problema se instalan separadores físicos sobre la línea del carril bus que, a pesar de ser una solución satisfactoria, son peligrosos y provocan accidentes.

En [Fernández-López et al., 2013] se presenta un sistema basado en visión artificial capaz de detectar, de forma automática, vehículos que obstaculizan el carril bus. El sistema se instala en los autobuses y consigue un gran efecto disuasorio, debido a que todas las infracciones captadas son sancionadas con una multa. El uso de esta tecnología hace que ya no sean necesarios los separadores físicos, lo que mejora la movilidad general y la seguridad vial de las ciudades.

El algoritmo usado en [Fernández-López et al., 2013] está basado en técnicas clásicas de procesamiento de imágenes. En la última década, la visión artificial ha experimentado un gran auge en cuanto a precisión debido a los nuevos métodos basados en aprendizaje profundo. En este trabajo se propone un sistema que, haciendo uso de los nuevos métodos, mejora la eficacia de [Fernández-López et al., 2013] en la detección de infracciones.

2. Trabajos previos

El algoritmo de detección de infracciones está basado en la siguiente hipótesis: *si el autobús ha tenido que abandonar el carril bus, existe un vehículo obstaculizando. Entonces, se comienza un proceso de detección para obtener la matrícula del infractor.* Sabiendo esto, se puede dividir el problema en los siguientes sub-problemas conocidos en la literatura de la visión artificial: detección de carril y reconocimiento automático de matrículas (ALPR).

2.1. Detección de carril

La detección de carril tiene como objetivo identificar y localizar las líneas límite de uno o varios carriles. Tradicionalmente, los métodos de detección estaban basados en la extracción de

características por color [Chiu and Lin, 2005] o bordes [López et al., 2010], combinándose con la transformada de Hough [Liu et al., 2010] o el filtro de Kalman [Danescu and Nedevschi, 2009]. Sin embargo, son algoritmos que sufren mucho frente a cambios de iluminación, cambios ambientales u occlusiones. Las técnicas basadas en aprendizaje profundo son más robustas y precisas. Se pueden dividir en dos categorías: *dos etapas* y *una etapa*.

Métodos de dos etapas. Los métodos de dos etapas [Kim and Lee, 2014, Pan et al., 2018, Neven et al., 2018] fueron los sucesores de las técnicas tradicionales. Consisten en predecir una máscara de las líneas del carril y después ajustar modelos como una parábola o un *spline* a la máscara. El trabajo [Kim and Lee, 2014] fue uno de los pioneros, se basa en usar una *Convolutional Neural Network* (CNN) para predecir los píxeles que pertenecen a las líneas de carril (Figura 1) y posteriormente aplicar el algoritmo *random sample consensus* (RANSAC) para localizar las líneas agrupando píxeles. No obstante, el problema de los métodos de dos etapas es que son más costosos computacionalmente que los de una etapa y, además, menos precisos porque los parámetros de la red de la etapa inicial no están optimizados para la segunda etapa.

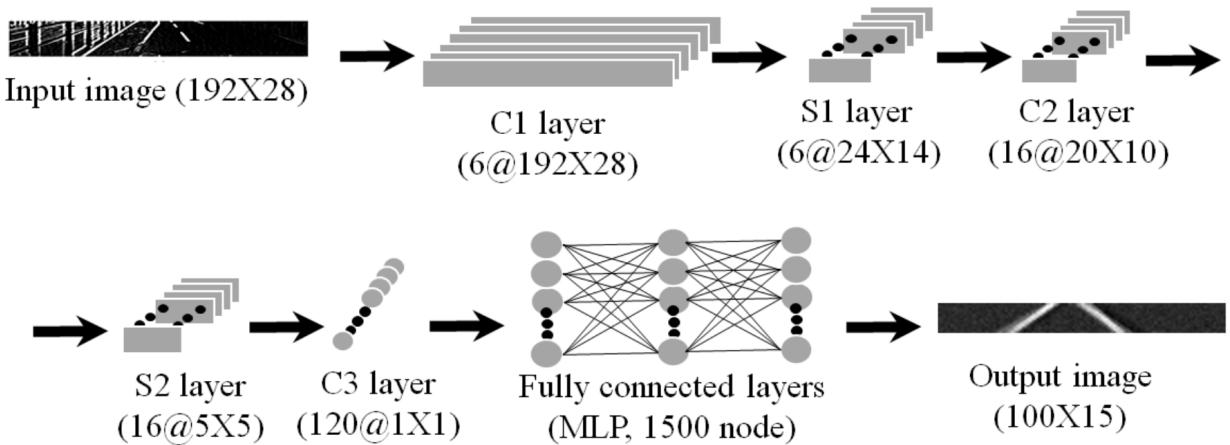


Figura 1: Arquitectura de CNN para predecir máscara de carril en [Kim and Lee, 2014].

Métodos de una etapa. Los métodos de una etapa ofrecen directamente el modelo de las líneas a partir de la imagen de entrada. Son los métodos que actualmente consiguen mejores resultados y tienen mayor velocidad. En [Van Gansbeke et al., 2019] se propone un método que, usando una sola arquitectura neuronal, hace la detección y la regresión de los parámetros de las líneas del carril. La arquitectura consiste en dos componentes: una red neuronal profunda que predice un mapa de pesos similar a la segmentación por cada línea, y un módulo de ajuste de mínimos cuadrados diferenciable que devuelve para cada mapa los parámetros de la curva que mejor se aproxime. Por lo tanto, la red aprende a generar características que evitan inestabilidades durante la etapa de ajuste de los modelos a las líneas.

LaneATT [Tabelini et al., 2021] propone un método basado en *anchors* típico en detección de objetos (YOLO [Redmon et al., 2016], SSD [Liu et al., 2016]), pero en vez de usar *bounding boxes* usa líneas. Define un *anchor* como una *línea virtual* en el plano imagen compuesta por un punto de origen y una dirección. La Figura 2 muestra un esquema general del método. Se usa una CNN (ResNet [He et al., 2016]) como *backbone* para extraer características de la imagen de entrada, después cada *anchor* se proyecta sobre el mapa de características, esta proyección se concatena con otro conjunto de características generado por un modelo de atención y, por último, hay dos cabezales que realizan las predicciones finales: uno de clasificación para dar la probabilidad de

ser línea, y otro de regresión para ofrecer los parámetros de localización de la línea. Se utiliza el modelo de atención, que es una red densa que genera características que agregan información global, porque la información del vector de características de la proyección suele ser información local y no es suficiente para realizar predicciones en situaciones con occlusiones o líneas de carril borradas.

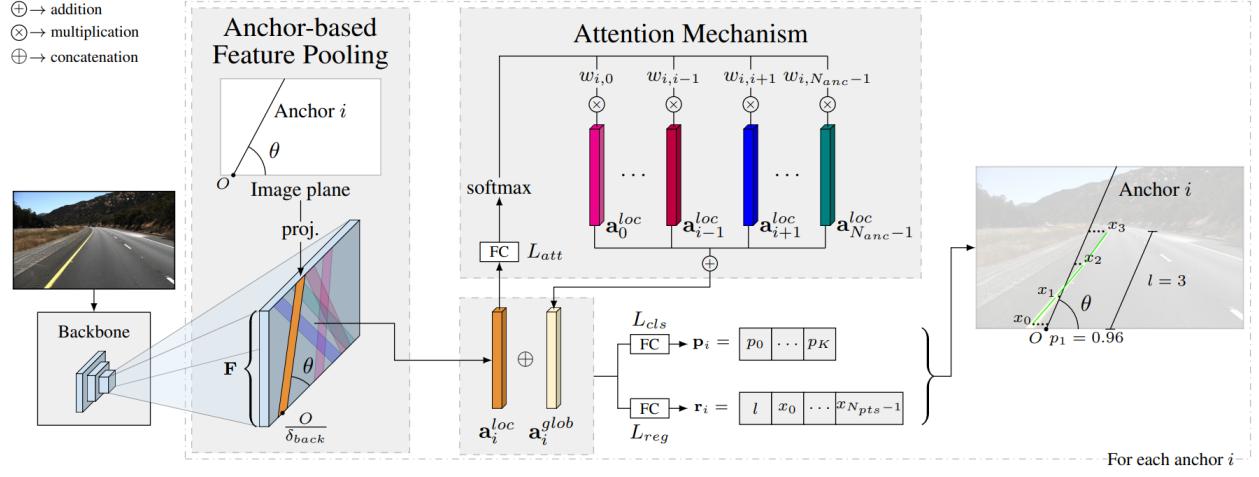


Figura 2: Esquema del método LaneATT [Tabelini et al., 2021].

CLRNet [Zheng et al., 2022] se basa en la misma idea que LaneATT [Tabelini et al., 2021], pero utiliza una pirámide de características para obtener la información local y global y, de esta manera, mejora la precisión de la localización. UFLD [Qin et al., 2020] propone el método más rápido del estado del arte. Como se muestra en la Figura 3, consiste en dividir la imagen en filas, cada fila en diferentes celdas y por cada celda ofrecer una probabilidad de contener línea o no. Cada celda se corresponde con un píxel del tensor de salida de una CNN, en este caso usan ResNet [He et al., 2016]. De esta manera, sólo se están clasificando regiones que indican la localización de las líneas en cada fila y, por lo tanto, se está reduciendo el coste computacional. Además, en la función de pérdida se utiliza información a priori de la forma y dirección de las líneas para solucionar problemas de occlusiones. Es un método muy rápido, pero no tan preciso comparado con las demás técnicas.

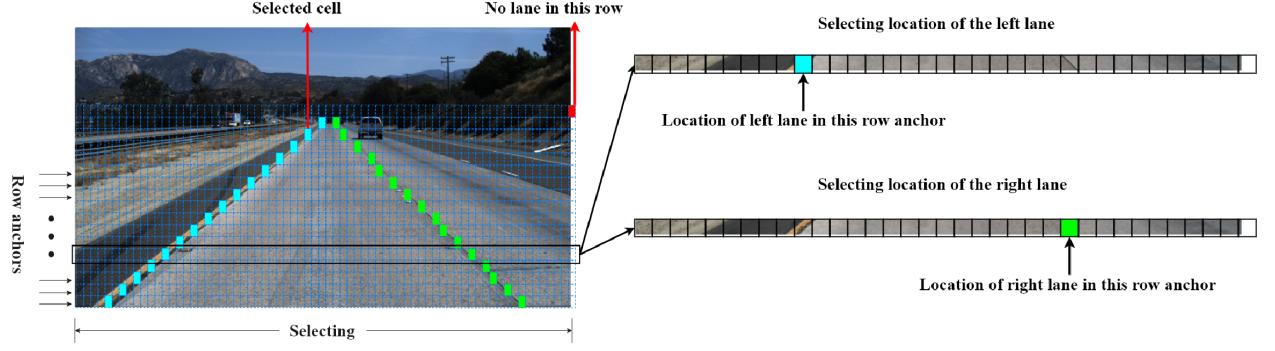


Figura 3: Algoritmo basado en filas de UFLD [Qin et al., 2020].

En la Tabla 1 se realiza una comparativa de precisión y rendimiento de los métodos explicados en esta sección. Para compararlos se ha usado el dataset CULane [Pan et al., 2018] (Figura 4)

Method	Backbone	FPS	Normal	Crowded	Dazzle	Shadow	No line	Arrow	Curve	Cross	Night
SCNN [Pan et al., 2018]	VGG16	7.5	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10
LaneATT [Tabelini et al., 2021]	ResNet18	153	91.17	72.71	65.82	68.03	49.13	87.82	63.75	1020	68.58
LaneATT [Tabelini et al., 2021]	ResNet34	129	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72
LaneATT [Tabelini et al., 2021]	ResNet122	20	91.74	76.16	69.47	76.31	50.46	86.29	64.05	1264	70.81
UFLD [Qin et al., 2020]	ResNet18	282	87.70	66.00	58.40	62.80	40.20	81.00	57.90	1743	62.10
UFLD [Qin et al., 2020]	ResNet34	170	90.70	70.20	59.50	69.30	44.40	85.70	69.50	2037	66.70
CLRNet [Zheng et al., 2022]	ResNet18	119	93.30	78.33	73.71	79.66	53.14	90.25	71.56	1321	75.11
CLRNet [Zheng et al., 2022]	ResNet34	103	93.49	78.06	74.57	79.92	54.01	90.59	72.77	1216	75.02
CLRNet [Zheng et al., 2022]	ResNet101	46	93.85	78.78	72.49	82.33	54.50	89.79	75.57	1262	75.51
CLRNet [Zheng et al., 2022]	DLA34	94	93.73	79.59	75.30	82.51	54.58	90.62	74.13	1155	75.37

Cuadro 1: Evaluación de los métodos de detección de carril en el dataset CULane [Pan et al., 2018] usando la métrica F1. En la categoría *Cross* se muestra el número de falsos positivos. Para la comparación de FPS se ha usado una GPU NVIDIA 1080Ti.

porque contiene imágenes urbanas y el objetivo de *BusVigia 2.0* es obtener la mayor precisión en esos entornos. Además, es el dataset más popular del estado del arte junto a Tusimple [TuSimple,], que está compuesto por imágenes capturadas en autopistas.

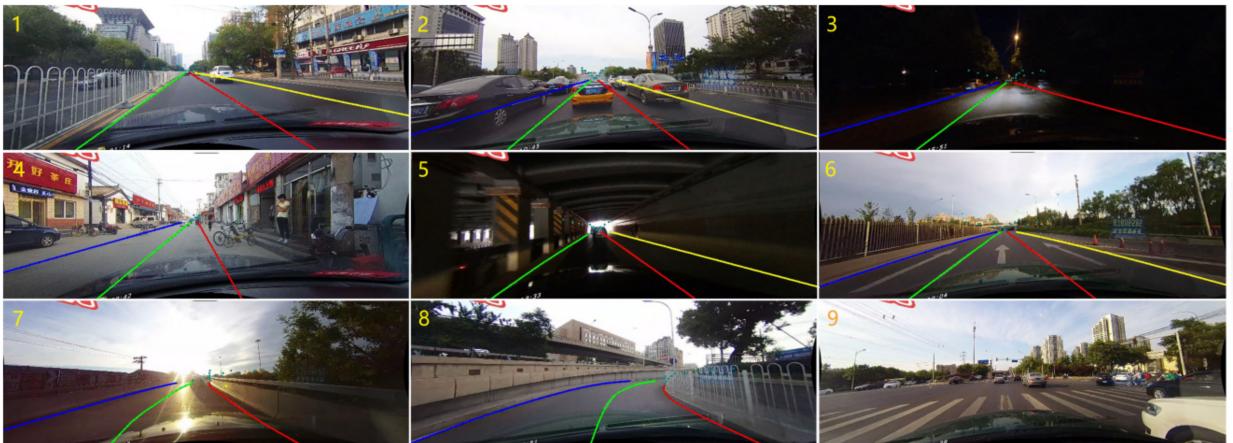


Figura 4: Ejemplos del dataset CULane [Pan et al., 2018].

2.2. Reconocimiento automático de matrículas (ALPR)

Un sistema de reconocimiento automático de matrículas o *Automatic License Plate Recognition* (ALPR) en inglés, suele estar compuesto de dos etapas principales: detección de coche-matrícula y reconocimiento de matrícula.

Detección de coche-matrícula. La etapa de detección coche-matrícula tiene como objetivo obtener las regiones que contienen matrículas en la imagen, para ello se suelen utilizar métodos de detección de objetos. En la literatura se diferencian dos tipos de métodos: *two-stage* y *one-stage*. Los métodos *two-stage* se basan en una primera fase de propuesta de regiones y, después, una segunda fase de clasificación de las regiones. El modelo más popular y con mejores resultados es Faster R-CNN [Ren et al., 2015], que usa una CNN y un método basado en *anchors* para proponer regiones. Los métodos *one-stage* son más rápidos porque no tienen fase de propuesta de regiones y, actualmente, también son más precisos. SSD [Liu et al., 2016], FCOS [Tian et al., 2019], RetinaNet

[Lin et al., 2017] y la familia YOLO [Redmon et al., 2016] son los modelos más populares. En los trabajos [Laroca et al., 2018, Laroca et al., 2021] primero se detecta el vehículo usando un detector de objetos y después la matrícula usando otro, lo que incrementa el coste computacional. Se hacía de esta manera porque los modelos tenían problemas para detectar objetos pequeños. Actualmente esto ya no es un problema, modelos como YOLOv5 [Jocher et al., 2021] detectan objetos pequeños.

Reconocimiento de matrícula. La etapa de reconocimiento de matrícula consiste en leer los caracteres alfanuméricos de las regiones obtenidas en la etapa anterior. Los métodos tradicionales [Du et al., 2013] primero realizaban una segmentación de los caracteres usando, por ejemplo, umbralización [Miyamoto et al., 1991] o información de los bordes [Capar and Gokmen, 2006], y posteriormente un reconocimiento de los caracteres usando, por ejemplo, filtros de Gabor [Hu et al., 2002]. Los métodos modernos basados en aprendizaje profundo pueden hacer la segmentación y reconocimiento de una sola vez. En [Menotti et al., 2014] usan una CNN random para extraer características de los caracteres. En [Wang et al., 2019] se propone un método basado en *Convolutional Recurrent Neural Network* (CRNN) y *Connectionist Temporal Classification* (CTC) [Shi et al., 2016] que reconoce caracteres a partir de una imagen recortada de matrícula. En [Montazzolli and Jung, 2017] se propone CR-NET como una CNN para segmentar y reconocer, que durante varios años ha sido uno de los métodos de referencia en el estado del arte. Trabajos como [Laroca et al., 2018] o [Laroca et al., 2021] hacen uso de CR-NET.

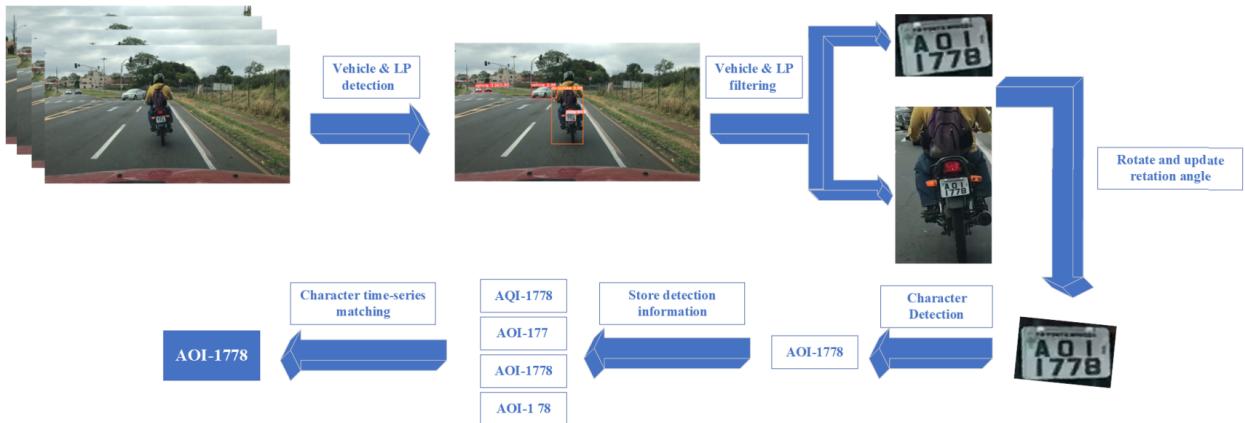


Figura 5: Sistema ALPR de [Quang et al., 2022].

En entornos reales, procesar un único *frame* suele provocar resultados erróneos. En [Quang et al., 2022] se hace frente a este problema usando un algoritmo que combina información de los caracteres en secuencias de *frames*. Actualmente se puede considerar el método más preciso del estado del arte. La Figura 5 muestra un esquema general del sistema. Usan YOLOv5 para detectar el vehículo y la matrícula al mismo tiempo. Todas las matrículas detectadas se almacenan y ordenan por tiempo en una cola, y después se usa un algoritmo para corregir la posible rotación que hayan sufrido. Para el reconocimiento de caracteres presentan un modelo basado en YOLOv5 reduciendo el número de filtros de cada capa para ganar velocidad y eliminando el cabezal de detección para características pequeñas y medianas, ya que los caracteres son grandes dentro de la región de la matrícula. Por último, usan su algoritmo *Character Time-series Matching* (CTM) que utiliza la información de una matrícula en varios *frames* consecutivos para realizar una predicción final.

En la Tabla 2 se comparan los métodos que han sido evaluados en el dataset UFPR-ALPR [Laroca et al., 2018]. Actualmente es el dataset más complicado del estado del arte, compuesto por secuencias de vídeo en escenas realistas (Figura 6).

Method	Accuracy
[Laroca et al., 2018]	64.89
[Laroca et al., 2021]	90
[Quang et al., 2022]	96.7

Cuadro 2: Evaluación de los métodos de ALPR en el dataset UFPR-ALPR [Laroca et al., 2018] usando la métrica *accuracy*.

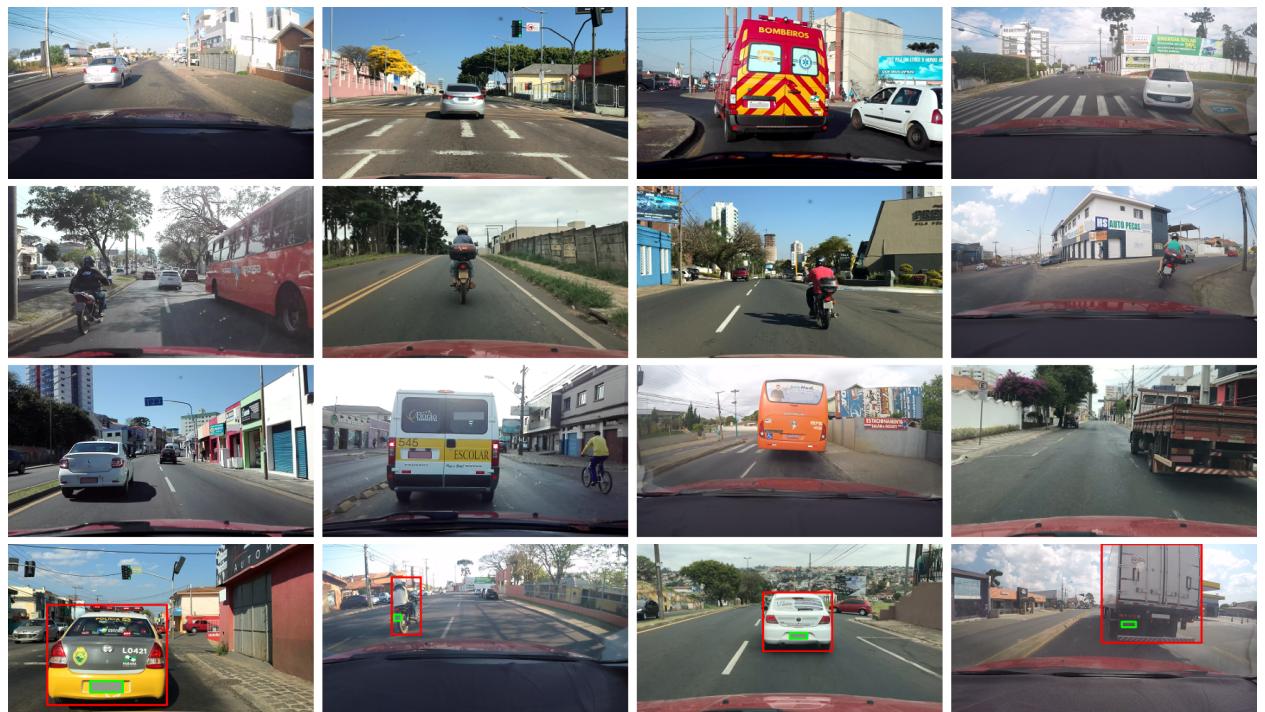


Figura 6: Ejemplos del dataset UFPR-ALPR [Laroca et al., 2018]. La última fila son ejemplos de anotaciones.

Referencias

- [Capar and Gokmen, 2006] Capar, A. and Gokmen, M. (2006). Concurrent segmentation and recognition with shape-driven fast marching methods. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 155–158. IEEE.
- [Chiu and Lin, 2005] Chiu, K.-Y. and Lin, S.-F. (2005). Lane detection using color-based segmentation. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005*. IEEE.
- [Danescu and Nedevschi, 2009] Danescu, R. and Nedevschi, S. (2009). Probabilistic lane tracking in difficult road scenarios using stereovision. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):272–282.
- [Du et al., 2013] Du, S., Ibrahim, M., Shehata, M., and Badawy, W. (2013). Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):311–325.
- [Fernández-López et al., 2013] Fernández-López, D., Montemayor, A. S., Pantrigo, J. J., Delgado, M. L., and Cabido, R. (2013). *Onboard Vision System for Bus Lane Monitoring*, page 286–295. Springer Berlin Heidelberg.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Hu et al., 2002] Hu, P., Zhao, Y., Yang, Z., and Wang, J. (2002). Recognition of gray character using gabor filters. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat. No. 02EX5997)*, volume 1, pages 419–424. IEEE.
- [Jocher et al., 2021] Jocher, G., Stoken, A., Ayush Chaurasia, Borovec, J., NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Changyu, L., Jiacong Fang, Abhiram V, Laughing, Tkianai, YxNONG, Skalski, P., Hogan, A., Jebastin Nadar, Imyhxy, Mammana, L., AlexWang1900, Fati, C., Montes, D., Hajek, J., Diaconu, L., Minh, M. T., , Marc, Albinxavi, , Fatih, , Oleg, and Wanghaoyang0106 (2021). ultralytics/yolov5: v6.0 - yolov5n 'nano' models, roboflow integration, tensorflow export, opencv dnn support.
- [Kim and Lee, 2014] Kim, J. and Lee, M. (2014). *Robust Lane Detection Based On Convolutional Neural Network and Random Sample Consensus*, page 454–461. Springer International Publishing.
- [Laroca et al., 2018] Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., and Menotti, D. (2018). A robust real-time automatic license plate recognition based on the yolo detector. In *2018 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE.
- [Laroca et al., 2021] Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., and Menotti, D. (2021). An efficient and layout-independent automatic license plate recognition system based on the yolo detector. *IET Intelligent Transport Systems*, 15(4):483–503.
- [Lin et al., 2017] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

- [Liu et al., 2010] Liu, G., Worgotter, F., and Markelic, I. (2010). Combining statistical hough transform and particle filter for robust lane detection and tracking. In *2010 IEEE Intelligent Vehicles Symposium*. IEEE.
- [Liu et al., 2016] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14, pages 21–37. Springer.
- [López et al., 2010] López, A., Serrat, J., Cañero, C., Lumbreiras, F., and Graf, T. (2010). Robust lane markings detection and road geometry computation. *International Journal of Automotive Technology*, 11(3):395–407.
- [Menotti et al., 2014] Menotti, D., Chiachia, G., Falcao, A. X., and Oliveira Neto, V. J. (2014). Vehicle license plate recognition with random convolutional networks. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE.
- [Miyamoto et al., 1991] Miyamoto, K., Nagano, K., Tamagawa, M., Fujita, I., and Yamamoto, M. (1991). Vehicle license-plate recognition by image analysis. In *Proceedings IECON'91: 1991 International Conference on Industrial Electronics, Control and Instrumentation*, pages 1734–1738. IEEE.
- [Montazzolli and Jung, 2017] Montazzolli, S. and Jung, C. (2017). Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 55–62. IEEE.
- [Neven et al., 2018] Neven, D., Brabandere, B. D., Georgoulis, S., Proesmans, M., and Gool, L. V. (2018). Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE.
- [Pan et al., 2018] Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X. (2018). Spatial as deep: Spatial cnn for traffic scene understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- [Qin et al., 2020] Qin, Z., Wang, H., and Li, X. (2020). *Ultra Fast Structure-Aware Deep Lane Detection*, page 276–291. Springer International Publishing.
- [Quang et al., 2022] Quang, H. C., Thanh, T. D., and Van, C. T. (2022). Character time-series matching for robust license plate recognition. In *2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*. IEEE.
- [Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- [Shi et al., 2016] Shi, B., Bai, X., and Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304.

- [Tabelini et al., 2021] Tabelini, L., Berriel, R., Paixao, T. M., Badue, C., De Souza, A. F., and Oliveira-Santos, T. (2021). Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 294–302.
- [Tian et al., 2019] Tian, Z., Shen, C., Chen, H., and He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636.
- [TuSimple,] TuSimple. Tusimple benchmark. Accesed September, 2020.
- [Van Gansbeke et al., 2019] Van Gansbeke, W., De Brabandere, B., Neven, D., Proesmans, M., and Van Gool, L. (2019). End-to-end lane detection through differentiable least-squares fitting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.
- [Wang et al., 2019] Wang, W., Yang, J., Chen, M., and Wang, P. (2019). A light cnn for end-to-end car license plates detection and recognition. *IEEE Access*, 7:173875–173883.
- [Zheng et al., 2022] Zheng, T., Huang, Y., Liu, Y., Tang, W., Yang, Z., Cai, D., and He, X. (2022). Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 898–907.