



Computational Intelligence in Electrical Engineering
Vol. 15, No. 1, 2024
pp. 57-72
Research Paper

Fast and flexible detection of vehicles in a sequence of images by deep networks

Nafiseh Zarei¹, Payman Moallem^{1*}, Mohammad Reza Shams², Rasoul Asgarian Dehcordi³

Dept. of Electrical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran

Dept. of Computer Engineering, Shahreza Campus, University of Isfahan, Shahreza, Iran

Power Department, Faculty of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran

Abstract:

Surveillance cameras can provide more accurate traffic information compared to GPS and infrared radiation sensors. With the intelligent processing of image information provided by them, the analysis of traffic problems is done more precisely. Meanwhile, the speed of car positioning is of particular importance because, after positioning the car, other processes are needed to make decisions that require time management. The purpose of the present study is to propose an algorithm based on deep networks that solves this challenge. In the presented algorithm, a fast and accurate detector network is designed based on multiple receptive fields, segmentation, and differential images, which increases the accuracy of the detector by producing semantic feature maps and filtering them. Also, it increases detector speed by reducing the parameters. Each vehicle's maneuver is decided based on the time and location information of that vehicle and the vehicles nearby. Then, according to the type of maneuver, the predictive network is selected in one of the modes of lane keeping, left turn or right turn. The networks designed in the proposed algorithm are complementary to each other. The performance of the proposed algorithm is demonstrated by experiments on the Highway and UA-DETRAC datasets.

Keywords: vehicle detection, position prediction, trajectory classifier.



This is an open access article under the CC BY-NC-ND/4.0/ License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).



<https://doi.org/10.22108/ISEE.2023.136806.1615>

مقاله پژوهشی

آشکارسازی سریع و منعطف وسایل نقلیه در رشته‌ای از تصاویر توسط شبکه‌های عمیق

نفیسه زارعی^۱، پیمان معلم^{۲*}، محمدرضا شمس^۳، رسول عسگریان دهکردی^۴

۱- دانشجوی دکتری گروه مهندسی برق، دانشکده فنی و مهندسی، دانشگاه اصفهان، اصفهان، ایران

nafisehzarei@eng.ui.ac.ir

۲- استاد گروه مهندسی برق، دانشکده فنی و مهندسی، دانشگاه اصفهان، اصفهان، ایران

p_moallem@eng.ui.ac.ir

۳- استادیار گروه مهندسی کامپیوتر، مرکز آموزش عالی شهرضا، دانشگاه اصفهان، شهرضا، ایران

m.r.shams@shr.ui.ac.ir

۴- دکتری برق، دانشکده مهندسی برق، دانشگاه صنعتی شاهرود، شاهرود، ایران

r_asgarian_dehkordi@yahoo.com

چکیده: دوربین‌های مداربسته در مقایسه با GPS و حسگرهای تشعشع مادون قرمز، قادر به ارائه اطلاعات دقیق‌تر ترافیکی هستند. با پردازش هوشمند اطلاعات تصویری فراهم‌شده توسط آنها تحلیل مسائل ترافیکی دقیق‌تر انجام می‌شود. در این میان، سرعت مکان‌یابی خودرو اهمیت ویژه‌ای دارد؛ زیرا پس از موقعیت‌یابی خودرو نیاز به پردازش‌های دیگری برای تصمیم‌گیری است که نیاز به مدیریت زمان دارد. هدف از مطالعه حاضر، ارائه الگوریتمی مبتنی بر شبکه‌های عمیق است که این چالش را برطرف سازد. در الگوریتم ارائه‌شده، یک شبکه آشکارساز سریع و دقیق بر پایه میدان‌های ادراکی چندگانه، قطعه‌بندی و تصاویر تفاضلی طراحی شده است که با تولید نقشه‌های ویژگی معنادار غربال‌شده، دقت آشکارساز و با کاهش پارامترها، سرعت آن را افزایش می‌دهد. در الگوریتم ارائه‌شده، نوع مانور هر خودرو با توجه به اطلاعات زمانی و مکانی آن خودرو و خودروهای اطرافش با استفاده از گریدنندی جاده تعیین می‌شود. سپس با توجه به نوع مانور، شبکه پیش‌بینی‌کننده را در یکی از حالات حفظ خط، گردش به چپ و گردش به راست خودرو انتخاب می‌کند. شبکه‌های طراحی‌شده در الگوریتم پیشنهادی مکمل یکدیگرند. عملکرد الگوریتم پیشنهادی با آزمایش بر روی مجموعه داده‌های Highway و UA-DETRAC نشان داده می‌شود.

واژه‌های کلیدی: آشکارسازی وسایل نقلیه، پیش‌بینی موقعیت، طبقه‌بندی مسیر.

۱- مقدمه

ترافیک [۶]، نظارت بر محیط شهری و اتومبیل‌های خودران [۷] نقش حیاتی ایفا می‌کند. الگوریتم‌های زیادی براساس بینایی کامپیوتری در این زمینه ارائه شده است [۸-۱۰]. هدف این الگوریتم‌ها کمک به سیستم مانیتورینگ خودکار است تا تجزیه و تحلیل رفتار خودرو در جاده یا در محیط شهری به‌طور مؤثرتری انجام شود؛ البته با توجه به محدودیت‌های قدرت محاسباتی و فضای ذخیره‌سازی در این سیستم، آشکارسازهای طراحی‌شده برای این الگوریتم‌ها باید بتوانند با حجم سخت‌افزاری کمتر، سرعت و دقت مناسب و مطلوبی داشته باشند [۱۱ و ۱۲]؛ بنابراین، طراحی آشکارسازهای سبک و قدرتمند در این زمینه بسیار جذاب

تشخیص وسایل نقلیه در بسیاری از کاربردها مانند مسائل امنیتی یا مسائل مربوط به حمل‌ونقل کالا [۱ و ۲]، تصاویر هوایی [۳ و ۴]، مدل‌سازی ترافیک [۵]، نظارت بر

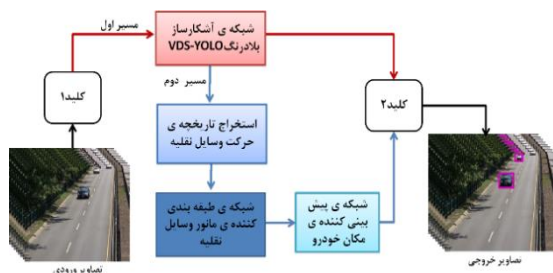
^۱ تاریخ ارسال مقاله: ۱۴۰۱/۱۱/۲۴

تاریخ پذیرش مقاله: ۱۴۰۲/۰۷/۰۸

نام نویسنده مسئول: پیمان معلم

نشانی نویسنده مسئول: ایران، اصفهان، دانشگاه اصفهان، دانشکده فنی و مهندسی، گروه مهندسی برق

بقیه فریم‌های آن براساس موقعیت هر خودرو در فریم‌های قبل، پیش‌بینی موقعیت خودرو انجام شده است؛ البته در چند ثانیه اول پردازش، تنها از آشکارساز استفاده می‌شود تا اطلاعات لازم و کافی برای شبکه پیش‌بینی‌کننده فراهم شود و دقت پیش‌بینی افزایش یابد. استفاده از این دو شبکه در یک الگوریتم، عملکرد یکدیگر را تکمیل می‌کند و باعث افزایش میانگین سرعت اجرای الگوریتم می‌شود؛ زیرا پیش‌بینی بسیار سریع‌تر از تشخیص است. با توجه به پیچیدگی‌های صحنه و نظر کاربر، زمان استفاده از شبکه‌های آشکارساز و پیش‌بینی در تناوب تعریف شده است و حتی طول دوره تناوب را می‌توان تغییر داد. این کار انعطاف‌پذیری الگوریتم از لحاظ تنظیم دقت و سرعت موقعیت‌یابی خودروهای حاضر در صحنه را افزایش می‌دهد.



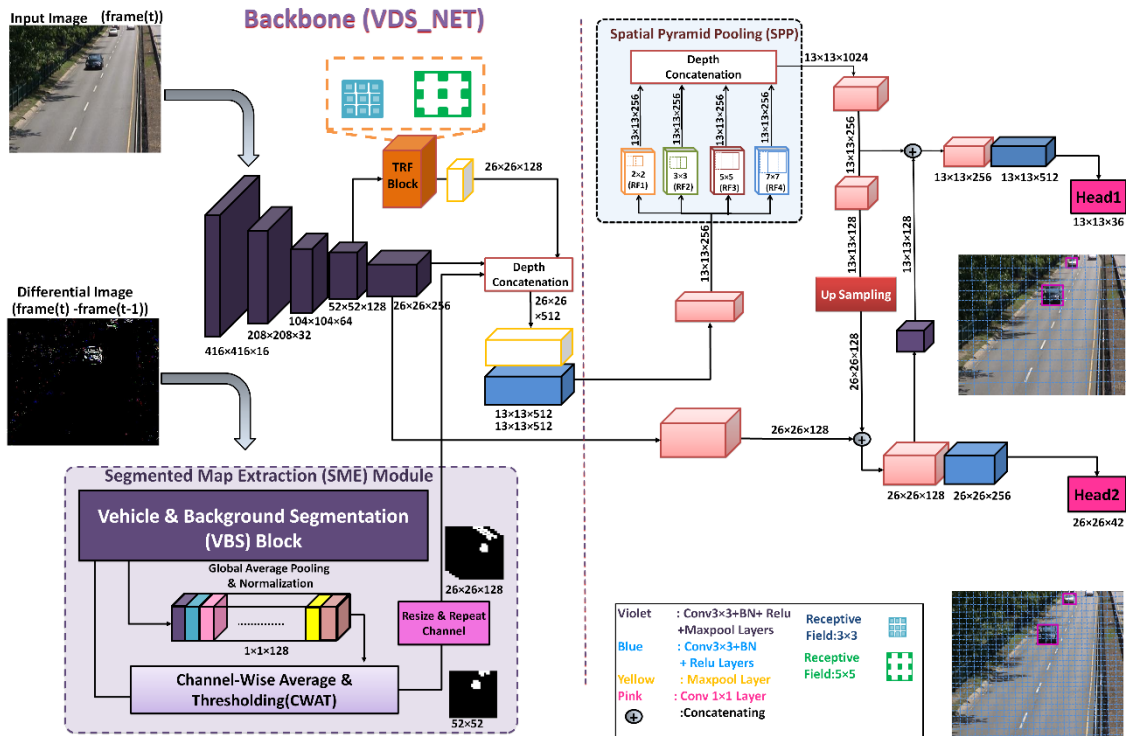
شکل (۱): الگوریتم طراحی شده در این مطالعه به نام AVD

الگوریتم AVD با دوره تناوب برابر با دو فریم اجرا می‌شود. تاریخچه زمانی با استفاده از اطلاعات به دست آمده از فریم‌های قبل به دست می‌آید و باعث افزایش دقت شبکه پیش‌بینی‌کننده می‌شود. علاوه بر کوتاه در نظر گرفتن زمان پیش‌بینی برای رسیدن به دقت بالاتر، مسیر خودرو نیز با یک شبکه طبقه‌بندی‌کننده تعیین می‌شود و سپس با توجه به مسیر تعیین شده، شبکه پیش‌بینی‌کننده که برای آن مسیر آموزش دیده است، برای پیش‌بینی موقعیت خودرو تعیین می‌شود. در ادامه، الگوریتم پیشنهادی به تفصیل توضیح داده می‌شود.

بوده و یک مشکل چالش‌برانگیز است که علایق و تلاش‌های دانشگاهی و صنعتی را به خود جلب می‌کند [۱۳]. توسعه سریع سخت‌افزار نیز به باعث شده است این تلاش‌ها هرچه بیشتر به سرانجام برسد. همچنین، رشد تصاعدی شبکه‌های مبتنی بر یادگیری عمیق در حوزه طبقه‌بندی، تشخیص و قطعه‌بندی معنادار اشیا در سالیان اخیر انقلاب بزرگی را به وجود آورده است [۱۱-۱۹] که نشان‌دهنده برتری آنها نسبت به روش‌های سنتی است. دلایل این پیشرفت‌ها علاوه بر ماهیت شبکه‌های عمیق و سخت‌افزارهای رو به توسعه، در افزایش مداوم داده‌ها نیز هست. این شبکه‌ها در حوزه آشکارسازی به دو روش مختلف ساخته می‌شوند. روش نخست که به روش دومرحله‌ای نیز معروف است، هزینه زمانی زیادی را شامل می‌شود؛ در حالی که روش دوم (یک مرحله‌ای) سریع‌تر است. آشکارسازهای پایه YOLO شبکه‌های یک مرحله‌ای هستند. اگرچه عملکرد شبکه YOLO با انتشار نسخه‌های جدیدتر به تدریج بهبود یافته است، در بیشتر آنها تعداد پارامترها و BFlops زیاد شده است؛ در نتیجه، سرعت آنها کاهش یافته است و از لحاظ سخت‌افزاری نیز مقرون به صرفه نیستند. در این مطالعه، الگوریتمی انعطاف‌پذیر از لحاظ سرعت و دقت به نام AVD^۱ ارائه شده است که در ضمن افزایش دقت، سرعت موقعیت‌یابی را به نحو چشمگیری افزایش می‌دهد و با مسئله انسداد خودرو نیز از طریق استفاده از شبکه‌های پیش‌بینی‌کننده مقابله می‌کند.

۲- روش پیشنهادی

الگوریتم پیشنهادی در این مطالعه به نام AVD - نشان داده شده در شکل (۱) - سرعت تشخیص وسایل نقلیه را افزایش می‌دهد. در الگوریتم ارائه شده، به‌طور متناوب از شبکه‌های آشکارساز (مسیر اول در شکل (۱)) و پیش‌بینی‌کننده آشکارساز (مسیر دوم در شکل (۱)) در یک دوره تناوب مشخص استفاده می‌شود. در واقع، در تعداد مشخصی از فریم‌های این دوره تناوب از آشکارساز و در



شکل (۲): آشکارساز VDS-YOLO طراحی شده در این مطالعه

۲-۱- شبکه آشکارساز طراحی شده

این مطالعه، یک شبکه آشکارسازی مبتنی بر YOLO به نام VDS-YOLO^۱ را پیشنهاد می‌کند. این شبکه در شکل (۲) نشان داده شده است. ستون فقرات^۲، گردن^۳ و سر^۴ سه جزء اصلی آشکارسازهای مبتنی بر YOLO هستند. با توجه به وضوح بالاتر و ویژگی‌های مکانی دقیق‌تر، نقشه‌های ویژگی استخراج شده از ستون فقرات در مکان‌یابی وسایل نقلیه، مؤثرتر از نقشه‌های ویژگی سایر لایه‌ها در شبکه آشکارساز عمل می‌کنند. قسمت‌های سر و گردن نیز برای تعیین کلاس خودرو مفیدترند؛ زیرا با وجود داشتن لایه‌هایی با جزئیات مکانی کمتر (به دلیل وضوح کمتر)، اطلاعات معنایی و عمق بیشتری را شامل می‌شوند. آشکارساز پیشنهاد شده در این مطالعه، علاوه بر افزایش دقت تشخیص موقعیت خودرو، مقاومت آن را نیز در برابر تغییرات مقیاس خودرو افزایش می‌دهد. این کار با مازول^۵ و بلوک TRF^۶ در ستون فقرات انجام می‌شود.

بلوک TRF در شکل (۳)، به منظور تهیه نقشه‌های ویژگی بهتر با میدان‌های ادراکی متفاوت در رزولوشن

26×26 طراحی شده است. در این بلوک از اتصالات باقیمانده^۸ استفاده شده است. این اتصالات با حفظ خروجی لایه‌های قبل، کیفیت آموزش شبکه را بهبود می‌بخشند. علاوه بر آن، بلوک TRF پیشنهادی شامل دو لایه^۹ پیچشی 3×3 متوالی است که یک میدان ادراکی 5×5 را تولید می‌کند. یک لایه^{۱۰} پیچشی 1×1 را نیز شامل می‌شود. نقشه‌های ویژگی استخراج شده در بلوک TRF به دو میدان پذیرنده 3×3 و 5×5 متصل و به لایه‌های کانولوشن بعدی منتقل می‌شوند. در واقع با طراحی و استفاده از بلوک TRF به جای بلوک‌های قبلی، ویژگی‌هایی با میدان گیرنده 3×3 ، 5×5 به لایه بعدی منتقل می‌شوند و جزئیات مکانی بیشتری را به دلیل در نظر گرفتن میدان ادراکی کوچک‌تر به لایه بعدی انتقال می‌دهند. میدان‌های ادراکی متفاوت در بلوک TRF، شبکه را در برابر تغییرات مقیاس قوی می‌سازد.



(الف)



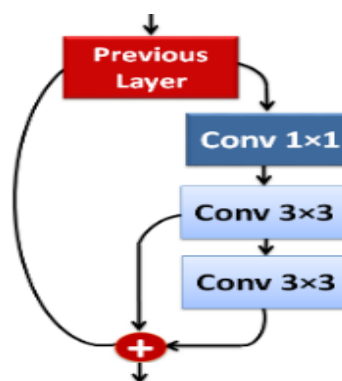
(ب)



(ج)

شکل (۴): نمونه‌ای از بخش‌بندی تصویر (الف) تصویر ورودی و (ب) تصویر تفاضلی از طریق مدل‌کردن پس‌زمینه به صورت پویا. (ج) تصویر قطعه‌بندی شده با استفاده از شبکه عمیق.

یکی دیگر از اشکالات اساسی روش‌های مرسوم مدل‌کردن پس‌زمینه، این است که تنها وسایل نقلیه در حال حرکت را می‌تواند از پس‌زمینه جدا کند و در جداسازی وسایل نقلیه ثابت ناتوان است؛ زیرا این روش، مبتنی بر حرکت است و ذاتاً از یافتن اشیای ثابت ناتوان است. همان‌طور که در شکل (۴-ج) مشاهده می‌شود روش شبکه مبنا به‌طور مؤثر وسایل نقلیه در مقیاس‌های مختلف را از پس‌زمینه جدا می‌کند. در این مطالعه، قطعه‌بندی با استفاده از تصاویر تفاضلی انجام می‌شود که به‌عنوان ورودی به یک شبکه عمیق مبتنی بر U-Net^{۱۱} به نام LSDN^{۱۱} داده می‌شوند. این شبکه در شکل (۵) دیده می‌شود و شامل یک رمزگذار و رمزگشا است. به‌منظور رسیدن به سرعت بالاتر، تنها رمزگذار آن در آشکارساز پیشنهادی به‌عنوان بلوک VBS^{۱۲} استفاده می‌شود. شبکه LSDN، با استفاده از همان مجموعه



شکل (۳): بلوک TRF پیشنهادی در این مطالعه

علاوه بر طراحی بلوک TRF، از ماژول SME نیز به‌منظور افزایش دقت شبکه آشکارساز استفاده می‌شود. این ماژول قبل از آموزش آشکارساز آموزش داده و در طول آموزش آشکارساز به‌اصطلاح منجمد^{۱۳} می‌شود؛ یعنی نرخ یادگیری برای لایه‌های این ماژول در طول آموزش آشکارساز صفر در نظر گرفته می‌شود. در واقع آشکارساز با روش یادگیری انتقالی آموزش داده می‌شود. ماژول SME که در قسمت پایین شکل (۲) نشان داده شده است، به‌صورت موازی با لایه‌های استخراج ویژگی در ابتدای آشکارساز اجرا می‌شود و با پردازش تصاویر تفاضلی، ویژگی‌های مهم‌تری را برای وسیله نقلیه فراهم می‌کند که در آنها توجه بیشتری به موقعیت وسایل نقلیه می‌شود. این نقشه‌های ویژگی به دو بخش تقسیم می‌شوند: پیش‌زمینه و پس‌زمینه. در علم بینایی ماشین پیش‌زمینه با استفاده از روش‌های مختلفی از پس‌زمینه جدا می‌شود. روش مخلوط گاوسی^{۱۴} و محاسبه پس‌زمینه پویا از آن جمله‌اند. در این روش‌ها حرکت دوربین یا اجسام دیگر غیر از پس‌زمینه باعث ایجاد خطا می‌شود. این در حالی است که روش‌های مبتنی بر شبکه، خطای بسیار کمتری را ایجاد می‌کنند. شکل (۴) تصویری را نشان می‌دهد که با استفاده از دو روش مختلف مدل‌سازی پس‌زمینه پویا و روش مبتنی بر شبکه به کلاس‌های خودرو و پس‌زمینه تقسیم شده است. روش اول، دریافتن پیش‌زمینه (وسیله نقلیه) هنگام حرکت برگ درختان دچار خطا می‌شود. علاوه بر این، همان‌طور که دیده می‌شود در این روش، وسایل نقلیه جدا شده از پس‌زمینه، انسجام کافی ندارند و بریده‌بریده هستند.

آشکارسازهای سریع است. داده‌های اضافی و تکراری در تصاویر متوالی ورودی نیز در تصاویر تفاضلی حذف می‌شوند؛ در نتیجه، تأکید بیشتری بر اطلاعات مفیدی است که از نظر تعداد بسیار کمتر از داده‌های تصویر ورودی است. استفاده از اطلاعات مفید با حجم کم به جای کل تصویر ورودی که حجم زیادی دارد، نیاز به افزایش عمق شبکه به منظور رسیدن به دقت بالاتر را از بین می‌برد و سرعت شبکه آشکارساز را از این طریق می‌توان افزایش داد. با وجود اینکه در تعداد زیادی از نقشه‌های ویژگی خروجی بلوک VBS در آشکارساز VDS_YOLO، وسایل نقلیه و پس‌زمینه به خوبی از هم تفکیک شده‌اند، تعداد کمی از آنها بسیار تاریک هستند و اطلاعات مهمی ندارند؛ بنابراین، ابتدا با استفاده از لایه GAP^{۱۳} اهمیت هر یک از ۱۲۸ نقشه ویژگی به صورت

$$F^l = [f^{(1)}, f^{(2)}, \dots, f^{(cl)}] \in R^{(cl \times H \times W)}$$

به دست می‌آید. c_l برابر با ۱۲۸ و H و W معرف طول و عرض نقشه‌های ویژگی $f^{(i)}$ هستند. میانگین هر کدام از نقشه‌های ویژگی مطابق با رابطه (۱) محاسبه می‌شود.

$$w^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W f^{(k)}(i, j) \quad (1)$$

این مقادیر با تابع SoftMax در رابطه (۲) نرمالیزه می‌شوند.

$$S(w_k) = \frac{\exp(w_k)}{\sum_{k=1}^{cl} \exp(w_k)} \quad (2)$$

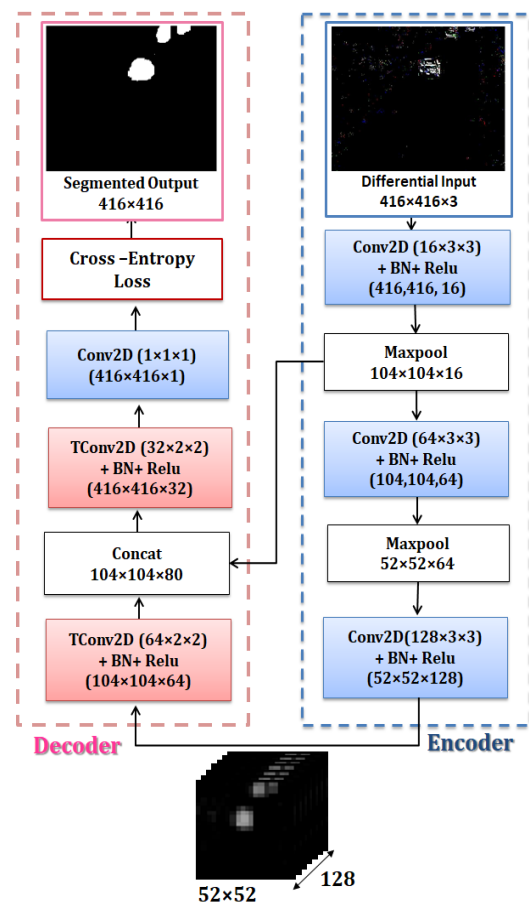
و در بلوک CWAT^{۱۴}، با استفاده از آنها میانگین وزن‌دار نقشه‌های ویژگی براساس مقادیر نرمال شده در رابطه (۳) محاسبه می‌شود.

$$f^{(CWAP)} = \sum_{k=1}^{cl} (S(w_k) \times f^{(k)}) \quad (3)$$

سپس میانگین وزن‌دار به دست آمده تکرار می‌شود تا از نظر تعداد با نقشه‌های ویژگی دیگری که قرار است با آنها دنبال هم قرار گیرند^{۱۵}، در تعادل باشد. بلوک GAP و CWAT از انتقال کانالی که حاوی نقشه ویژگی کم‌ارزشی است، به لایه بعدی جلوگیری می‌کند.

تصاویری که برای آموزش آشکارساز استفاده می‌شود، با Ground-Truth متفاوت آموزش داده می‌شود. ورودی شبکه LSDN، تصویر تفاضلی است که تنها اجسام متحرک در آن مشخص می‌شوند؛ زیرا اگر جسم، ثابت و بدون حرکت باشد، در تصویر خروجی تفاضل یک فریم و فریم ماقبلش مقدار صفر را خواهد داشت.

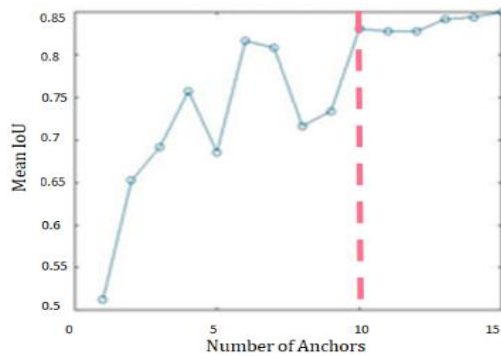
بنابراین، آشکارساز پیشنهادی در اجسام متحرک با دقت بالایی قادر به تشخیص است و درباره اجسام ثابت، مانند شبکه YOLO پایه عمل می‌کند.



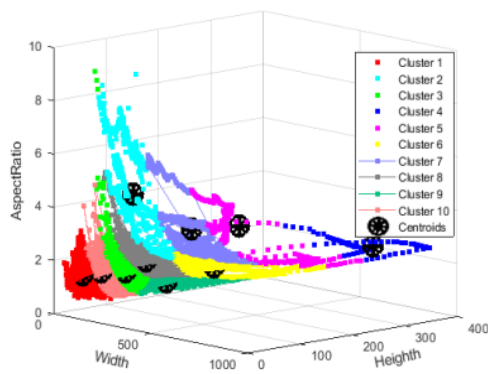
شکل (۵): شبکه بخش‌بندی کننده LSDN که رمزگذار آن، بلوک VBS در شبکه VDS-YOLO پیشنهادی است.

بلوک VBS شامل سه لایه کانولوشن با کرنل 3×3 ، لایه نرمال‌سازی، یک تابع فعال‌سازی خطی اصلاح‌شده (Relu) و دولایه ادغام است. در این بلوک سعی شده است پارامتر قابل یادگیری لایه‌ها تا حد امکان کاهش یابد؛ زیرا هدف این مطالعه، افزایش سرعت تشخیص با دقت قابل مقایسه با

مشکی‌رنگ نشان داده شده‌اند. در الگوریتم K-means، در ابتدا تعداد خوشه‌ها مشخص می‌شود. مراکز خوشه‌ها ابتدا به صورت دستی انتخاب می‌شوند و در طول الگوریتم تغییر می‌کنند؛ به طوری که تابع هدف مینیمم شود. داده‌ها براساس فاصله اقلیدسی‌شان تا مراکز خوشه‌ها به خوشه‌های مختلف تعلق می‌گیرند و در واقع خوشه‌بندی می‌شوند. این مراحل تکرار می‌شود تا مراکز خوشه‌ها تغییر چندانی نکنند.



(الف)



(ب)

شکل (۶): الف، نمودار IoU برای تعیین تعداد مناسب لنگرها با توجه به داده‌های آموزشی UA_DETTRAC ب، خوشه‌بندی داده‌های آموزشی UA_DETTRAC در فضای سه‌بعدی با الگوریتم K-means.

۲-۲- آماده‌سازی تاریخچه حرکت وسایل نقلیه

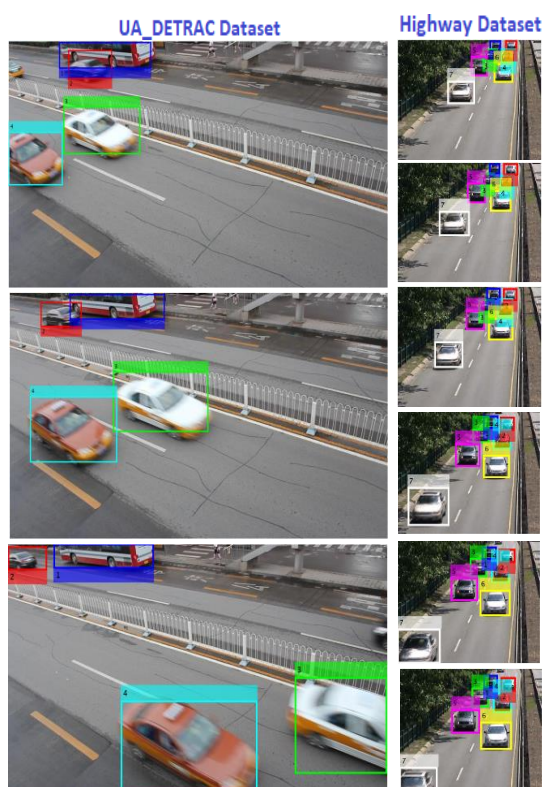
بلوک پیش‌بینی موقعیت از اطلاعات سری زمانی هر وسیله نقلیه در فریم‌های قبل استفاده می‌کند. این اطلاعات، تاریخچه حرکت وسایل نقلیه در رشته تصویری است که مسیر وسایل نقلیه را نشان می‌دهند.

$$X = [X^{(t-t_h)}, \dots, X^{(t-1)}, X^t] \quad (۴)$$

علاوه بر طراحی بلوک TRF و ماژول SME، انتخاب تعداد و اندازه مناسب لنگرها^{۱۶} نیز در افزایش دقت آشکارساز پیشنهادی ما مؤثر بوده است. معمولاً در مراحل ابتدایی شبکه‌های آشکارساز، انتخاب‌های زیادی برای هدف وجود دارد که در مراحل بعد غربال می‌شوند و با معیارهایی نظیر MeanIoU^{۱۷} کاهش می‌یابند. در این معیار که نسبت هم‌پوشانی هدف آشکارشده با GroundTruth را تعیین می‌کند، اگر نسبت هم‌پوشانی کمتر از حد آستانه در نظر گرفته شود، به طور مثال ۰.۴٪ باشد، هدف نادیده گرفته می‌شود. با تعریف لنگر، آشکارساز از ابتدا اهدافی که نسبت طول به عرض و اندازه آنها به لنگر تعریف‌شده نزدیک نیست را کاندید هدف در نظر نمی‌گیرد. این کار باعث می‌شود تعداد کاندیدهای هدف و در نتیجه، تعداد آشکارسازی‌های غلط کاهش یابد. در واقع، لنگرها یک جعبه مرزی از پیش تعریف شده با ارتفاع و عرض معلوم هستند که به افزایش دقت آشکارساز کمک می‌کنند. آنها براساس اندازه وسایل نقلیه در مجموعه داده آموزشی انتخاب می‌شوند.

در این مطالعه، همان‌طور که در شکل (۶-الف) نشان داده شده، تعداد لنگرها برای آموزش شبکه با توجه به معیار MeanIoU برای مجموعه داده آموزشی تعیین شده است. MeanIoU برای تعداد مختلف k لنگر، محاسبه شده است. سپس با توجه به اینکه در $k = 10$ بهترین میزان هم‌پوشانی بین هدف و لنگرها دیده شده، تعداد ۱۰ لنگر برای آموزش شبکه آشکارساز انتخاب شده است. در واقع، این تعداد لنگر بهترین نتیجه ارزیابی تشخیص را در پی دارند.

پس از تعیین تعداد لنگرها، اندازه لنگرها باید مشخص شود. معمولاً طول و عرض لنگرها با استفاده از الگوریتم‌های خوشه‌بندی مانند K-means مشخص می‌شود. در این مطالعه، برای رسیدن به دقت بالاتر علاوه بر طول و عرض نسبت عرض به طول وسایل نقلیه در داده‌های آموزشی نیز در تعیین لنگرها دخالت داده شده است. همان‌طور که در شکل (۶-ب) دیده می‌شود در فضای ۳ بعدی طول، عرض و نسبت طول به عرض، داده‌های آموزشی در ۱۰ گروه خوشه‌بندی شده‌اند. در این شکل، خوشه‌ها با رنگ‌های متفاوت و مراکز آنها با دایره‌ها



شکل (۷): نتایج دیداری برچسب‌گذاری پایدار

در حالی که t_i مبدأ زمانی تاریخچه حرکت خودرو است.

$$X^t = [x_0^{(t)}, y_0^{(t)}, \dots, x_n^{(t)}, y_n^{(t)}] \quad (5)$$

x و y مختصات وسیله نقلیه در فریم‌های قبل از فریم در حال پردازش هستند. با توجه به اینکه در هر تصویر چندین وسیله نقلیه وجود دارد، تا زمانی که وسیله نقلیه در میدان دید دوربین مداربسته قرار دارد، باید یک نام یا برچسب منحصر به فرد به آن اختصاص داده شود و سپس تاریخچه حرکت هر خودرو در یک زیرمجموعه اختصاصی مانند یک تانوسور شبیه به FIFO^{۱۸} برای همان وسیله نقلیه ذخیره شود. شکل (۷)، خروجی برچسب‌گذاری چندین فریم متوالی از مجموعه داده‌های این مطالعه را نشان می‌دهد. در این شکل برچسب‌های مختلف با رنگ‌های متفاوت نشان داده شده‌اند؛ برای مثال، در مجموعه داده UA_DETRAC برچسب ۱، با رنگ آبی، برچسب ۲، با رنگ قرمز و برچسب‌های ۳ و ۴ با رنگ‌های سبز و آبی فیروزه‌ای مشخص شده‌اند. همان‌طور که دیده می‌شود در طول زمان برچسب اتوبوس و سایر وسایل نقلیه تغییر نکرده است. وجود این برچسب‌ها سبب می‌شود تاریخچه مسیر حرکت خودروهای مختلف به درستی در تانوسور ثبت شود و مکان یک خودرو در فریم n ام در تاریخچه حرکت خودروی دیگر ثبت نشود. برای مجموعه داده Highway نیز همین توضیحات صادق‌اند. همان‌طور که دیده می‌شود برچسب هر خودرو در طول فریم‌های متوالی تغییر نکرده است. معیار فاصله شباهت هر وسیله نقلیه با مشخصات $[x^{(t+1)}, y^{(t+1)}, I^{(t+1)}, A^{(t+1)}]$ و معیار RMSE حاصل می‌شود که در آن، A معرف مساحت وسیله نقلیه، I میانگین شدت روشنایی مربوط به پیکسل‌های متعلق به وسیله نقلیه و x و y موقعیت افقی و عمودی وسیله نقلیه در فریم $t+1$ ام است. در واقع، فاصله شباهت بین دو وسیله نقلیه با مجذور مربعات فواصل بین موقعیت، شدت متوسط و مساحت خودروها در فریم جاری و فریم قبل آن (فریم t ام) تعیین می‌شود.

۳-۲- شبکه‌های طبقه‌بندی کننده مانور و

پیش‌بینی کننده موقعیت وسایل نقلیه

در الگوریتم AVD ارائه شده در این مطالعه، با استفاده متناوب از طبقه‌بند مسیر، شبکه‌های بازگشتی پیش‌بینی کننده و شبکه‌های آشکارساز مکان‌یابی وسایل نقلیه انجام می‌شود. دلیل افزایش سرعت الگوریتم، سرعت بیشتر پیش‌بینی نسبت به آشکارسازی است. همچنین، ابعاد خودرو علاوه بر موقعیت آن قابل پیش‌بینی است؛ اما به دلیل اینکه دوره پیش‌بینی در این تحقیق دو فریم است و تشخیص و پیش‌بینی به صورت یک فریم در میان انجام می‌شود، تغییر در ابعاد خودرو بسیار ناچیز و قابل صرف نظر است. ورودی شبکه‌های پیش‌بینی، تاریخچه موقعیت خودروهای حاضر در صحنه است و چون بسیار دقیق است، دقت پیش‌بینی را افزایش می‌دهد. در واقع، شبکه‌های آشکارساز و پیش‌بینی کننده در الگوریتم AVD مکمل یکدیگرند. طبقه‌بند مسیر نیز برای افزایش بیشتر شبکه پیش‌بینی کننده طراحی شده است. شبکه‌های بازگشتی مبتنی بر LSTM

$$c_t = \tanh(W_s X_t + U_s h_{t-1}) \quad (۹)$$

$$s_t = c_t \odot i_t + s_{t-1} \odot f_t \quad (۱۰)$$

$$o_t = \sigma(U_o h_{t-1} + W_o X_t) \quad (۱۱)$$

$$h_t = \tanh(s_t) \odot o_t \quad (۱۲)$$

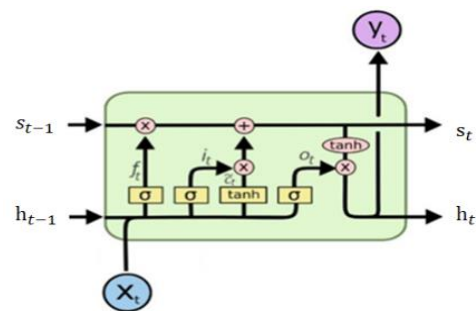
X_t ، اطلاعات ورودی در فریم فعلی است و تاریخچه حرکت وسیله نقلیه است که از بلوک آماده‌سازی اطلاعات سری زمانی به دست آمده است؛ در حالی که h_{t-1} خروجی شبکه در قاب قبلی و h_t خروجی شبکه در فریم جاری است. همچنین، "⊙" نشان‌دهنده نماد ضرب نقطه‌ای است. ماتریس‌های وزن $W_f, U_f, W_o, U_o, W_i, U_i$ به ترتیب دروازه‌های ورودی، خروجی و فراموشی را نشان می‌دهند؛ در حالی که سلول حالت s_t که نقش حافظه را در شبکه بازگشتی ایفا می‌کند، در هر فریم به‌روزرسانی می‌شود. در این مطالعه، از شبکه LSTM برای رگرسیون مسیر وسایل نقلیه در سه حالت حفظ خط، گردش به چپ و گردش به راست وسایل نقلیه در مجموعه داده‌های این مطالعه استفاده می‌شود. معماری الگوریتم پیشنهادی در شکل (۹) نشان داده شده است.

الگوریتم پیشنهادی شامل بخش‌هایی درباره آماده‌سازی اطلاعات زمانی، طبقه‌بندی مسیر و پیش‌بینی موقعیت خودرو است. بخش طبقه‌بندی مسیر به منظور بهبود عملکرد شبکه پیش‌بینی‌کننده موقعیت طراحی شده است. این شبکه طبقه‌بندی‌کننده است که با توجه به تاریخچه مسیر حرکت وسیله نقلیه و وسایل نقلیه اطراف آن تعیین می‌کند خودرو در یک خط مستقیم حرکت می‌کند یا تغییر مسیر می‌دهد. در این مطالعه، یک طبقه‌بندی‌کننده با سه کلاس مسیر مستقیم، تغییر جهت به چپ و تغییر جهت به راست پیشنهاد می‌شود و از شبکه‌های RLD-LSTM، KL-LSTM و LLD-LSTM نیز برای پیش‌بینی مسیر استفاده می‌شود. پیش‌بینی موقعیت هر وسیله نقلیه براساس کلاس مسیر مشخص شده توسط طبقه‌بندی‌کننده انجام می‌شود.

وابستگی‌های زمانی طولانی مدت را به حساب می‌آورند و بنابراین، از مدل‌های پیش‌بینی موقعیت سنتی مانند مدل شتاب ثابت (CA) دقیق‌ترند. در مدل شتاب ثابت، با توجه به رابطه (۷)، موقعیت خودرو در قاب فعلی (p_2) براساس موقعیت خودرو در قاب قبلی (p_1) و سرعت و شتاب خودرو حاصل می‌شود.

$$p_2 = \frac{1}{2} a \Delta t^2 + v \Delta t + p_1 \quad (۶)$$

که در آن شتاب (a) و سرعت (v) معلوم در نظر گرفته می‌شوند؛ با این حال، این فرض همیشه درست نیست و سرعت و شتاب خودروها با توجه به شرایط ترافیکی و تصمیم راننده ممکن است بارها تغییر کند. در شبکه‌های بازگشتی، تغییر مسیر توسط شبکه برای مقادیر متفاوت سرعت و شتاب در طول زمان و مسیرهای ممکن رانندگی آموخته شده است. از آنجایی که این شبکه‌ها چندین تابع فعال‌سازی غیرخطی سیگموئید و تانژانت شبکه هذلولی را شامل می‌شوند، الگوهای حرکتی پیچیده را پیش‌بینی می‌کنند (شکل (۸)).



شکل (۸): ساختار شبکه بازگشتی LSTM

این توابع تعیین می‌کنند کدام اطلاعات ذخیره شده از فریم‌های قبلی، حفظ و کدام‌یک از حافظه شبکه حذف شوند. همچنین، چه اطلاعاتی از فریم فعلی باید ذخیره و پردازش شوند. در این مطالعه، از شبکه بازگشتی LSTM استفاده شده است. معادلات مربوط به عملکرد دروازه‌های^{۱۹} این شبکه به صورت زیر بیان می‌شوند:

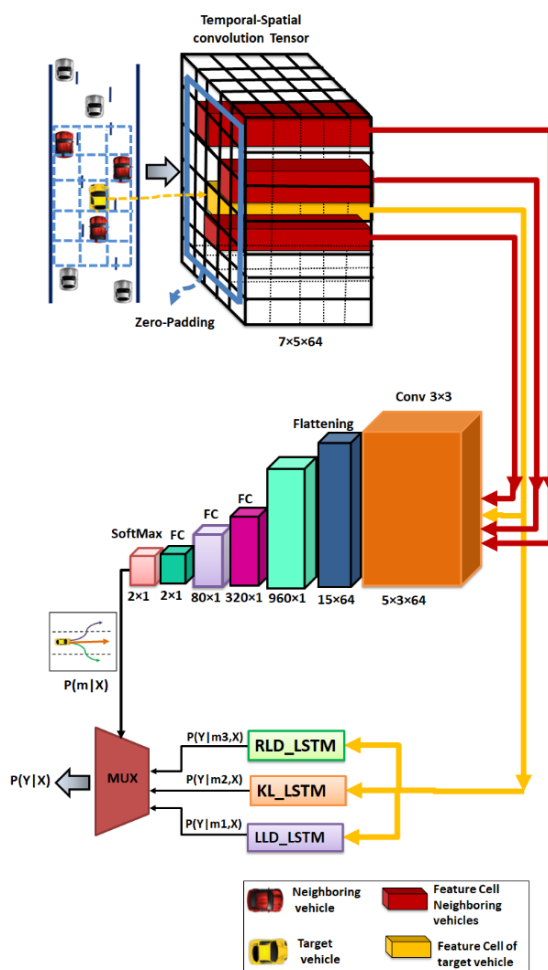
$$i_t = \sigma(W_i X_t + U_i h_{t-1}) \quad (۷)$$

$$f_t = \sigma(W_f X_t + U_f h_{t-1}) \quad (۸)$$

شبیه‌سازی MATLAB انجام شده‌اند و روی رایانه‌ای با ویندوز ده ۶۴ بیتی و پردازنده Core i7-3.60GHz با ۱۶ گیگابایت RAM و کارت گرافیک GTX 1060 اجرا شده‌اند.

۱-۴- مجموعه داده‌ها

از آنجایی که در این تحقیق، علاوه بر شبکه‌های آشکارساز و بخش‌بندی، از شبکه‌های پیش‌بینی‌کننده موقعیت نیز استفاده می‌شود و ورودی این شبکه‌ها نیز داده‌های سری زمانی هستند که مسیر حرکت خودرو را نشان می‌دهند، داده‌هایی برای آموزش و آزمایش الگوریتم پیشنهادی لازم است که شامل تصاویر متوالی باشد. مجموعه داده UA-DETRAC و مجموعه داده Highway از دسته پایه CDNet2014 دارای این خاصیت‌اند و در این مطالعه استفاده شده‌اند. مجموعه داده UA DETRAC شامل تعداد زیادی خودرو در کلاس‌ها و مقیاس‌های مختلف است و به‌جز وسایل نقلیه، شامل اجسام متحرکی مانند دوچرخه و عابران پیاده نیز هستند. برای آموزش شبکه‌های VDS_YOLO و LSDN، مجموعه داده‌ای مناسب است که Ground-Truth مناسب برای کارهای آشکارسازی خودرو و بخش‌بندی را شامل شود. مجموعه داده Highway برای بخش‌بندی مناسب است. این مجموعه داده، مجموعه‌ای از تصاویر است که به‌طور پیاپی با دوربین‌های نظارت بزرگراه گرفته شده و برای آموزش شبکه‌های LSDN در این مطالعه استفاده شده است؛ اما مجموعه داده Highway فاقد اطلاعات کافی برای آشکارسازی است. مجموعه داده UA-DETRAC نیز حاوی Ground-Truth مناسب برای آشکارسازی وسایل نقلیه است؛ اما برای بخش‌بندی مناسب نیست؛ بنابراین، در این مطالعه، داده‌های مناسب برای هر دو مجموعه داده برای آموزش آشکارساز VDS_YOLO و شبکه بخش‌بندی‌کننده LSDN با استفاده از نرم‌افزار MATLAB در برنامه کاربردی Video-Labeler^{۲۲} آماده شده است (شکل (۱۲)). در این برنامه کاربردی با استفاده از برچسب‌زدن‌های مستطیلی مناطق مورد علاقه^{۲۳} به‌صورت دستی برای آشکارسازی و برچسب‌زدن تمام پیکسل‌های وسایل نقلیه



شکل (۹): انتخاب شبکه بازگشتی پیش‌بینی‌کننده موقعیت با توجه به خروجی طبقه‌بند مانور

۴- نتایج و بحث

در این بخش، داده‌های UA-DETRAC و Highway از مجموعه داده‌های CDNet2014 برای ارزیابی الگوریتم AVD پیشنهادی برای تعیین موقعیت وسیله نقلیه استفاده می‌شود. الگوریتم AVD از چندین شبکه عمیق تشکیل شده است. آموزش شبکه تشخیص VDS_YOLO و شبکه قطعه‌بندی تفاضلی سبک‌وزن LSDN طراحی شده برای ساخت بلوک VBS در شبکه VDS_YOLO با استفاده از بهینه‌ساز Adam^{۲۰} بهینه شده است. بهینه‌ساز SGD برای آموزش شبکه‌های پیش‌بینی مبتنی بر LSTM استفاده می‌شود. با مقایسه میانگین دقت^{۲۱} (AP) و میانگین زمان اجرای الگوریتم AVD با تحقیقات قبلی، عملکرد آن ارزیابی می‌شود. کدنویسی در محیط برنامه‌نویسی و

$$mAP = \frac{1}{M} \sum_{i=1}^M AP_i \quad (16)$$

همچنین، تعداد پارامترهای شبکه پیشنهادی خود و تعداد محاسبات آن با سایر آشکارسازها مقایسه می‌شود.

۴-۳- نتایج ارزیابی

در این مطالعه، آشکارساز VDS-YOLO شبکه بخش بندی کننده LSDN، شبکه طبقه بندی کننده مسیر و شبکه های پیش بینی موقعیت با استفاده از الگوریتم های بهینه سازی ADAM و SGD و اندازه دسته ۲۹ و دوره ۳۰ ۱۰۰ آموزش داده شده اند. اندازه دسته، تعداد نمونه های استفاده شده در هر بار به روزرسانی وزن های یک شبکه را نشان می دهد. تعداد متفاوت اندازه دسته آزمایش شد و با توجه به اندازه بزرگ تصاویر مجموعه داده UA_DETTRAC که برابر با ۹۶۰×۵۴۰ است، برای دستیابی به سرعت بالاتر و اجتناب از خطای حافظه، اندازه دسته ۴ در نظر گرفته شد. همچنین، دیگر به این دلیل که ورودی شبکه تصاویر متوالی است و این تصاویر شبیه به هم اند، اندازه دسته ۴ یا ۸ یا ۱۶ تفاوت بسیار اندکی در دقت دارند. هاپرپارامتر دیگری که در آموزش شبکه باید تنظیم شود، نرخ یادگیری است که می تواند تأثیر چشمگیری بر عملکرد مدل داشته باشد. نرخ یادگیری کوچک منجر به همگرایی آهسته و به روزرسانی دقیق تر وزن ها می شود؛ در حالی که نرخ یادگیری زیاد منجر به همگرایی سریع، اما به روزرسانی غیردقیق وزن ها می شود. انتخاب نرخ یادگیری مناسب معمولاً با آزمون و خطا انجام می شود؛ اما یک روش مناسب در بهینه سازی این است که در دوره های اول آموزش نرخ یادگیری بزرگ تر، انتخاب و در دوره های بعدی به تدریج کوچک تر شود. در واقع، نرخ یادگیری با زمان نسبت عکس داشته باشد. این روش باعث همگرایی شبکه به مقدار بهینه سراسری و اجتناب از گرفتار شدن در بهینه های محلی می شود [۲۷].

بر این اساس، شبکه پیشنهادی در ۳۰ دوره اول با نرخ یادگیری ۰,۰۰۱ و در ۳۰ دوره بعدی با نرخ یادگیری ۰,۰۰۰۱ و در ۴۰ دوره آخر با نرخ یادگیری را ۰,۰۰۰۰۱

برای بخش بندی روی تصاویر متوالی، Ground-Truth مناسب تهیه شده است.

۴-۲- روش های ارزیابی

در این مطالعه، شبکه آشکارساز با استفاده از مجموعه داده های Highway و UA-DETRAC آموزش داده شده و سپس، عملکرد آن با استفاده از معیارهای ارزیابی رایج مانند متوسط دقت (AP) و محاسبه میانگین زمان اجرا یا نرخ فریم^{۲۴} (FPS) ارزیابی شده است. میانگین زمان اجرا سرعت تشخیص را در فریم های متوالی اندازه گیری می کند. متوسط دقت نیز دقت را برای سطوح مختلف معیار ارزیابی^{۲۵} اندازه گیری می کند. معیارهای دقت و بازیابی با استفاده از معادله های (۱۴ و ۱۳) محاسبه می شود.

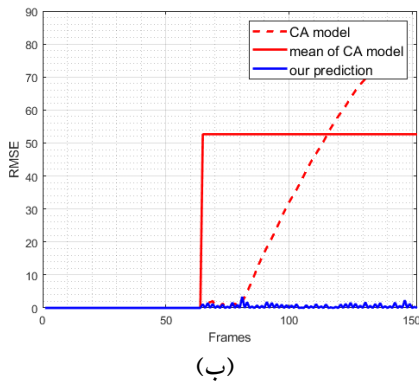
$$precision = \frac{TP}{TP + FP} \quad (13)$$

$$recall = \frac{TP}{TP + FN} \quad (14)$$

در این معادلات، مثبت واقعی^{۲۶} مربوط به زمانی است که مدل به درستی هدف مدنظر را تشخیص می دهد. مثبت کاذب^{۲۷} زمانی به وجود می آید که الگوریتم به اشتباه، هدف را تشخیص می دهد. در واقع، قسمتی از پس زمینه را به عنوان هدف یا پیش زمینه معرفی می کند. منفی کاذب^{۲۸} زمانی اتفاق می افتد که الگوریتم هدف واقعی را به عنوان پس زمینه تشخیص می دهد. در این حالت، گفته می شود مدل هدف را از دست داده است؛ بنابراین، بازیابی، زمانی کاهش می یابد که هدف از دست برود و دقت، زمانی کاهش می یابد که تشخیص های غلط یا به عبارت دیگر، مثبت کاذب وجود داشته باشد. از رابطه (۱۵) برای تعیین متوسط دقت برای یک کلاس استفاده می شود:

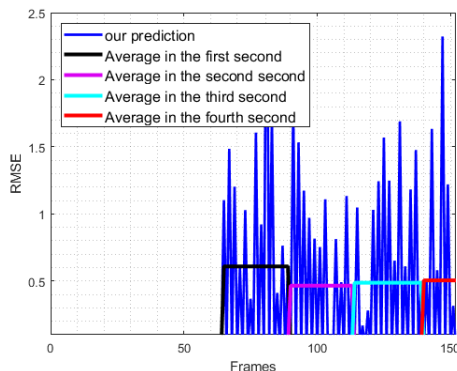
$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) pre(r_{i+1}) \quad (15)$$

جایی که $r_1, r_2, r_3, \dots, r_n$ سطوح پوشش هستند که ابتدا دقت آنها درون یابی می شود، سپس میانگین متوسط دقت در تمام کلاس ها به صورت رابطه (۱۶) محاسبه می شود:



شکل (۱۰): پیش‌بینی مسیر و خطای RMSE مربوط به پیش‌بینی موقعیت وسیله نقلیه

در شکل (۱۱)، نتایج الگوریتم AVD که به‌طور متناوب از شبکه‌های پیش‌بینی و آشکارسازی استفاده می‌کند، نشان داده شده است.

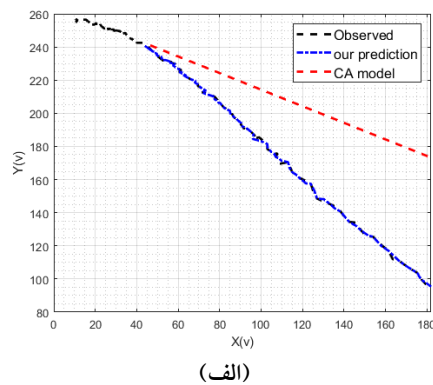


شکل (۱۱): میانگین خطای RMSE در ثانیه‌های متوالی برای پیش‌بینی مسیر وسیله نقلیه در الگوریتم AVD

همان‌طور که مشاهده می‌شود خطا در الگوریتم پیش‌بینی ما برخلاف سایر تحقیقات [20 و 21]، صعودی نیست و این نتیجه، دستاورد مهمی است که به دلیل جلوگیری از انحراف شبکه پیش‌بینی‌کننده حاصل شده است. استفاده متناوب از آشکارساز در طول زمان در الگوریتم AVD از انحراف شبکه پیش‌بینی‌کننده جلوگیری می‌کند و میزان خطا را ۵۰٪ کاهش می‌دهد. در واقع، شبکه‌های پیش‌بینی و تشخیص در الگوریتم AVD مکمل یکدیگرند. همچنین، پیش‌بینی ما دقیق‌تر از [۲۲] است؛ زیرا تأثیر عوامل ترافیکی در طبقه‌بندی مانور خود نیز محاسبه شده است. شکل (۱۲) نمودار دقت و شکل (۱۳) نتایج

آموزش می‌بیند.

شبکه آشکارساز VDS_YOLO، با استفاده از بلوک VBS که در شبکه LSDN آموزش دیده و با روش یادگیری انتقالی به آشکارساز اضافه شده است، دقت الگوریتم AVD را افزایش می‌دهد. شبکه‌های پیش‌بینی بازگشتی در این الگوریتم، سرعت مکان‌یابی خودرو را افزایش می‌دهند و شبکه طبقه‌بندی‌کننده مسیر، دقت پیش‌بینی را افزایش می‌دهد. پیش‌بینی مبتنی بر شبکه نسبت به روش‌های مرسوم پیش‌بینی موقعیت مانند مدل شتاب ثابت، دقت بیشتری دارد. در شبکه‌های بازگشتی که به‌منظور پیش‌بینی استفاده می‌شوند، از فرض‌های نادرستی مانند ثابت در نظر گرفتن سرعت یا شتاب وسیله نقلیه استفاده نمی‌شود؛ بلکه قبل از اجرا، مسیرهای ممکن برای حرکت خودرو به این شبکه‌ها آموزش داده می‌شود. در عمل، آنها با توجه به تاریخچه حرکت خودرو، مکان خودرو را در فریم بعدی پیش‌بینی می‌کنند. در این مطالعه، شبکه‌های مجزا برای سه مسیر حرکت خودرو با حالت حفظ خط، تغییر مسیر به سمت چپ و تغییر مسیر به سمت راست در نظر گرفته شده است. این کار دقت پیش‌بینی را افزایش می‌دهد. در شکل (۱۰) خطای RMSE با در نظر گرفتن تاریخچه زمانی برابر با ۶۴ فریم گذشته برای شبکه پیشنهادی، محاسبه و نمودار آن رسم شده است؛ در حالی که در مدل شتاب ثابت، تنها از یک فریم قبلی برای پیش‌بینی موقعیت استفاده می‌شود. همان‌طور که در این شکل نشان داده شده خطای روش پیشنهادی ما بسیار کمتر از مدل شتاب ثابت است.



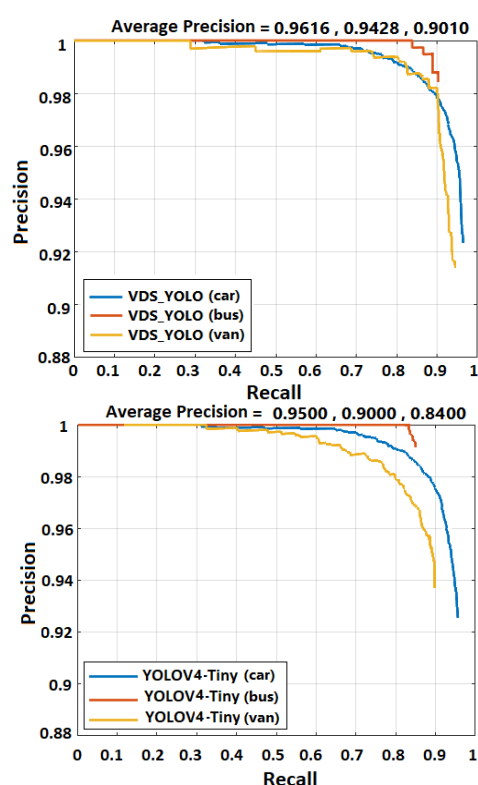
(الف)

VDS_YOLO و استفاده از میانگین‌گیری کانال (CWA) در ماژول SME باعث می‌شود در لایه‌های ابتدایی آشکارساز، نقشه‌های ویژگی تولید شود که در آن وسایل نقلیه به خوبی از پس زمینه تصویر متمایز می‌شوند. با وجود این مکانیسم، دقت شبکه آشکارساز پیشنهادی در عین طراحی سبک از نظر تعداد پارامتر، با آشکارسازهای بلادرنگ مقایسه پذیر است. عامل دیگری که به افزایش دقت آشکارساز پیشنهادی کمک می‌کند، استفاده از سرهای آشکارسازی متعدد در مقیاس‌های مختلف است که چالش آشکارسازی اجسام کوچک YOLO را بهبود می‌بخشد. علاوه بر این، استفاده از الگوریتم خوشه‌بندی سه‌بعدی برای انتخاب تعداد بیشتری از لنگرها و تعیین اندازه‌های دقیق‌تر آنها و استفاده از بلوک TRF در لایه‌های ابتدایی آشکارساز، آشکارساز را در برابر تغییرات مقیاس خودرو مقاوم می‌کند. سرعت آشکارساز VDS_YOLO پیشنهادی با کاهش پارامترها در آشکارساز افزایش می‌یابد؛ در نتیجه، نیازهای سخت‌افزاری آشکارساز کاهش می‌یابد؛ در حالی که سرعت آن افزایش می‌یابد.

جدول (۱) آشکارساز ما را با جدیدترین شبکه‌های آشکارسازی در تصاویر متوالی از داده‌های UA-DETRAC که دارای سه نوع وسیله نقلیه است و داده‌های Highway که دارای یک نوع وسیله نقلیه است، مقایسه می‌کند. همان‌طور که مشاهده می‌شود آشکارساز ما از نظر دقت با آنها مقایسه‌پذیر است و دقت بهتری دارد. جدول (۲) پارامترهای قابل‌یادگیری و تعداد محاسبات آشکارساز پیشنهادی ما را با سایر آشکارسازها مقایسه می‌کند. آشکارساز پیشنهادی ما سریع‌تر و مقرون به صرفه‌تر است؛ زیرا پارامترهای قابل‌یادگیری کمتری نسبت به رقیب دارد.

در جدول (۱) آشکارساز پیشنهادی و سایر شبکه‌های بلادرنگ از لحاظ دقت مقایسه شده‌اند. در جدول (۲)، از لحاظ تعداد پارامترها و تعداد محاسبات مقایسه انجام شده است. همان‌طور که دیده می‌شود شبکه پیشنهادی VDS_YOLO، ۳۲٪ از شبکه YOLOV4_TINY سبک‌تر است؛ با این حال، ۲۸٪ برای مجموعه داده Highway و ۴۶٪ برای مجموعه داده UA_DETRAC دقیق‌تر است. دلیل این برتری، علاوه بر بلوک TRF و

دیداری پیاده‌سازی الگوریتم تشخیص را برای مجموعه داده‌های استفاده‌شده در این مطالعه نشان می‌دهد. همان‌طور که در نمودارهای شکل (۱۲) دیده می‌شود افزایش دقت در آشکارساز پیشنهادی VDS_YOLO برای اتوبوس و مینی‌بوس بیشتر است. دلیل این امر، علاوه بر ماژول SME، انتخاب دقیق تعداد لنگرها براساس معیار MeanIoU و تعیین اندازه آنها براساس الگوریتم خوشه‌بندی K-means است.

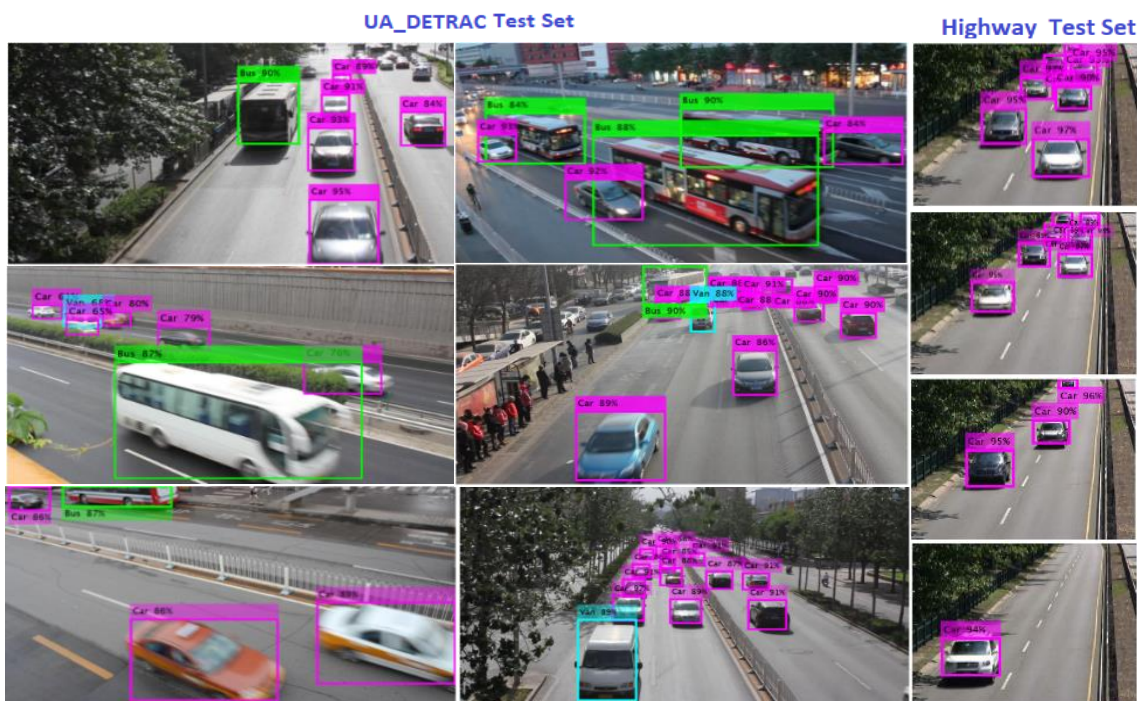


شکل (۱۲): مقایسه منحنی‌های دقت برای آشکارساز پیشنهادی

YOLOV4_Tiny و آشکارساز VDS_YOLO

در این مطالعه، برای وسایل نقلیه‌ای که آشکارساز بیش از یک جعبه محیطی را تعیین می‌کند، از فیلتر حذف غیرحداکثرها^{۳۱} استفاده می‌شود و جعبه محیطی با امتیاز آشکارسازی بالاتر انتخاب می‌شود. همان‌طور که مشاهده می‌شود دقت شبکه آشکارساز پیشنهادی با آخرین نسخه‌های آشکارساز YOLO که از لحاظ حجم پارامترها و تعداد محاسبات نیز سنگین نیستند، مقایسه‌پذیر است. استفاده از یادگیری انتقالی در آموزش آشکارساز

انتخاب مناسب انکورها، تولید نقشه‌های ویژگی است. قطعه بندی شده و استفاده از آنها در آشکارساز پیشنهادی



شکل (۱۳): نتایج دیداری آشکارساز پیشنهادی

جدول (۱): مقایسه عملکرد آشکارساز پیشنهادی و سایر شبکه‌های بلادرنگ از لحاظ میانگین دقت

Methods	Input	Saliency Attention Mechanism	mAP	UADETRAC			Highway
				Car	Bus	Van	car
SSD300 [23]	300	False	0.7418	0.84	0.82	0.72	0.9416
RefineDet320 [23]	320	False	0.7697	0.86	0.85	0.76	0.9600
Faster R-CNN [23]	-	False	0.7267	0.84	0.85	0.70	0.9520
YOLOV2 TINY [24]	416	False	0.6900	0.72	0.67	0.68	0.7400
YOLOV3 TINY [25]	416	False	0.7800	0.8000	0.780	0.76	0.8421
YOLOV4 TINY [26]	416	False	0.8966	0.9500	0.900	0.8400	0.9400
SSAM-YOLO [22]	416	True	0.9206	0.9508	0.9306	0.8804	0.9640
VDS_YOLO (ours)	416	True	0.9426	0.9860	0.9428	0.9010	0.9684

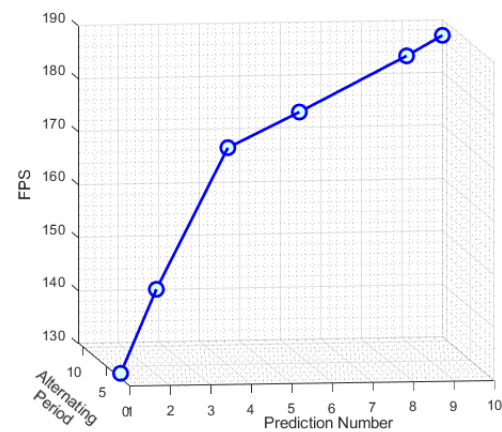
جدول (۲): مقایسه آشکارسازی پیشنهادی و شبکه‌های آشکارساز مبتنی بر YOLO از لحاظ تعداد پارامترها و محاسبات

YOLO-base methods	Backbone	Number of Parameters $\times 10^6$	FLOPs $\times 10^9$
YOLOV2_TINY [24]	FCCL[60]	7.47	5.4
YOLOV3_TINY [25]	FCCL[60]	7.56	5.5
YOLOV4_TINY [26]	CspDarknet53 Tiny	6.06	4.3
SSAM-YOLO [22]	SemAtt_Net	4.28	3.1
VDS_YOLO (ours)	VDS_Net	4.12	3

فریم‌های کوچک‌تر، تقسیم و سپس در هر بسته فریم در الگوریتم پیشنهادی، تصاویر متوالی ورودی به بسته

ویژگی بهتری را نسبت به سایر آشکارسازهای خانواده YOLO ایجاد می‌کنند که باعث افزایش دقت آشکارساز VDS_YOLO در عین سبک و کم‌حجم بودن آن می‌شود. به دلیل وجود این مکانیسم، در آموزش آشکارساز از روش یادگیری انتقالی استفاده شده است. این مکانیسم با طراحی یک شبکه سبک بخش بندی‌کننده مبتنی بر U-NET به نام LSDN اجرا شده است. ورودی این شبکه، تصاویر تفاضلی است که داده‌های مربوط به رنگ و بافت تصویر و همبستگی‌های مربوط به پس‌زمینه تصویر در آن حذف شده و امکان طراحی شبکه LSDN به صورت سبک و کم‌عمق را فراهم آورده است. علاوه بر آن، الگوریتم پیشنهادی به کاربر اجازه می‌دهد با تغییر تعداد دفعات استفاده از شبکه پیش‌بینی و آشکارسازی و تغییر طول دوره تناوب مدنظر در الگوریتم AVD، سرعت و دقت موردنیاز را با توجه به کاربردهای مختلف تنظیم کند. استفاده از بلوک TRF در لایه‌های ابتدایی آشکارساز که از اتصالات باقیمانده بهره می‌گیرد، مقاومت آشکارساز را در برابر تغییرات مقیاس افزایش می‌دهد. عامل دیگر افزایش دقت آشکارساز، تعداد مناسب لنگرها و خوشه‌بندی داده‌های آموزشی در فضای سه‌بعدی است. همه عوامل ذکرشده، آشکارساز پیشنهادی را از لحاظ دقت و سرعت با آشکارسازهای بلادرنگ مقایسه‌پذیر می‌سازد. استفاده از شبکه پیش‌بینی‌کننده در الگوریتم پیشنهادی نیز، میانگین زمان اجرای الگوریتم را کاهش و سرعت آن را افزایش می‌دهد؛ زیرا با توجه به اینکه در شبکه پیش‌بینی‌کننده موقعیت، تنها از تاریخچه حرکت خودرو استفاده می‌شود و کل تصویر پردازش نمی‌شود، سریع‌تر از شبکه آشکارساز اجرا می‌شود. طبقه‌بندی مانور و مسیر خودرو نیز برای افزایش دقت پیش‌بینی طراحی شده است. در این شبکه، علاوه بر مسیر هر خودرو، مسیر و موقعیت خودروهای اطراف آن خودرو نیز در تعیین مسیر آن در نظر گرفته شده است. وجود داده‌های استخراج‌شده از آشکارساز به‌عنوان ورودی شبکه پیش‌بینی‌کننده نیز عاملی دیگر بر افزایش دقت این شبکه است که علاوه بر داشتن سرعت بالا، قابلیت آن را دوچندان می‌کند. به عبارت دیگر، شبکه‌های طراحی و استفاده‌شده در الگوریتم AVD مکمل یکدیگرند و باعث افزایش سرعت

کوچک n فریمی به تعداد m بار، شبکه پیش‌بینی و $n-m$ بار شبکه آشکارساز اجرا می‌شود و این روال تکرار می‌شود. پارامترهای سوییچینگ بین شبکه‌های آشکارساز و پیش‌بینی‌کننده یعنی n و m تغییر داده و نتایج آزمایشات به صورت نمودار در شکل (۱۴) نمایش داده شدند. همان‌طور که دیده می‌شود سرعت الگوریتم موقعیت‌یابی پیشنهادی، با افزایش تعداد m زیاد می‌شود. در واقع، دقت الگوریتم AVD با طراحی بهینه شبکه پیشنهادی VDS_YOLO و سرعت آن با افزایش تعداد دفعات استفاده از شبکه پیش‌بینی‌کننده محقق می‌شود.



شکل (۱۴): مقایسه بین زمان اجرای الگوریتم پیشنهادی AVD برای یک دوره n فریمی که در آن شبکه پیش‌بینی m دفعه و شبکه تشخیص $n-m$ دفعه اجرا می‌شود.

۵- نتیجه‌گیری

الگوریتم پیشنهادی AVD در این مطالعه سرعت موقعیت‌یابی خودرو را افزایش می‌دهد و از لحاظ تنظیم سرعت و دقت بسیار انعطاف‌پذیر است. این الگوریتم به‌طور متناوب از شبکه آشکارساز مبتنی بر YOLO، طبقه‌بندی‌کننده مانور و شبکه پیش‌بینی بازگشتی در فریم‌های مختلف استفاده می‌کند. براساس پیچیدگی تصاویر و تقاضای کاربر درباره سرعت و دقت، ممکن است از شبکه آشکارساز به‌تنهایی یا همراه با شبکه‌های دیگر استفاده شود. شبکه آشکارساز پیشنهادی به نام VDS_YOLO از تصاویر تفاضلی و یک مکانیسم توجه مبتنی بر بخش‌بندی پیشنهادشده در این مطالعه استفاده می‌کند و نقشه‌های

networks for normal and abnormal situation recognition in the automatic access monitoring and control system of vehicles," *Neural Computing and Applications*, Vol. 33, No. 8, pp. 3069-3083, Apr 2021.

- [12] G. Singh, M. Pal, Y. Yadav, and T. Singla, "Deep neural network-based predictive modeling of road accidents," *Neural Computing and Applications*, Vol. 32, pp. 12417-12426, Aug 2020.
- [13] W. Chu, Y. Liu, C. Shen, D. Cai, and X.-S. Hua, "Multi-task vehicle detection with region-of-interest voting," *IEEE Transactions on Image Processing*, Vol. 27, No. 1, pp. 432-441, Oct 2017.
- [14] K. B. Lee and H. S. Shin, "An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels," in *2019 International Conference on deep learning and machine learning in emerging applications (Deep-ML)*, 2019: IEEE, pp. 7-11.
- [15] Z. Chen *et al.*, "Vehicle detection in high-resolution aerial images based on fast sparse representation classification and multiorder feature," *IEEE transactions on intelligent transportation systems*, Vol. 17, No. 8, pp. 2296-2309, 2016.
- [16] N. K. Chauhan and K. Singh, "A review on conventional machine learning vs deep learning," in *2018 International conference on computing, power and communication technologies (GUCON)*, IEEE, pp. 347-352, Sep 2018.
- [17] C.-T. Lam, B. Ng, and C.-W. Chan, "Real-time traffic status detection from on-line images using generic object detection system with deep learning," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, IEEE, pp. 1506-1510, Oct 2019.
- [18] K. S. Chandrasekar and P. Geetha, "A new formation of supervised dimensionality reduction method for moving vehicle classification," *Neural Computing and Applications*, vol. 33, pp. 7839-7850, 2021.
- [19] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Computing and Applications*, Vol. 33, pp. 613-624, Jan 2021.
- [20] L. Tang, F. Yan, B. Zou, W. Li, C. Lv, and K. Wang, "Trajectory prediction for autonomous driving based on multiscale spatial-temporal graph," *IET Intelligent Transport Systems*, Vol. 17, No. 2, pp. 386-399, 2023.
- [21] J. Yan *et al.*, "Trajectory prediction for

مکان‌یابی با دقت مطلوب می‌شوند.

مراجع

- [1] G. Verma, A. Gupta, S. Bansal, and H. Dhiman, "Monitoring Maritime Traffic with Ship Detection via YOLOv4." pp. 1-7, Feb 2022.
- [2] N. Kavitha and D. Chandrappa, "Optimized YOLOv2 based vehicle classification and tracking for intelligent transportation system," *Results in Control and Optimization*, Vol. 2, p. 100008, Apr 2021.
- [3] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geoscience and Remote Sensing Letters*, Vol. 12, No. 9, pp. 1938-1942, Jun 2015.
- [4] Y. Shen, D. Zhang, Z. Song, X. Jiang, and Q. Ye, "Learning to reduce information bottleneck for object detection in aerial images," *IEEE Geoscience and Remote Sensing Letters*, Apr 2023.
- [5] Y.-F. Lu, J.-W. Gao, Q. Yu, Y. Li, Y.-S. Lv, and H. Qiao, "A Cross-Scale and Illumination Invariance-Based Model for Robust Object Detection in Traffic Surveillance Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, Apr 2023.
- [6] Y.-C. Chiou, L. W. Lan, and C.-M. Tseng, "A novel method to predict traffic features based on rolling self-structured traffic patterns," *Journal of intelligent transportation systems*, Vol. 18, No. 4, pp. 352-366, Oct 2014.
- [7] Y. Chen, D. Zhao, H. Li, D. Li, and P. Guo, "A temporal-based deep learning method for multiple objects detection in autonomous driving," in *2018 international joint conference on neural networks (IJCNN)*, IEEE, pp. 1-6, Jul 2018.
- [8] Y. Cai, H. Wang, Z. Zheng, and X. Sun, "Scene-adaptive vehicle detection algorithm based on a composite deep structure," *IEEE access*, Vol. 5, pp. 22804-22811, Sep 2017.
- [9] Z. Hu, T. Turki, N. Phan, and J. T. Wang, "A 3D atrous convolutional long short-term memory network for background subtraction," *IEEE Access*, Vol. 6, pp. 43450-43459, Jul 2018.
- [10] Y. Li, S. Wang, Q. Tian, and X. Ding, "Learning cascaded shared-boost classifiers for part-based object detection," *IEEE transactions on image processing*, Vol. 23, No. 4, pp. 1858-1871, Feb 2014.
- [11] O. S. Amosov, S. G. Amosova, Y. S. Ivanov, and S. V. Zhiganov, "Using the deep neural

- [25]P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," in *2020 6th international conference on advanced computing and communication systems (ICACCS)*, 2020: IEEE, pp. 687-694.
- [26]Q. Liu, X. Fan, Z. Xi, Z. Yin, and Z. Yang, "Object detection based on Yolov4-Tiny and Improved Bidirectional feature pyramid network," in *Journal of Physics: Conference Series*, 2022, Vol. 2209, No. 1: IOP Publishing, p. 012023.
- [27]C. Darken and J. Moody, "Note on learning rate schedules for stochastic optimization," *Advances in neural information processing systems*, Vol. 3, 1990.
- intelligent vehicles using spatial-attention mechanism," *IET Intelligent Transport Systems*, Vol. 14, No. 13, pp. 1855-1863, 2020.
- [22]N. Zarei, P. Moallem, and M. Shams, "Fast-Yolo-Rec: incorporating yolo-base detection and recurrent-base prediction networks for fast vehicle detection in consecutive images," *IEEE Access*, Vol. 10, pp. 120592-120605, 2022.
- [23]F. Zhang, F. Yang, C. Li, and G. Yuan, "CMNet: A connect-and-merge convolutional neural network for fast vehicle detection in urban traffic surveillance," *IEEE Access*, Vol. 7, pp. 72660-72671, 2019.
- [24]H. R. Alsanad, O. N. Ucan, M. Ilyas, A. U. R. Khan, and O. Bayat, "Real-time fuel truck detection algorithm based on deep convolutional neural network," *IEEE Access*, Vol. 8, pp. 118808-118817, 2020.

¹ Accelerate Vehicle Detection

² Vehicle Differential Segmentation YOLO

³ Backbone

⁴ Neck

⁵ Head

⁶ Segmented Map Extraction

⁷ Three Receptive Field

⁸ Residual

⁹ Freeze

¹⁰ Mixture of Gaussian

¹¹ Lightweight Segmentation Deep Network

¹² Vehicle and Background Segmentation Block

¹³ Global Average Pooling

¹⁴ Channel-Wise Average Thresholding

¹⁵ Concatenation

¹⁶ Anchor Box

¹⁷ Mean Intersection over Union

¹⁸ First In First Out

¹⁹ Gate

²⁰ Adaptive Moment Estimation

²¹ Average Precision

²² Application

²³ Region of Interest

²⁴ Frame Rate

²⁵ Recall

²⁶ True Positive (TP)

²⁷ False Positive (FP)

²⁸ False Negative (FN)

²⁹ Batch Size

³⁰ Epoch

³¹ Non-Maximal Suppression