

تشخیص و شناسایی علائم ترافیکی مبتنی بر یادگیری عمیق با سیستم‌های تعبیه شده

پیمان بابائی*^۱، فائزه علامه^۲

^{۱،۲} گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد تهران غرب، دانشگاه آزاد اسلامی، تهران، ایران.

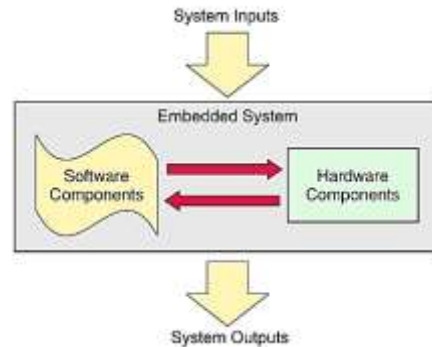
Peyman.Babaei@IAU.ac.ir

چکیده - تشخیص و شناسایی علائم ترافیکی، نیازمند به کارگیری الگوریتم‌های تشخیص و طبقه بندی است و اساساً از اطلاعات بصری مانند شکل و رنگ علائم ترافیکی استفاده می‌کنند. با این حال، این الگوریتم‌ها در آزمون‌های بلادرنگ با اشکالاتی مواجه هستند همچنین دستیابی به تشخیص چند هدف بسیار دشوار و نیازمند تسریع در عملکرد الگوریتم‌های مربوطه است. سیستم‌های تشخیص علائم ترافیکی مبتنی بر شبکه عصبی عمیق ممکن است به دلیل نیازهای محاسباتی و مصرف منابع، محدودیت‌هایی در کاربردهای عملی داشته باشند. اکثر سیستم‌های تعبیه شده به طور مستقیم با فرآیندها یا محیط تعامل دارند و بر اساس ورودی‌های خود تصمیم‌گیری می‌کنند. این امر باعث می‌شود که سیستم واکنش پذیر باشد و در زمان واقعی به ورودی‌های پردازش برای اطمینان از عملکرد صحیح پاسخ دهد. این مقاله یک شبکه عصبی سبک وزن را برای تشخیص علائم ترافیکی ارائه می‌کند که به صحت و دقتی بالا با پارامترهای قابل آموزش کمتر دست می‌یابد. برای ارزیابی کارایی و قابلیت اطمینان، مدل پیشنهادی مبتنی بر شبکه عصبی کانولوشنی به منظور شناسایی علائم ترافیکی، آزمایش‌های گسترده‌ای بر روی مجموعه داده *GTSRB* انجام شده است. همچنین نتایج به دست آمده با چند معماری پیشرفته از جمله *VGG16*، *MobileNetv2* و *ResNet50* مقایسه شده است. نتایج نشان می‌دهد که مدل پیشنهادی به عملکرد خوبی دست یافته است و بر پتانسیل آن برای استقرار مدل شناسایی علائم ترافیکی بلادرنگ و سیستم‌های کمک رانندگی تاکید می‌کند. رانندگان محاسباتی و اندازه کوچک مدل پیشنهادی، آن را برای شناسایی علائم ترافیکی بلادرنگ کاربردی تر و مناسب تر می‌کند.

کلید واژه - تشخیص و شناسایی علائم ترافیکی، سیستم‌های تعبیه شده، شبکه‌های عصبی کانولوشنی.

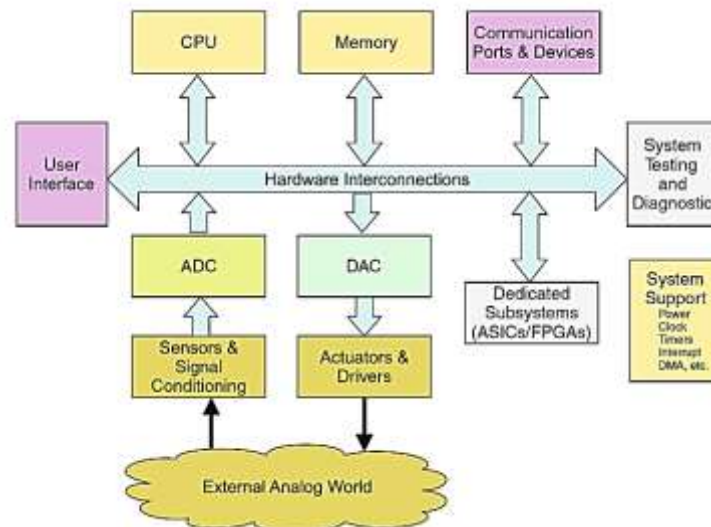
مقدمه

یک سیستم تعبیه شده را می‌توان به طور کلی به عنوان دستگاهی تعریف کرد که شامل اجزای سخت‌افزاری و نرم‌افزاری محکم به هم متصل شده برای انجام یک عملکرد واحد است، بخشی از یک سیستم بزرگ‌تر را تشکیل می‌دهد، به طور مستقل توسط کاربر قابل برنامه‌ریزی است و انتظار می‌رود که با حداقل منابع کار کند. صرف‌نظر از عملکردی که یک سیستم تعبیه شده انجام می‌دهد، ساختار آن دو مجموعه اصلی و محکم از اجزاء را نشان می‌دهد: مجموعه‌ای از قطعات سخت‌افزاری که شامل یک واحد پردازش مرکزی، معمولاً به شکل یک میکروکنترلر است و یک سری از برنامه‌های نرم‌افزاری که معمولاً به عنوان سیستم عاملی است که به سخت‌افزار کارایی می‌دهد. شکل ۱ نمای کلی آن و این دو جزء اصلی و ارتباط متقابل آنها را نشان می‌دهد. ورودی‌ها معمولاً در یک سیستم تعبیه شده، متغیرهای فرآیند و پارامترهایی هستند که از طریق حسگرها و پورت(دروازه)های ورودی/خروجی وارد می‌شوند. خروجی‌ها به صورت اقدامات کنترلی بر روی محرک‌های سیستم یا اطلاعات پردازش شده برای کاربران یا سایر زیرسیستم‌هایی درون برنامه هستند. در برخی موارد، تبادل اطلاعات ورودی/خروجی با کاربران از طریق یک رابط کاربری که ممکن است شامل کلیدها، حسگرها، دیودهای نوری، نمایشگرها و سایر انواع دستگاه‌های نمایشگر باشد، انجام می‌شود.



شکل ۱: نمای کلی یک سیستم تعبیه شده.

در ادامه نگاهی دقیق‌تر به اجزای سخت‌افزار و نرم‌افزاری خواهیم داشت که یک سیستم تعبیه‌شده را یکپارچه می‌کنند. اجزای سخت‌افزاری یک سیستم تعبیه‌شده شامل تمام لوازم الکترونیکی لازم برای انجام عملکردی است که برای آن طراحی شده است. بنابراین، ساختار خاص یک سیستم، بر اساس برنامه، می‌تواند به طور قابل ملاحظه‌ای با سیستم دیگر متفاوت باشد. با وجود این تفاوت‌ها، سه جزء سخت‌افزاری اصلی در یک سیستم تعبیه‌شده ضروری هستند (شکل ۲): واحد پردازش مرکزی (CPU)، حافظه سیستم و مجموعه‌ای از پورت‌های ورودی/خروجی. CPU دستورالعمل‌های نرم‌افزاری را برای پردازش ورودی‌های سیستم و اتخاذ تصمیماتی اجرا می‌کند که عملکرد سیستم را هدایت می‌کند. حافظه برنامه‌ها و داده‌های لازم برای عملکرد سیستم را ذخیره می‌کند. اکثر سیستم‌ها بین حافظه‌های برنامه و داده‌ها تفاوت قائل می‌شوند. حافظه برنامه، برنامه‌های نرم‌افزاری اجرا شده توسط CPU را ذخیره می‌کند. حافظه داده، داده‌های پردازش‌شده توسط سیستم را ذخیره می‌کند. پورت‌های I/O امکان انتقال سیگنال‌ها را بین CPU و دنیای خارج از آن فراهم می‌کنند. فراتر از این نقطه، باتوجه به کاربرد، ممکن است تعدادی دیگر از دستگاه‌های پشتیبانی و ورودی/خروجی مورد نیاز برای عملکرد سیستم وجود داشته‌باشد.



شکل ۲: عناصر سخت‌افزاری در یک سیستم تعبیه شده.

به‌طور کلی سیستم‌های تعبیه‌شده را می‌توان به دو دسته سیستم‌های تعبیه‌شده کوچک و سیستم‌های تعبیه‌شده با کارایی بالا طبقه‌بندی کرد. سیستم‌های تعبیه‌شده کوچک حول یک تراشه میکروکنترلر متمرکز است که کل برنامه را کنترل می‌کند. این سیستم‌ها به شدت یکپارچه هستند و در صورت نیاز فقط چند جزء آنالوگ، حسگرها، محرک‌ها و رابط کاربری را اضافه می‌کنند. این سیستم‌ها با حداقل‌ها کار می‌کنند، هزینه بسیار پایینی دارند و به صورت انبوه تولید می‌شوند. نرم‌افزار در این سیستم‌ها معمولاً تک وظیفه‌ای است و به‌ندرت به RTOS نیاز دارند.

نمونه‌هایی از این سیستم‌ها عبارتند از: سیستم‌های نظارت بر فشار باد تایر و کنترل‌کننده‌های اجاق مایکروویو. سیستم‌های تعبیه شده با کارایی بالا نشان‌دهنده کلاس سیستم‌های تعبیه شده بسیار تخصصی است که به محاسبات سریع، استحکام، تحمل خطا و قابلیت نگهداری بالا نیاز دارند. این سیستم‌ها معمولاً به پردازنده‌های گرافیکی اختصاصی NVIDIA یا ASICS نیاز دارند و ممکن است شامل DSP و FPGA به عنوان بخشی از سخت‌افزار اصلی باشند. در بسیاری از موارد، پیچیدگی نرم‌افزار آنها استفاده از RTOS را برای مدیریت وظایف متعدد الزامی می‌کند. آنها در مقادیر کم تولید می‌شوند و هزینه آنها بسیار بالا است. لذا سیستم‌های تعبیه‌شده‌ای هستند که در کاربردهای نظامی یا هوافضا استفاده می‌شوند، مانند کنترل‌کننده‌های پرواز، سیستم‌های هدایت موشک و سیستم‌های ناوبری فضاپیما. در این طبقه بندی می‌توانیم نواحی خاکستری را پیدا کنیم که در آن ویژگی‌های دو یا سه مورد از آنها همپوشانی دارند و ممکن است برنامه‌ها به یک کلاس خاص مرتبط شوند. با این حال، اگر به طیف وسیعی از برنامه‌های کاربردی تعبیه شده نگاه کنیم، در بیشتر موارد، تشخیص کلاسی که یک برنامه خاص به آن تعلق دارد، به طور کلی آسان می‌شود.

یادگیری عمیق و شبکه‌های عصبی کانولوشنی

عملیات کانولوشنی

عملیات ریاضی دو تابع f و g برای به دست آوردن تابع سوم را کانولوشنی می‌نامند. در واقع، انتگرالی که مقدار همپوشانی g را با جابجایی آن بر روی تابع f تعیین می‌کند، کانولوشن نامیده می‌شود. به طور کلی، عملگر ستاره عملیات کانولوشنی را بیان می‌کند.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

تابع کانولوشنی یک میانگین وزنی از $f(t)$ است. تابع وزنی g زمانی که تابع t تغییر می‌کند تأکید بیشتری بر ورودی تابع f دارد.

در اصطلاحات شبکه کانولوشنی، تابع f به عنوان ورودی، نقشه ویژگی به عنوان خروجی و تابع g به عنوان هسته شناخته می‌شود. در سیستم‌های دیجیتال، تابع کانولوشنی می‌تواند با فرض f زمانی که g روی مقادیر صحیح t تعریف می‌شود، گسسته شود:

$$(f * g)(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

در شبکه‌های عصبی کانولوشنی، آرایه‌های چند بعدی به عنوان ورودی و آرایه چند بعدی پارامترها که تنسور نامیده می‌شوند، به عنوان هسته در نظر گرفته می‌شوند. به عنوان مثال، در یک تصویر دو بعدی با ورودی I ، از دو عدد هسته دوبعدی K استفاده شده است.

$$S(i, j) = (I * k)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

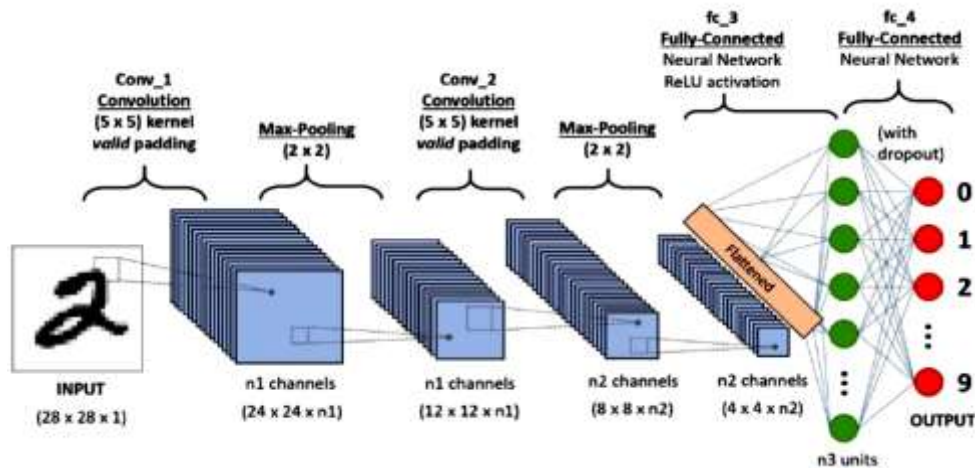
و نقشه ویژگی (یا خروجی) مکان S است.

تعامل هر واحد ورودی با هر واحد خروجی توسط ماتریسی از پارامترها با پارامترهای جداگانه توصیف می‌شود. بنابراین، لایه‌های شبکه عصبی کاملاً متصل از ضرب ماتریسی استفاده می‌کنند. با پرداختن به ورودی‌های با ابعاد بیشتر مانند تصاویر، اتصال نوروها به همه نوروها غیرعملی خواهد بود، بنابراین شبکه‌های کاملاً متصل از تصاویر قابل مقیاس نیستند. به کارگیری یک هسته کوچکتر از ورودی باعث می‌شود عملیات کانولوشن، هر نورو را فقط به یک ناحیه محلی از ورودی متصل کند. مقایسه شبکه‌های کاملاً متصل سنتی با شبکه‌های کانولوشنی از نظر کارایی بسیار موثر است. الگوریتم‌های هر نمونه دارای $O(m \times n)$ زمان اجرا و ضرب ماتریسی نیاز به $m \times n$ پارامتر دارند (m ورودی، n خروجی). برای

اینکه الگوریتم‌ها را به پارامترهای $k \times n$ و زمان اجرا را به $O(k \times n)$ برسانیم، باید تعداد اتصالات هر خروجی را به K محدود کنیم. به عنوان مثال، برای پردازش یک تصویر، هزاران یا میلیون‌ها پیکسل ممکن است در ورودی باشد. از لایه‌های دارای هسته می‌توان برای کاهش تعداد پیکسل‌های مصرف شده به ده‌ها یا صدها پیکسل استفاده کرد. ذخیره پارامترهای کمتر با استفاده از کانولوشن امکان‌پذیر است و منجر به کاهش تقاضای محاسباتی شبکه می‌شود. به جای یادگیری مجموعه‌ای جداگانه از پارامترها برای هر مکان، استفاده از هر یک از اعضای هسته در هر موقعیت ورودی منجر به اشتراک‌گذاری پارامترها در شبکه‌های کانولوشنی می‌شود. در یک شبکه کاملاً متصل معمولاً برای محاسبه خروجی یک لایه، هر عنصر ماتریس فقط یک بار استفاده می‌شود.

معماری شبکه‌های عصبی کانولوشنی

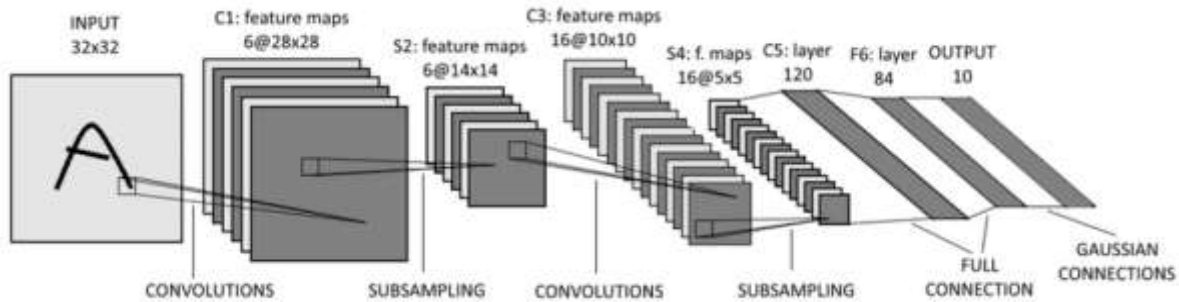
زیر مجموعه کاربردی از شبکه‌های عصبی به نام شبکه‌های عصبی کانولوشنی (CNN) معرفی شدند. دلیل ارائه شبکه‌های عصبی و شبکه‌های عصبی کانولوشنی این است که با پیشرفت دستگاه‌های هوشمند و رسانه‌های اجتماعی مختلف، حجم زیادی از داده‌ها در اینترنت در کسری از زمان تولید می‌شود. با این حال، این منابع داده گسترده، نمی‌تواند توسط انسان‌ها به‌طور دقیق پردازش شود. از این رو، انسان‌ها برای پردازش خودکار داده‌های گسترده به کمک رایانه نیاز دارند. علاوه بر این، مشکلات طبقه‌بندی، تشخیص شیء و الگو را می‌توان توسط معماری فوق‌العاده شبکه‌های عصبی کانولوشنی انجام داد که سریع‌تر و کارآمدتر است. شبکه‌های عصبی کانولوشنی نه تنها می‌توانند شبکه‌های عمیق‌تری ایجاد کنند، بلکه روند آموزش را تسریع می‌کنند. با توجه به این روند آموزشی سریع، شبکه‌های عصبی کانولوشنی منجر به ایجاد شبکه‌های عمیق‌تر می‌شوند. اگرچه شبکه‌های عصبی مرسوم از نمایش‌برداری استفاده می‌کنند، عملکرد شبکه‌های عصبی کانولوشنی بر روی تصاویر بر اساس ارتفاع، عرض و تعداد کانال‌ها است. برای درک راحت‌تر مفاهیم، در لایه‌های شبکه‌های عصبی کانولوشنی، پیکسل‌ها توسط یک فیلتر یا هسته که مجموعه‌ای از بایاس‌ها و وزن‌ها است، روی لایه پنهان بعدی قرار می‌گیرند. گاهی اوقات فیلترها می‌توانند یک آشکارساز ویژگی مانند یک آشکارساز لایه عمودی یا افقی باشند. اعمال فیلتر بر روی یک تصویر معمولاً اندازه تصویر را کاهش می‌دهد. با این حال، تعداد کانال‌ها افزایش خواهد یافت. ساده‌سازی اطلاعات در خروجی لایه کانولوشن بسیار مهم است، زیرا این کار را می‌توان با کاهش لایه کانولوشن انجام داد. بنابراین، مفهوم ادغام معرفی شد. یکی از مهم‌ترین تفاوت‌های شبکه‌های عصبی معمولی با شبکه‌های عصبی کانولوشن، وجود لایه ادغام است. به‌طور کلی، شبکه‌های عصبی کانولوشنی از لایه کانولوشنی تشکیل شده‌اند که به عنوان لایه‌های کانولوشنی، لایه‌های ادغام و به دنبال آن یک لایه کاملاً متصل در انتها شناخته می‌شود که در شکل ۳ نشان داده شده است.



شکل ۳: معماری یک شبکه عصبی کانولوشنی

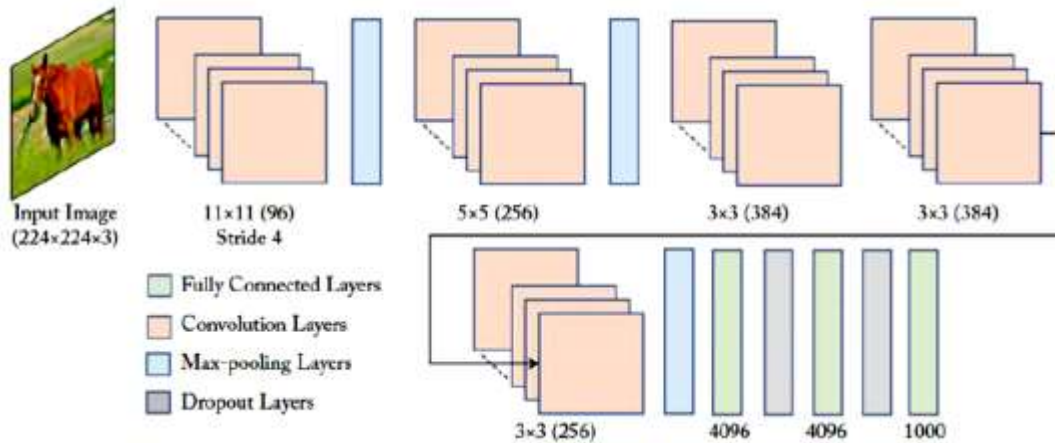
مانند شبکه‌های عصبی کلاسیک، شبکه‌های عصبی کانولوشنی از توابع فعال‌سازی مختلفی در لایه‌های خروجی خود استفاده می‌کنند که ورودی لایه‌های بعدی است. تابع خطی ReLU، تابع سیگموئید و تابع تانژانت هیپربولیک انواع مختلفی از توابع فعال‌سازی هستند که می‌توانند برای

شبکه‌های عصبی کانولوشنی اعمال شوند. درک صحنه، تشخیص و شناسایی اشیاء و استخراج ویژگی مسائل بینایی رایانه‌ای مواردی هستند که به عنوان معماری‌های پیشنهادی شبکه‌های عصبی کانولوشنی دسته‌بندی می‌شوند. در سال ۱۹۹۰، مدلی توسعه داده شد که می‌تواند به عنوان اولین کاربرد موفق شبکه‌های عصبی کانولوشنی به حساب آید. کاربردهای متعددی در مسائل بینایی رایانه‌ای مانند تشخیص ارقام دارد. معماری LeNet در شکل ۴ نشان داده شده است.



شکل ۴: معماری LeNet [۱]

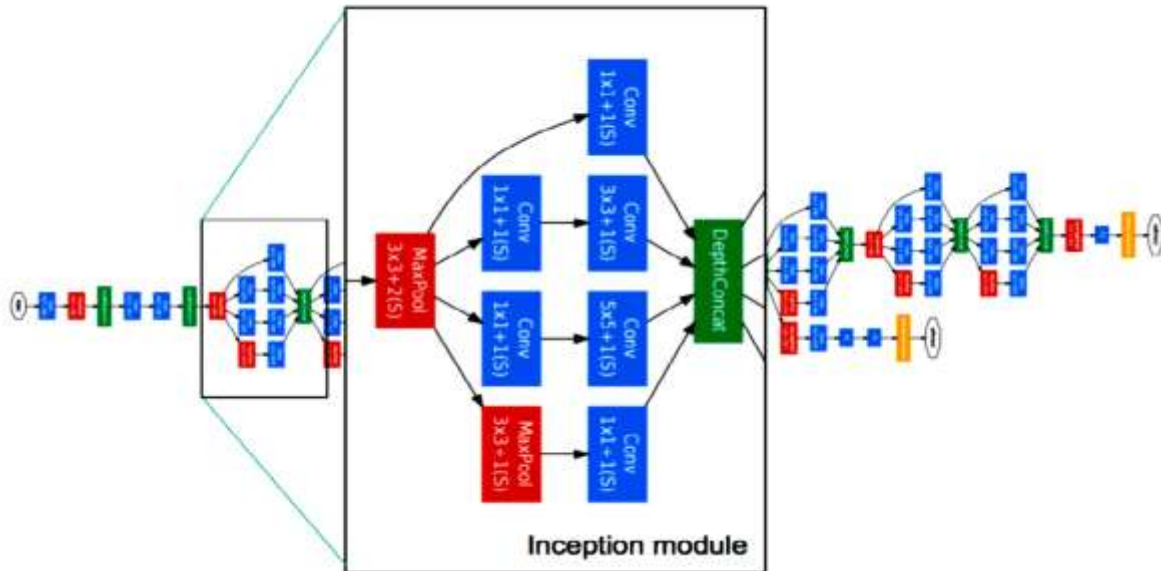
معماری AlexNet، یکی از نخستین آثاری است که شبکه‌های عصبی کانولوشنی را در بینایی رایانه‌ای رواج داد. اگرچه این شبکه، بسیار شبیه به LeNet بود، اما دارای لایه‌های کانولوشنی بزرگ‌تر و عمیق‌تر بود که روی هم چیده شده بودند. AlexNet در سال ۲۰۱۲ به چالش ILSVRC فرستاده شد. این معماری شامل ۶۳۰ میلیون اتصال، ۶۰ میلیون پارامتر و ۶۳۰ میلیون نورون است. (شکل ۵)



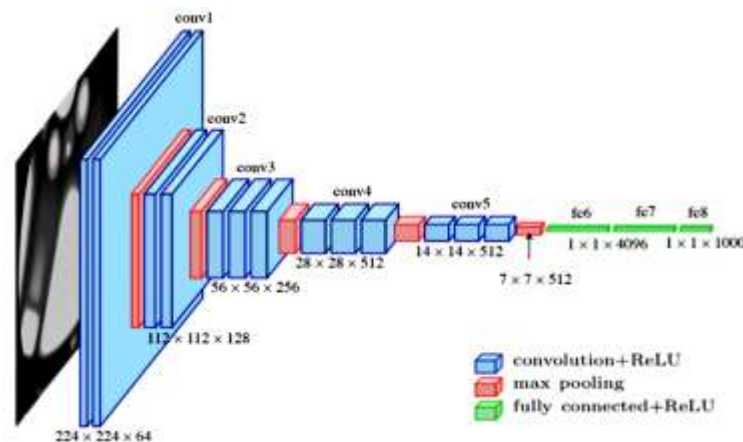
شکل ۵: معماری AlexNet [۱]

این شبکه GoogleNet در سال ۲۰۱۴ رتبه اول را در چالش تشخیص تصویر در مقیاس بزرگ (ILSVRC) کسب کرد. تعداد پارامترها به طور چشمگیری، از ۶۰ میلیون به ۴ میلیون در مقایسه با AlexNet کاهش یافت. این معماری، ترکیبی از لایه‌های ادغام و کانولوشن به همراه ماژول‌های آغازین آنها بود. GoogleNet اولین شبکه‌ای است که از معماری مدرن شبکه‌های عصبی کانولوشنی استفاده می‌کند که فقط برای لایه‌های کانولوشن و ادغام استفاده نمی‌شود، بلکه در معماری inception نیز کاربرد دارد. همانطور که در شکل ۶ نشان داده شده است، به عنوان inception network نیز شناخته می‌شود. علاوه بر این، در مسیرهای موازی با اندازه مختلف نقشه ویژگی‌ها کار می‌کند. مزیت ماژول inception پرش از اتصالات در شبکه و ایجاد مینی ماژول است که از طریق شبکه تکرار می‌شود. هدف این ماژول کاهش تعداد پارامترها در شبکه است. هدف اصلی ماژول قرار دادن تمام بلوک‌های اساسی پردازش در سیستم موازی است. مزیت اصلی معماری GoogleNet ایجاد شبکه بزرگ با انباشتن چندین ماژول در کنار هم است.

VGGNet یکی دیگر از شبکه‌های عصبی کانولوشنی است که در چالش تشخیص تصویری در مقیاس بزرگ ImageNet (ILSVRC) در سال ۲۰۱۴ به عملکرد بالایی دست یافت. در آن زمان، VGGNet به عنوان یک شبکه عصبی پیچیده بسیار عمیق در نظر گرفته می‌شد. این به توضیح عمق شبکه برای افزایش دقت تشخیص یا طبقه‌بندی کمک می‌کند. اگرچه این شبکه بسیار عمیق است، اما سادگی آن قابل توجه است. این معماری شبکه از ابتدا تا انتها فقط از لایه‌های کانولوشن 3×3 و ادغام 2×2 استفاده می‌کند که در شکل ۷ نشان داده شده است. یکی از بزرگ‌ترین نقاط ضعف VGGNet این است که از حافظه و پارامترهای زیادی (۱۴۰ میلیون) استفاده می‌کند که به نوبه خود هزینه بیشتری دارد.

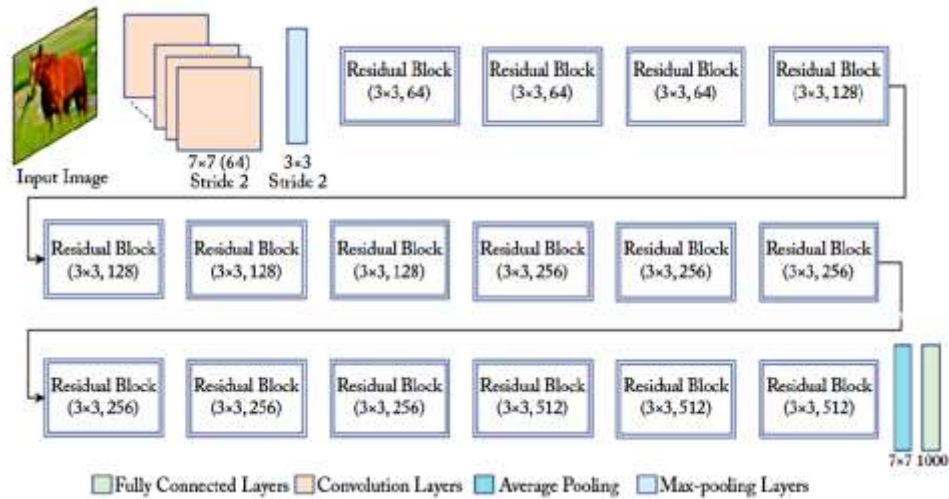


شکل ۶: معماری ماژول inception [۱]



شکل ۷: معماری VGG-16 [۱]

ResNet یک شبکه عصبی کانولوشنی عمیق است. در چالش تشخیص تصویر در مقیاس بزرگ ImageNet (ILSVRC) در سال ۲۰۱۵ رتبه نخست را کسب کرد. در مجموع از ۱۵۲ لایه تشکیل شده بود. به دلیل توانایی آن در آموزش یک مدل عمیق 100×1 لایه با استفاده از اتصالات باقیمانده تحت عنوان اتصالات باقیمانده شناخته شد. مزیت اصلی استفاده از اتصالات باقیمانده، کسب دانش در حین آموزش و افزایش سرعت زمان آموزش است. ResNet از نرمال‌سازی دسته‌ای بعد از هر لایه استفاده می‌کند و همانطور که در شکل ۸ نشان داده شده است، از اتصال پرش برای بهینه‌سازی ساده استفاده می‌کند.



شکل ۸: معماری ResNet [۱]

یادگیری عمیق و تشخیص علائم ترافیکی

حوزه تشخیص علائم ترافیکی در سال‌های اخیر شاهد پیشرفت‌های متعددی بوده‌است و بسیاری از محققان برای توسعه الگوریتم‌های کارآمد و دقیق به تکنیک‌های یادگیری عمیق روی آورده‌اند. در [۲] از یک شبکه عصبی کانولوشنی (CNN) با یک مدل از پیش آموزش دیده MobileNet، برای تشخیص علائم ترافیکی استفاده کردند. معماری MobileNet به گونه‌ای طراحی شده‌است که سبک و کارآمد باشد و برای استفاده در برنامه‌های سیار و بینایی سیستم‌های تعبیه شده مناسب باشد. روش پیشنهادی از نرمال‌سازی دسته‌ای، فعال‌سازی ReLU و یک لایه softmax برای محاسبه احتمال اطمینان ورودی به عنوان علامت ترافیک استفاده می‌کند. این مدل بر روی مجموعه داده‌های تشخیص علائم ترافیکی آلمان (GTSRB) با استفاده از بهینه‌ساز Adam با نرخ یادگیری ۰,۰۰۱ آموزش داده شد و آموزش مدل برای ۳۰ دوره بدون افزایش داده و ۲۰۰ دوره دیگر با تکنیک‌های افزایش داده‌ها مانند چرخش، مقیاس‌گذاری، تغییر و تبدیل انجام شد. نتیجه نهایی دقت، ۹۹,۶۶ درصد بود که نشان دهنده اثربخشی روش پیشنهادی است. در [۳] دقت تشخیص علائم ترافیکی با استفاده از دو روش یادگیری عمیق YOLOv5 و SSD مقایسه شدند. YOLOv5 یک الگوریتم تشخیص شی بلادرنگ است که کل تصویر را با یک شبکه عصبی پردازش می‌کند و آن را به بخش‌هایی تقسیم می‌کند تا جعبه‌های مرزی و احتمالات هر قسمت را تخمین بزند. از سوی دیگر، SSD با حذف نیاز به شبکه‌های پیشنهادی منطقه برای هر جزء، روند را تسریع می‌کند. نویسندگان مجموعه داده‌ای از ۲۱۸۲ تصویر علائم ترافیکی را از ۸ کلاس مختلف جمع‌آوری کردند که به صورت ۶۴٪ مجموعه آموزشی، ۱۶٪ مجموعه اعتبار سنجی و ۲۰٪ مجموعه آزمون تقسیم کردند. مدل‌ها با استفاده از تکنیک‌های افزایش داده‌ها، مانند چرخش و تغییر اندازه آموزش داده شدند. برای YOLOv5، اندازه تصویر روی 640×640 تنظیم شد، اندازه دسته‌ها ۱۶ بود، و مدل برای ۲۰۰ دوره آموزش داده شد. در [۴] یک راه حل مبتنی بر شبکه عصبی کانولوشنی (CNN) برای تشخیص علائم ترافیکی پیشنهاد کردند. معماری شبکه عصبی کانولوشنی پیشنهادی شامل لایه‌های کانولوشن، کاملاً متصل و softmax است. نویسندگان، آزمایش‌هایی را با پیکربندی‌های مختلف معماری شبکه عصبی کانولوشنی انجام دادند که بهترین آنها شامل ۳ لایه کانولوشن، ۱ لایه کاملاً متصل و یک لایه softmax بود. دقت روش پیشنهادی بر روی مجموعه داده GTSRB، به ۹۹,۹۴ درصد دست یافت. در [۵] یک سیستم تشخیص علائم ترافیکی را با استفاده از شبکه عصبی کانولوشن LeNet پیشنهاد کردند که ترکیبی از دو مدل از پیش آموزش دیده، VGG16 و LeNet است. این سیستم بر روی مجموعه داده GTSRB آموزش داده شد و به یک مجموعه آموزشی ۳۹۲۰۹ و یک مجموعه آزمایشی ۱۲۶۳۰ تقسیم شد. این سیستم با استفاده از بهینه‌ساز Adam با تابع softmax آموزش داده شد. این آزمایش نشان داد که مدل LeNet بهتر از مدل VGG16 با سطوح دقت ۹۸,۶ درصد و دقت ۹۶,۷ درصد عمل کرد. در [۶] یک مدل LeNet-5 پیشرفته برای طبقه بندی علائم ترافیکی پیشنهاد کردند. مدل پیشنهادی شامل دو لایه کانولوشن برای استخراج ویژگی‌ها از تصاویر، به دنبال آن دو لایه کانولوشنی و یک لایه کاملاً متصل بود. برای افزایش پایداری و سرعت آموزش مدل، نرمال‌سازی دسته‌ای و حذف با نرخ ۰,۵ بعد از لایه کاملاً متصل اضافه شد. LeNet-5 بهبود یافته بر روی مجموعه داده‌های GTSRB و

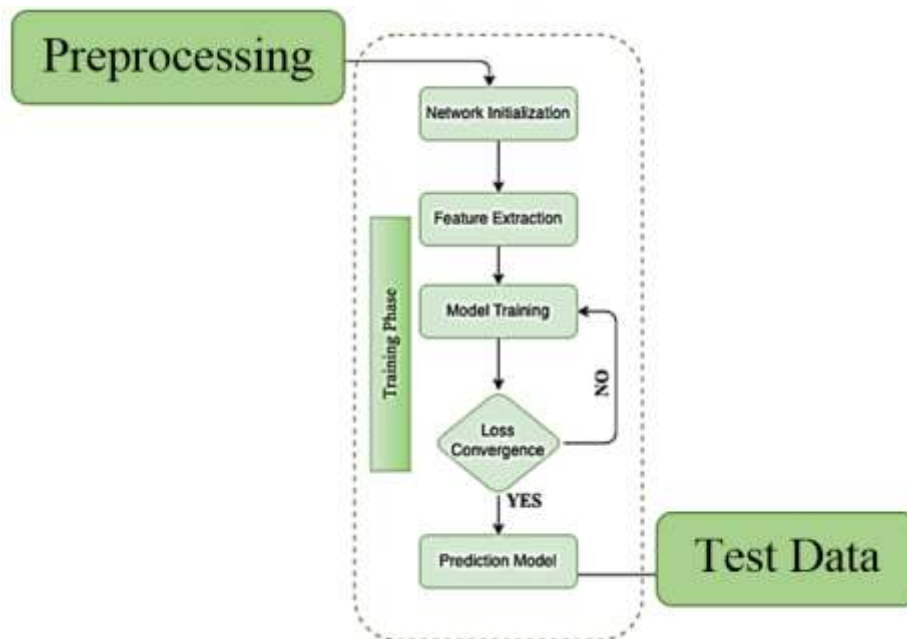
BTSD آموزش داده شد و تصاویر با استفاده از متعادل سازی هیستوگرام، تبدیل مقیاس خاکستری، تغییر اندازه و نرمال سازی، پیش پردازش شدند. این آزمایش، نشان داد که بهینه ساز Adam با فعال سازی ReLU بهتر از Adadelta با فعال سازی LeakyReLU عمل می کند و مدل LeNet-5 به دقت ۹۹،۸۴ درصد در مجموعه داده GTSRB و ۹۸،۳۷ درصد در مجموعه داده BTSD دست یافت. در [۷]، چندین مدل شبکه عصبی کانولوشنی (CNN) برای عملکرد آنها در طبقه بندی علائم ترافیکی ارزیابی شدند. سه مدل ارزیابی شده VGG16، ResNet50 و CNN پیشنهادی نویسندگان بر اساس AlexNet بودند. مدل شبکه عصبی کانولوشنی پیشنهادی با اندازه فیلتر کاهش یافته و اندازه گام ۱ طراحی شد. همچنین از Maxpool با اندازه گام ۲ برای نمونه برداری از نقشه ویژگی، یک لایه مسطح و سه لایه کاملاً متصل استفاده می کند. در آخرین لایه کاملاً متصل، از ۴۳ نورون برای نشان دادن هر یک از ۴۳ کلاس مختلف علائم ترافیکی، همراه با یک تابع فعال سازی softmax برای طبقه بندی استفاده می شود. فعال سازی ReLU بعد از هر لایه کانولوشنی برای جلوگیری از میانگین به صفر شدن ویژگی های پیچیده استفاده می شود. مجموعه داده مورد استفاده، معیار تشخیص علائم ترافیک آلمان (GTSRB) بود. مجموعه آموزشی برای کاهش شدت و کاهش هزینه محاسباتی به مقیاس خاکستری تبدیل شد و به دنبال آن یکسان سازی هیستوگرام برای کاهش کنتراست برای اطمینان از توزیع یکنواخت شدت پیکسل ها انجام شد. نتایج نشان داد که VGG16 به دقت ۹۵،۵ درصد، ResNet50 به دقت ۹۵،۴ درصد دست یافت و شبکه عصبی کانولوشنی پیشنهادی با AlexNet به بالاترین دقت ۹۶،۰ درصد دست یافت. در [۸]، یک روش طبقه بندی و تشخیص علائم ترافیکی را با شبکه عصبی کانولوشنی عمیق پیشنهاد کردند. در معماری مدل شبکه عصبی کانولوشنی پیشنهادی، لایه های کانولوشنی، یک لایه ادغام و یک لایه جمع حداکثری وجود دارد. برای آزمایش از سه مجموعه داده شامل GTSRB، BTSC و TSRD+GTSRB به ترتیب با ۴۳، ۶۲ و ۱۰۱ کلاس استفاده شد. به منظور جلوگیری از برازش بیش از حد و برای افزایش تعمیم پذیری آموزش، از تکنیک های تقویت داده ها مانند چرخش، بزرگ نمایی و مقیاس استفاده شد. با روش پیشنهادی شبکه عصبی کانولوشنی، GTSRB به دقت ۹۹،۷۶ درصد، BTSC به دقت ۹۹،۷۹ درصد و TSRD+GTSRB به دقت ۹۸،۳۷ درصد دست یافت. در [۹]، یک رویکرد با دو شبکه عصبی کانولوشنی برای طبقه بندی علائم ترافیکی استفاده شد. مدل MCNN با استفاده از مجموعه داده اصلی، آموزش داده شد، در حالی که مدل CCNN با نمونه های افزوده شده، آموزش داده شد. نتایج هر دو مدل با محاسبه احتمالات سوپرکلاس و زیر کلاس علائم ترافیکی و انتخاب برچسب تشخیص واقعی بر اساس وزن احتمال بالاتر مقایسه شد. مجموعه داده مورد استفاده مجموعه داده GTSRB بود و اندازه تصاویر به ۳۲×۳۲ تغییر داده شد و برای افزایش حجم نمونه مجموعه داده، از تکنیک های تقویت داده ها مانند مقیاس بندی، چرخش و چرخش استفاده شد. ۴۳ کلاس در مجموعه داده GTSRB به پنج ابر کلاس، از جمله علائم منع دایره ای قرمز، علائم خطر مثلثی قرمز، علائم اجباری دایره ای آبی، علائم ممنوعیت دایره ای سیاه و سایر علائم گروه بندی شدند. آزمایش ها برای ۱۰۰ دوره با اندازه دسته ای ۳۰ اجرا شد. نتایج نشان داد که CCNN با افزایش داده ها به سطح دقت بالاتری (۹۸،۲۶٪) در مقایسه با مدل MCNN (۹۷،۹۶٪) دست یافت. در [۱۰]، یک معماری سبک وزن شبکه عصبی کانولوشنی به نام DeepThin را معرفی کردند که چندین لایه کانولوشنی با اندازه هسته متوسط را روی هم قرار می دهد. مجموعه داده های مورد استفاده GTSRB و مجموعه داده های طبقه بندی علائم ترافیکی بلژیکی (BTSC) با اندازه تصاویر به ۴۵×۴۵ پیکسل و تبدیل به مقیاس خاکستری بودند. این روش دقت ۹۹،۷۲ درصد برای GTSRB و ۹۹،۲۹ درصد برای BTSC پس از ترکیب تکنیک هایی مانند یادگیری گروهی و تنظیم دقیق را به دست آورد. در [۱۱] روشی را برای طبقه بندی علائم ترافیکی با استفاده از MicronNet ارائه کردند. MicronNet یک شبکه عصبی عمیق کوچک است که برای استفاده در دستگاه های تعبیه شده طراحی شده است و MicronNet-BF دقت آن را با ادغام آن با نرمال سازی دسته ای بهبود بخشیده است. مجموعه داده GTSRB استفاده شد و MicronNet-BF با زمان پردازش فقط ۱،۴۱ ثانیه به دقت ۹۹،۳۸ درصد دست یافت. گنجاندن نرمال سازی دسته ای دقت را ۱،۰۵٪ در مقایسه با MicronNet اصلی بهبود بخشید. در [۱۲] با استفاده از چندین شبکه عصبی ResNets، از جمله ResNet18، ResNet50 و ResNet152، یک رویکرد جدید برای تشخیص علائم ترافیکی معرفی کردند. این مدل ها به ترتیب دارای ۱۸، ۵۰ و ۱۵۲ لایه پنهان وزن دار بودند. مجموعه های با بیش از ۴۰۰۰۰ تصویر را با استفاده از یک ماشین خودران جمع آوری کردند، از ۲۲۴ تصویر با وضوح بالا برای آزمایش استفاده کردند. مجموعه داده شامل سه کلاس که هر کدام شامل ۳۰۰ تصویر است. اندازه تصاویر به ۲۲۴×۲۲۴ پیکسل تغییر یافت و به شرح زیر تقسیم شد: ۸۰٪ آموزش، ۲۰٪ اعتبار سنجی. از پانزده تصویر برای آزمایش استفاده شد. هیچ افزایش داده ای برای مجموعه داده اعمال نشد. این مدل برای ۱۰ دوره با اندازه دسته ای ۱۰ آموزش داده شد و آخرین لایه کاملاً متصل به خوبی تنظیم شد. بالاترین دقت تست ۹۳٪ برای ResNet50، ۶۰٪ برای ResNet18 و ۳۳٪ برای ResNet152 بود. در [۱۳] یک معماری شبکه عصبی

کانولوشنی بهبود یافته LeNet-5 را برای طبقه‌بندی علائم ترافیکی پیشنهاد کردند. معماری LeNet-5 شامل ۲ لایه کانولوشنی، ۲ لایه ادغام، ۲ لایه کاملاً متصل و یک لایه طبقه‌بندی بود. برای بهبود عملکرد مدل، یک هسته گابور به عنوان اولین هسته کانولوشنی استفاده شد و نرمال‌سازی دسته‌ای پس از هر لایه ادغام انجام شد. تابع فعال‌سازی ReLU برای رسیدگی به مسائل ناپایداری گرادینان استفاده شد و نرخ حذف ۰٫۵ در لایه‌های کاملاً متصل اعمال شد. از مجموعه داده GTSRB استفاده کردند که ۷۵٪ برای آموزش و ۲۵٪ برای آزمایش بود و این مدل به دقت ۹۹٫۷۵ درصد دست یافت. در [۱۴] یک شبکه کانولوشنی چند مقیاسی (MSCN) و یک شبکه عصبی عمیق با رویکرد چند ستونی (MCDNN) را برای تشخیص علائم ترافیکی پیشنهاد کردند. مجموعه داده TSRD برای آموزش استفاده شد، در حالی که مجموعه داده GTSRB برای آزمایش استفاده شد. برای بهبود عملکرد مدل، افزایش داده و تنظیم دقیق را انجام دادند. با این حال، نتایج نشان داد که دقت در ابتدا افزایش سپس با افزایش تعداد کلاس‌ها کاهش می‌یابد. روش پیشنهادی به دقت ۹۰٫۱۳ درصد دست یافت. در [۱۵]، مطالعه‌ای برای بررسی تأثیر ابعاد مختلف فیلترهای لایه کانولوشنی بر عملکرد یک شبکه عصبی کانولوشنی برای طبقه‌بندی علائم ترافیکی انجام شد. ابعاد در نظر گرفته شده در آزمایش ۳، ۵، ۹، ۱۳، ۱۵، ۱۹، ۲۳، ۲۵ و ۳۱ بود. از مجموعه داده GTSRB برای آزمایش‌ها استفاده شد و تصاویر موجود در مجموعه داده با نرمال‌سازی و تغییر اندازه به 32×32 پیکسل پیش پردازش شدند. تابع خطا برای بهینه‌سازی شبکه به کار گرفته شد و ۱ گام برای لایه کانولوشنی و ۲ گام برای لایه ادغام، استفاده شد. نتایج نشان داد که فیلترهای لایه کانولوشنی با ابعاد 9×9 و 19×19 به ترتیب بالاترین سطوح دقت ۸۶٫۴ درصد و ۸۶٫۸ درصد را با سرعت طبقه‌بندی سریع ۰٫۰۰۴۴۷۲ و ۰٫۰۰۲۷۸۶ ثانیه داشتند. در [۱۶] یک روش مبتنی بر شبکه عصبی کانولوشنی برای طبقه‌بندی علائم ترافیکی پیشنهاد شد. معماری شبکه عصبی کانولوشنی پیشنهادی دارای ۱۲ لایه، شامل ۴ لایه کانولوشنی، ۲ لایه ادغام حداکثری، ۴ لایه حذفی، ۱ لایه مسطح و ۱ لایه کاملاً متصل است. تابع خطای متقاطع آنتروپی برای بهینه‌سازی شبکه استفاده شد و آزمایش‌ها بر روی مجموعه داده GTSRB انجام شد. اندازه تصاویر به 30×30 تغییر داده شد و از سایز دسته‌ای ۳۰ استفاده شد و روش پیشنهادی به دقت ۹۹٫۶۶ درصد دست یافت. به طور مشابه، در [۱۷] از شبکه عصبی کانولوشنی برای طبقه‌بندی علائم ترافیکی استفاده شد. مدل متشکل از ۴ لایه کانولوشنی، ۲ لایه ادغام حداکثری، یک لایه حذفی، یک لایه مسطح و ۲ لایه متراکم بود. اندازه فیلتر کانولوشنی 3×3 در معماری اعمال شد و تابع فعال‌سازی ReLU در لایه‌های مخفی مختلف به کار گرفته شد. همچنین از تابع خطای متقاطع آنتروپی، بهینه‌ساز آدام و نرخ یادگیری ۰٫۰۰۱ استفاده شد. مجموعه داده GTSRB و تکنیک‌های افزایش داده‌ها نیز استفاده شد. معماری شبکه عصبی کانولوشنی پیشنهادی به دقت ۹۹٫۲۰ درصد با سرعت طبقه‌بندی ۶٫۶۳ ثانیه برای داده‌های آزمایشی دست یافت. در [۱۸] سه مدل شبکه عصبی کانولوشنی را مقایسه کردند: AlexNet، DarkNet-53، و EfficientNet-b0. معماری AlexNet از ۱۳ لایه شامل ۵ لایه کانولوشنی، ۳ لایه ادغام حداکثری، ۲ لایه حذفی و ۳ لایه کاملاً متصل تشکیل شده است. توابع فعال‌سازی مورد استفاده ReLU و Softmax بودند. معماری DarkNet-53 از ۵۳ لایه، عمدتاً 1×1 و 3×3 لایه کانولوشنی، یک لایه نرمال‌سازی دسته‌ای و یک لایه LeakyReLU بعد از هر لایه کانولوشنی تشکیل شده بود. EfficientNet-b0 شامل ۷ لایه MBConv، یک لایه کانولوشنی و یک لایه ادغام و کاملاً متصل بود. مجموعه داده GTSRB برای آزمایش استفاده شد و ابعاد تصویر ورودی برای هر الگوریتم متفاوت و AlexNet 227×227 ، DarkNet-53 416×416 و EfficientNet-b0 224×224 بود. AlexNet با ۹۷٫۴۵٪ و DarkNet-53 با ۹۴٫۶۹٪ و EfficientNet-b0 با ۹۸٫۶۴٪ بالاترین دقت را ارائه دادند. در [۱۹] یک روش طبقه‌بندی علائم ترافیکی با استفاده از شبکه عصبی عمیق (DNN) و شبکه عصبی کوانتومی (QNN) پیشنهاد شد. از مجموعه‌ای از N فیلتر کوانتومی برای ساختن لایه‌های کانولوشنی کوانتومی استفاده کردند. هر دو شبکه DNN و QNN دارای لایه‌هایی با ویژگی‌های کانولوشنی، ادغام حداکثری، حذف و لایه متراکم بودند. نتایج نشان داد که DNN به دقت ۹۹٫۸۶ درصد دست یافت، در حالی که QNN دقت ۹۴٫۴۰ درصد را به دست آورد. در [۲۰] همچنین یک روش طبقه‌بندی علائم ترافیکی با استفاده از شبکه عصبی کانولوشنی پیشنهاد شد. شبکه از ۳ لایه کاملاً متصل، ۳ لایه ادغام حداکثری و ۳ لایه کانولوشنی تشکیل شده است. این شبکه برای تبدیل یک تصویر 100×100 در مقیاس خاکستری به اندازه کوچکتر و طبقه‌بندی آن به یکی از ۴۱ کلاس علائم ترافیکی طراحی شده است. ReLU به عنوان تابع فعال‌سازی استفاده شد و شبکه برای ۱۰ دوره آموزش داده شد. از مجموعه داده علائم ترافیکی اندونزیایی استفاده شد که شامل ۳۱۳۳ تصویر ترافیکی و ۴۱ طبقه بندی بود. معماری پیشنهادی به دقت ۹۳ درصد دست یافت. در [۲۱] یک مدل طبقه بندی علائم ترافیکی با استفاده از شبکه عصبی کانولوشنی پیشنهاد شد. این مدل از ۱۱ لایه شامل چهار لایه کانولوشنی، دو لایه تجمیع، یک لایه مسطح و چهار لایه کاملاً متصل تشکیل شده است. این مدل با استفاده از مجموعه داده‌های GTSRB و یک مجموعه داده محلی از علائم ترافیکی هندی آموزش داده شد. یک مجموعه داده

ترکیبی از ترکیب دو مجموعه داده تشکیل شد که منجر به ۱۰۲ کلاس در کل و ۶۵۸۱۰ تصویر شد. مراحل پیش پردازش مانند مقیاس خاکستری، متعادل سازی هیستوگرام، و نرمال سازی، به مجموعه داده اعمال شد. مدل به سطوح دقت ۹۹,۸۵ درصد در مجموعه داده آلمانی، ۹۱,۰۸ درصد در مجموعه داده هندی و ۹۵,۴۵ درصد در مجموعه داده ترکیبی دست یافت.

معماری شبکه عصبی پیشنهادی

این بخش یک شبکه عصبی سبک وزن را برای تشخیص علائم ترافیکی ارائه می‌کند که به صحت و دقت بالایی با پارامترهای قابل آموزش کمتری دست می‌یابد. مدل پیشنهادی بر روی مجموعه داده‌های معیار تشخیص علائم ترافیکی آلمان (GTSRB) آموزش داده شده است. ابتدا تصاویر علائم ترافیکی جمع آوری و به عنوان مجموعه داده استفاده شد. در بلوک اول، تصاویر با استفاده از روش‌های مختلف برای بهبود شناسایی علائم ترافیکی پیش پردازش شدند. سپس، داده‌های از پیش پردازش شده به مرحله آموزش فرستاده شدند، جایی که مدل پیشنهادی بر روی مجموعه آموزشی با فرآیندهای بهینه شده آموزش داده شد. در نهایت، عملکرد مدل در مجموعه آزمایشی برای تعیین عملکرد کلی آن مورد ارزیابی قرار گرفت. شکل ۹ یک نمایش بصری از گردش کار و مراحل مختلف درگیر در این فرآیند را ارائه می‌کند که درک بهتر روش مورد استفاده در این مطالعه را تسهیل می‌کند.



شکل ۹: فلوچارت مراحل اجرایی یک معماری کانولوشنی

پیش پردازش

مرحله پیش پردازش داده‌ها شامل سه مرحله است: تغییر مقیاس تصویر، نرمال سازی و تقویت داده‌ها. مجموعه داده‌های مورد استفاده در این مقاله GTSRB شامل تصاویر با نسبت ابعادی بین 15×15 تا 250×250 پیکسل است. برای سازگاری با شبکه‌های عصبی، داشتن اندازه‌های ثابت تصویر ضروری است. علاوه بر این، ذکر این نکته ضروری است که کاهش اندازه تصویر به نسبت پیکسل کمتر، مانند 80×80 یا 50×50 ، پیچیدگی مدل را کاهش می‌دهد. با این حال، ممکن است بر توانایی مدل برای نمایش دقیق اطلاعات بصری تأثیر منفی بگذارد و عملکرد طبقه‌بندی را کاهش دهد. در آزمایش‌های انجام شده، مدل خود را با اندازه‌های تصویر مختلف آزمایش کردیم و دریافتیم که 100×100 پیکسل، مبادله بهینه بین پیچیدگی محاسباتی و دقت طبقه‌بندی را فراهم می‌کند. نرمال سازی داده‌ها برای اطمینان از توزیع ثابت پارامترهای ورودی (مقادیر پیکسل)، که امکان همگرایی سریع‌تر در طول آموزش شبکه را فراهم می‌کند، حیاتی است. برای دستیابی به این هدف، مقدار میانگین

کم می‌شود و نتیجه بر انحراف معیار آن تقسیم می‌شود و در نتیجه یک توزیع گاوسی با مقدار صفر ایجاد می‌شود. افزایش داده‌ها گام مهمی برای رفع نبود تعادل در مجموعه داده است، جایی که برخی از برجسبها در مقایسه با سایر برجسبها دارای تعداد زیادی تصویر هستند. این روش داده‌های اضافی را از نمونه‌های موجود با اعمال تبدیل‌هایی مانند چرخش، بزرگنمایی، تبدیل و تنظیمات روشنایی ایجاد می‌کند. تابع ImageDataGenerator از کتابخانه Keras، امکان می‌دهد پارامترهایی مانند ضریب مقیاس، دامنه تغییر ارتفاع و عرض، محدوده چرخش، چرخش‌های افقی و عمودی و موارد دیگر را تنظیم کنیم تا داده‌های جدید تولید کنیم که می‌تواند آموزش مدل را بهبود بخشد. افزایش به‌کارگیری یادگیری عمیق جامعه تحقیقاتی را بر آن داشته‌است تا برای دقت بهتر در کارهایی مانند طبقه‌بندی تصاویر تلاش کنند. در نتیجه، شاهد ابداع معماری‌های عمیق‌تری بوده‌ایم که دارای ۲۳.۸ میلیون پارامتر و دارای ۱۴۳.۶ میلیون پارامتر هستند. با این حال، افزایش عمق همیشه به سادگی منجر به بهبود دقت نمی‌شود. در واقع، استفاده از هسته‌های بیش از حد می‌تواند باعث کاهش عملکرد بدون کاهش مسئله بایاس بالا و واریانس بالا شود. روند تکراری یادگیری ویژگی‌های جدید بر اساس ویژگی‌هایی که قبلاً آموخته‌اند، همیشه بهینه نیست. در پاسخ به چالش تعادل عمق و دقت، یک معماری شبکه عصبی سبک وزن برای تشخیص علائم ترافیکی پیشنهاد می‌شود که ضمن استفاده از پارامترهای آموزش پذیر کمتر، به دقت و صحت بالاتری دست یابد.

لایه‌های شبکه عصبی کانولوشنی با تبدیل اطلاعات خام از تصاویر ورودی به یک تانسور، استخراج ویژگی‌ها را انجام می‌دهند. این فرآیند مدل را قادر می‌سازد تا الگوها و ویژگی‌های مهم را در تصویر شناسایی کند. سپس تانسور با استفاده از لایه ادغام میانگین جهانی به یک دسته علائم ترافیکی خاص طبقه بندی می‌شود. علاوه بر این، پارامترهای متغیر همه این لایه‌ها با به حداقل رساندن خطای طبقه‌بندی در مجموعه آموزشی بهینه می‌شوند. لایه‌های کانولوشنی اجزای اساسی شبکه‌های عصبی هستند که هر یک از نقشه‌های ورودی را با یک فیلتر دوبعدی کانوالو می‌کنند، به طوری که x و y ابعاد ورودی را نشان می‌دهند. هر لایه کانولوشنی شامل نورون‌هایی با بایاس و وزن‌های قابل یادگیری است که به لایه اجازه می‌دهد در طول زمان یاد بگیرد و سازگار شود. نقشه خروجی لایه n با جمع کردن پاسخ‌های کانولوشنی $n-1$ لایه ایجاد می‌شود. وزن‌های w فیلتری را نشان می‌دهند که نقشه ورودی و نقشه‌های خروجی را به هم متصل می‌کند، در حالی که β نشان‌دهنده بایاس نقشه خروجی است. هدف اصلی لایه‌های کانولوشنی استخراج و شناسایی ویژگی‌های خاص از نقشه‌های ورودی و ایجاد انتزاعات سطح بالاتر از این ویژگی‌ها با ترکیب فعال‌سازی فیلتر با استفاده از معادله زیر است.

$$\mathbb{R}(z) = \sum_{i=1}^{n-1} \alpha_i^{n-1} * \omega_{ij}^n + \beta^n$$

در روش پیشنهادی، از تابع فعال‌سازی واحد خطی اصلاح‌شده (ReLU) به دلیل غیر خطی بودن، کارایی محاسباتی و توانایی مدیریت ورودی‌های غیرمنفی مانند مقادیر پیکسل استفاده می‌کنیم. معادلات زیر نمایش ریاضی تابع فعال‌سازی ReLU را با شرایط نشان می‌دهد. با صفر کردن مقادیر منفی، تابع فعال‌سازی ReLU به جلوگیری از ناپدید شدن گرادینانها و بهبود عملکرد مدل کمک می‌کند.

$$\mathbb{R}(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

$$\mathbb{R}(z) = \max(0, z)$$

لایه‌های ادغام حداکثری با کاهش اندازه فضایی نقشه‌های ویژگی، نقش مهمی در شبکه عصبی کانولوشنی ایفا می‌کنند و در نتیجه تعداد پارامترها و هزینه‌های محاسباتی را کاهش می‌دهند. علاوه بر آن، می‌توانند با انتخاب ویژگی‌های ثابت برتر-که توانایی تعمیم مدل را بهبود می‌بخشد، به جلوگیری از برازش بیش از حد کمک کنند. حداکثر فعال‌سازی روی مناطق غیرهمپوشان، خروجی لایه ادغام حداکثری را تعیین می‌کند. در طول این فرآیند، نقشه ورودی نمونه‌برداری و بعد عمق حفظ می‌شود. برای شناسایی خودکار علائم و سیستم‌های کمک رانندگی در وسایل نقلیه خودران که در آن توانایی تشخیص و طبقه‌بندی دقیق علائم ترافیکی برای ایمنی جاده بسیار مهم است، نرمال‌سازی دسته‌ای

می‌تواند بسیار سودمند باشد. فعال‌سازی هر لایه را در یک سری کوچک از نمونه‌های آموزشی نرمال‌سازی می‌کند، در نتیجه وابستگی به تغییرات داخلی را کاهش می‌دهد و به مدل پیشنهادی اجازه می‌دهد تا به طور مؤثرتری یاد بگیرد. علاوه بر این، با بهبود پایداری و کارایی آموزش، مدل پیشنهادی می‌تواند به همگرایی سریع‌تر و عملکرد تعمیم بهتر منجر شود. در چهارچوب پیشنهادی، ادغام میانگین سراسری را قبل از لایه‌های مسطح و متراکم قراردادیم تا ضمن بهبود عملکرد تعمیم، هزینه‌های اضافی و محاسباتی را کاهش دهیم. این تکنیک، مقدار میانگین هر نقشه ویژگی را محاسبه، یک بردار ویژگی واحد برای کل تصویر برای طبقه‌بندی ایجاد و از بیش از حد برازش جلوگیری می‌کند و در این حال معماری را ساده می‌کند و پارامترهای قابل آموزش را کاهش می‌دهد. استفاده از ادغام میانگین سراسری، بهبود عملکرد کلی مدل و کاهش هزینه‌های محاسباتی را در برنامه‌های مختلف بینایی ماشین نشان داده‌است. چهارچوب پیشنهادی از طریق یک فرآیند تکراری که هدف آن متعادل کردن پیچیدگی محاسباتی و عملکرد مدل و به حداقل رساندن خطای طبقه‌بندی در مجموعه آموزشی است، بهینه می‌شود. برای انتخاب مقادیر بهینه برای هایپرپارامترها، مانند اندازه هسته و اندازه خروجی، یک جست‌وجو در طیفی از مقادیر انجام و مدل را برای هر ترکیبی از هایپرپارامترها آموزش دادیم و ارزیابی کردیم و مجموعه‌ای را انتخاب کردیم که بالاترین دقت را در مجموعه اعتبار سنجی از یک تکنیک جست‌وجو با اعتبارسنجی متقابل برای ارزیابی عملکرد هر ترکیبی از هایپرپارامترها و انتخاب پیکربندی بهینه دارد. با این حال، انتخاب تعداد و اندازه لایه‌های کانولوشنی مشکل است که می‌تواند به طور قابل توجهی بر عملکرد یک معماری شبکه عصبی کانولوشنی تأثیر بگذارد. پیکربندی معماری پیشنهادی از جمله توابع فعال‌سازی، اندازه‌های هسته، گام‌ها و شکل‌های خروجی هر لایه در شکل ۱۰ نشان داده شده‌است.

Layer Name	Output Size	Kernel Size
Input	$100 \times 100 \times 3$	-
Conv2d	$98 \times 98 \times 16$	3×3
Conv2d	$98 \times 98 \times 32$	3×3
MaxPooling2D	$49 \times 49 \times 32$	2×2
Conv2d	$49 \times 49 \times 32$	3×3
Conv2d	$49 \times 49 \times 64$	5×5
Conv2d	$49 \times 49 \times 64$	3×3
MaxPooling2D	$24 \times 24 \times 64$	2×2
Batch Normalization	$24 \times 24 \times 64$	-
Conv2d	$24 \times 24 \times 64$	3×3
Conv2d	$24 \times 24 \times 128$	5×5
Conv2d	$24 \times 24 \times 256$	3×3
Global Average Pooling 2D	$24 \times 24 \times 128$	-
Flatten	256	-
Dense1	1024	-
Dense2	Number of Classes	-

شکل ۱۰: پیکربندی معماری شبکه پیشنهادی

به طور خلاصه، روش پیشنهادی شامل مراحل مختلفی از جمله پیش پردازش داده‌ها، طراحی معماری مدل و آموزش و ارزیابی است. با استفاده از تابع فعال‌سازی ReLU، نرمال‌سازی دسته‌ای و فرآیندهای مناسب، دقت بهبود یافته در تشخیص علائم ترافیکی به دست آمده‌است.

تابع خطا و الگوریتم بهینه‌سازی

از بهینه‌ساز آدام و تابع خطای متقاطع آنتروپی در طول آموزش استفاده می‌کنیم. بهینه‌ساز آدام یک الگوریتم بهینه‌سازی پرکاربرد است که از نرخ‌های یادگیری تطبیقی برای همگرایی مؤثرتر به حداقل سراسری نسبت به الگوریتم شیب نزولی تصادفی استفاده می‌کند. از سوی دیگر،

آنتروپی متقاطع به عنوان تابع ضرر برای مقایسه توزیع‌های پیش‌بینی‌شده و واقعی مسئله طبقه‌بندی انتخاب می‌شود. این تابع، احتمال ۱ را به کلاس واقعی و احتمال ۰ را به کلاس‌های دیگر اختصاص می‌دهد. مدل پیشنهادی شامل یک طبقه‌بندی‌کننده softmax در آخرین لایه است که از تابع خطای آنتروپی متقاطع استفاده می‌کند، همانطور که در رابطه زیر نشان داده شده است. در این معادله، k نشان‌دهنده کلاس‌های مختلف، \hat{p}_k توزیع احتمال پیش‌بینی‌شده، و p_k توزیع واقعی است که به‌عنوان یک بردار One-hot نشان داده می‌شود.

$$\mathcal{L}(\hat{p}, p) = \sum_k [\hat{p}_k \log(\hat{p}_k) + (1 - \hat{p}_k) \log(1 - \hat{p}_k)]$$

همانطور که در معادله زیر نشان داده شده است، تابع σ Softmax برای محاسبه y اعمال می‌شود که یک بردار K بعدی از امتیازهای با ارزش واقعی دلخواه Z را می‌گیرد و آن را به یک بردار K بعدی $\sigma(z_k)$ از مقادیر نگاشت می‌کند. بین مقادیر ۰ و ۱، که جمع آنها ۱ می‌شود. در آزمایش‌های انجام شده، ترکیب بهینه‌ساز آدام و تابع خطای متقاطع آنتروپی، نتایج مطلوبی را در افزایش دقت مدل نشان می‌دهد.

$$\sigma(\vec{z}_k) = \frac{e^{z_k}}{\sum_{i=1}^N (e^{z_i})} \quad \therefore k = 1, 2, \dots, N$$

مجموعه داده

مجموعه داده‌های معیار تشخیص علائم ترافیک آلمان (GTSRB) شامل ۵۱۹۲۲ تصویر است که در قالب PNG ذخیره شده‌اند. ابعاد تصاویر در این مجموعه داده از 15×15 پیکسل تا 250×250 پیکسل است. مجموعه داده GTSRB برای آموزش، اعتبارسنجی و آزمایش مدل به سه گروه تقسیم شده است: ۳۱۴۳۳ تصویر برای آموزش، ۷۸۵۹ تصویر برای اعتبارسنجی و ۱۲۶۳۰ تصویر برای آزمایش. این مجموعه داده شامل ۴۳ دسته متمایز (شکل ۱۱) است. لذا معیار تشخیص علائم ترافیکی آلمان، یک مجموعه داده از علائم ترافیکی است که به طور گسترده در زمینه تشخیص و شناسایی علائم ترافیکی استفاده می‌شود. GTSRB یک منبع جامع و قابل اعتماد برای ارزیابی عملکرد الگوریتم‌های تشخیص علائم ترافیکی ارائه می‌دهد. با این حال، توجه به این نکته مهم است که مجموعه داده GTSRB در درجه اول از علائم ترافیکی آلمانی تشکیل شده است که ممکن است به طور دقیق تنوع علائم ترافیکی مورد استفاده در مناطق دیگر را نشان ندهند. این امر تعمیم مدل‌های آموزش‌دیده شده بر روی مجموعه داده GTSRB را محدود می‌کند و ممکن است منجر به کاهش عملکرد هنگام اعمال در مناطق دیگر شود. با وجود این محدودیت، مجموعه داده GTSRB به دلیل اندازه، حاشیه‌نویسی با کیفیت بالا و نمایش سناریوهای واقعی، منبعی عالی را برای محققان در زمینه تشخیص علائم ترافیکی تبدیل می‌کند.

جزئیات پیاده‌سازی

هدف، دستیابی به دقت بالا، به حداقل رساندن زمان آموزش و استفاده از منابع است. علاوه بر این، مدل پیشنهادی برای استقرار در دنیای واقعی طراحی شده است، به طوری که دقت و قابلیت اطمینان در اولویت هستند. استفاده از تصاویر با وضوح بالاتر منجر به تشخیص و طبقه‌بندی دقیقتر علائم ترافیکی می‌شود که در تضمین ایمنی رانندگان و عابران پیاده بسیار مهم است. در طول فرآیند آموزش، از بهینه‌ساز آدام با نرخ یادگیری 0.00025 و اندازه دسته‌ای ۱۲۸ استفاده کردیم. علاوه بر این، تکنیک‌های مختلف افزایش داده‌ها را برای گسترش اندازه مجموعه آموزشی و بهبود توانایی تعمیم مدل بر روی Jupyter که مجهز به GPU GeForce RTX 1080 Ti بود انجام و با استفاده از پایتون ۳.۸ پیاده‌سازی شده است. مدل را با اندازه‌های تصویر مختلف آزمایش کردیم تا بهترین تعادل بین پیچیدگی محاسباتی (زمان و منابع) و عملکرد طبقه‌بندی را پیدا کنیم. وضوح تصویر 100×100 تعادل خوبی بین دقت طبقه‌بندی و پیچیدگی مدل ایجاد می‌کند. یافته‌ها نشان می‌دهد که وضوح پایین‌تر تأثیر منفی بر دقت مدل در طبقه‌بندی علائم ترافیکی دارد، در حالی که وضوح بالاتر منجر به افزایش پیچیدگی مدل و زمان‌های آموزشی طولانی‌تر می‌شود. (همانطور که در جدول ۱ نشان داده شده است).



شکل ۱۱: مجموعه داده‌های معیار تشخیص علائم ترافیکی آلمان (GTSRB).

جدول ۱: اندازه‌های تصویر ورودی و تعادل بین پیچیدگی محاسباتی (زمان و حافظه) و عملکرد طبقه‌بندی

امتیاز F1	دقت	میزان حافظه مصرفی	زمان محاسبه	وضوح تصویر
97.1	96.9	18.3MB	935s	70X70
98.43	98.31	21.6MB	1217s	100X100
98.64	98.63	28.7MB	2184s	150X150

معیارهای ارزیابی

معمولاً از شاخص‌های کلیدی عملکرد مدل برای ارزیابی اثربخشی مدل‌های طبقه‌بندی استفاده می‌شود. معیار صحت پیش‌بینی‌های انجام شده توسط مدل را اندازه‌گیری می‌کند و -همانطور که در روابط زیر نشان داده شده است- به عنوان نسبت تعداد نمونه‌های طبقه‌بندی شده صحیح به تعداد کل نمونه‌ها محاسبه می‌شود. از سوی دیگر، معیار دقت، نسبت مثبت‌های واقعی را در بین تمام پیش‌بینی‌های مثبت انجام‌شده توسط مدل، اندازه‌گیری می‌کند. معیار یادآوری نسبت مثبت‌های واقعی را در بین تمام نمونه‌های مثبت واقعی در مجموعه داده اندازه‌گیری می‌کند. معیار امتیاز F1 میانگین هارمونیک دقت و یادآوری است که تعادلی بین دو معیار ایجاد می‌کند. از شاخص‌های کلیدی عملکرد فوق‌الذکر برای اندازه‌گیری اثربخشی مدل پیشنهادی استفاده می‌کنیم. این معیارها توانایی مدل را در طبقه‌بندی دقیق علائم ترافیکی ارزیابی می‌کند.

$$\mathbb{A} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

$$\mathbb{P} = \frac{T_P}{T_P + F_P}$$

$$\mathbb{R} = \frac{T_P}{T_P + F_N}$$

$$\mathbb{F1-score} = 2 \times \frac{\mathbb{P} \times \mathbb{R}}{\mathbb{P} + \mathbb{R}}$$

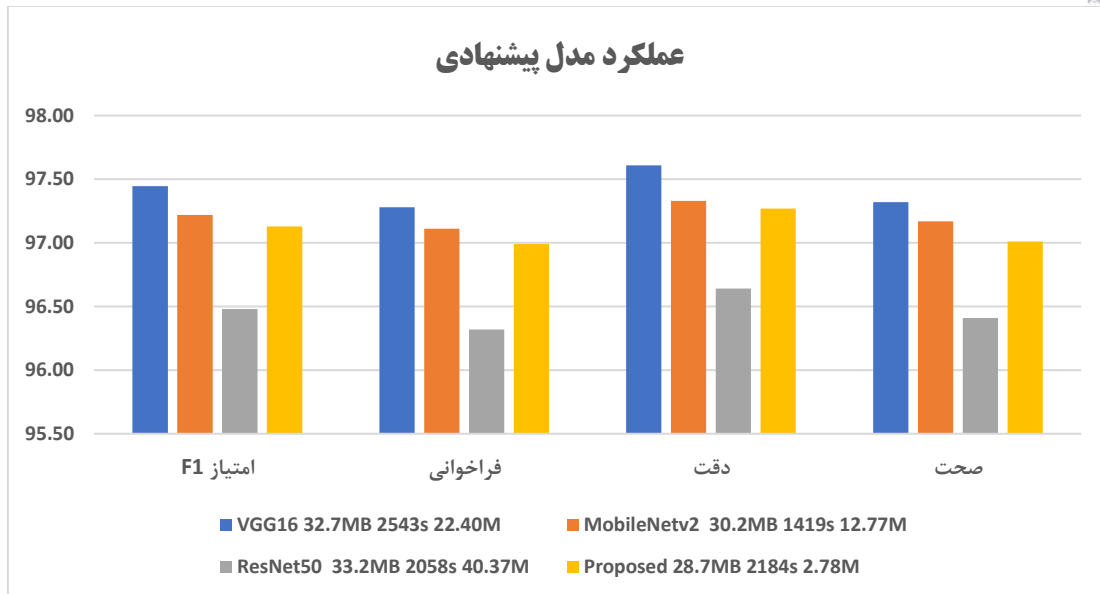
نتایج ارزیابی

برای ارزیابی کارایی و قابلیت اطمینان مدل پیشنهادی مبتنی بر شبکه عصبی کانولوشنی برای شناسایی علائم ترافیکی، آزمایش‌های گسترده‌ای را بر روی مجموعه داده GTSRB انجام دادیم. برای ایجاد یک معیار، نتایج را با چند تکنیک پیشرفته از جمله VGG16، MobileNetv2 و ResNet50 مقایسه کردیم. نتایج نشان داد که مدل پیشنهادی به عملکرد خوبی دست یافته است و بر پتانسیل آن برای استقرار مدل شناسایی علائم ترافیکی بلادرنگ و سیستم‌های کمک رانندگی تاکید می‌کند. مقایسه مدل پیشنهادی با سایر مدل‌های پیشرفته در مجموعه داده‌های GTSRB در جدول ۲ ارائه شده است. مدل پیشنهادی در مقایسه با VGG16، MobileNetv2 و ResNet50 به دقت خوبی دست یافت. لازم به ذکر است که راندمان محاسباتی و اندازه کوچک مدل پیشنهادی، آن را برای شناسایی علائم ترافیکی بلادرنگ کاربردی تر و مناسب تر می‌کند.

جدول ۲: مقایسه مدل پیشنهادی با سایر مدل‌های پیشرفته بر روی مجموعه داده GTSRB

مدل معماری	صحت	دقت	فراخوانی	امتیاز FI	تعداد پارامتر	زمان آموزش	حافظه مصرفی
VGG16	97.32	97.61	97.28	97.44	22.40M	2543s	32.7MB
MobileNetv2	97.17	97.33	97.11	97.22	12.77M	1419s	30.2MB
ResNet50	96.41	96.64	96.32	96.48	40.37M	2058s	33.2MB
روش پیشنهادی	97.01	97.27	96.99	97.13	2.78M	2184s	28.7MB

با مقایسه عملکرد مدل پیشنهادی و مدل‌های فوق‌الذکر، قابل مشاهده است که مدل پیشنهادی از نظر زمان اجرا و مصرف حافظه به ترتیب با مقادیر زمان اجرای ۲۱۸۴ ثانیه و حافظه مصرفی ۲۸,۷ مگابایتی قابل مقایسه با معماری‌های پیشرفته است و تعادل خوبی بین زمان اجرا و حافظه مصرفی به دست آمده است و آن را به یک راه‌حل جذاب برای استقرار در دنیای واقعی بر روی دستگاه‌های تعبیه شده با توان محاسباتی و حافظه محدود تبدیل می‌کند. در شکل ۱۲ بررسی عملکرد مدل پیشنهادی با سایر مدل‌های پیشرفته بر روی مجموعه داده GTSRB نشان داده شده است. لذا عملکرد کلی آن از نظر زمان آموزش، تعداد پارامترها و مصرف حافظه، آن را به یک راهکار مناسب برای استقرار بلادرنگ در سیستم‌های تعبیه شده تبدیل می‌کند. روش پیشنهادی به گونه‌ای طراحی شد که سبک و کارآمد باشد و از مهندسی ویژگی‌های پیشرفته و شبکه عصبی کانولوشنی برای بهبود دقت و استحکام کلی و در عین حال کاهش تعداد پارامترهای مورد نیاز برای آموزش و استنتاج استفاده کند.



شکل ۱۲: بررسی عملکرد مدل پیشنهادی با سایر مدل‌های پیشرفته بر روی مجموعه داده GTSRB

نتیجه‌گیری

مدل پیشنهادی در این مقاله به طور موثر، طیف متنوعی از علائم ترافیکی را با اشکال، اندازه‌ها و رنگ‌های مختلف شناسایی و طبقه‌بندی می‌کند که نشان‌دهنده استحکام و سازگاری آن در سناریوهای دنیای واقعی است. موفقیت این مدل در تشخیص و طبقه‌بندی انواع علائم ترافیکی، پتانسیل آن را برای افزایش اقدامات ایمنی در مدیریت ترافیک و سیستم‌های رانندگی خودمختار برجسته می‌کند. علاوه بر این، عملکرد مطلوب این مدل پیشنهادی در تشخیص و طبقه‌بندی علائم ترافیکی، پتانسیل کاربرد آن را در مناطق حیاتی ایمنی، مانند وسایل نقلیه خودران، که در آن تشخیص و طبقه‌بندی دقیق و سریع علائم ترافیکی ضروری است، نشان می‌دهد. هنگام طراحی مدل‌های یادگیری ماشین برای کاربردهای عملی مانند تشخیص علائم ترافیکی، توجه به کارایی محاسباتی و اندازه مدل بسیار مهم است. علاوه بر آن، در نظر گرفتن رابطه بین تعداد پارامترها/عملیات و زمان اجرا هنگام طراحی چنین مدل‌هایی نیز مهم است. به طور کلی، انتظار می‌رود مدلی با پارامترها و عملیات کمتر نسبت به مدل پیچیده‌تر، زمان پاسخگویی سریع‌تری داشته باشد؛ علت این است که یک مدل ساده‌تر به محاسبات کمتری نیاز دارد و پیش‌بینی‌های سریع‌تری را ممکن می‌سازد. در کاربردهای حیاتی ایمنی مانند سیستم‌های خودرو، طراحی چنین چهارچوب‌هایی که چهارچوب‌هایی دقیق و کارآمد با سربار محاسباتی کم و حداقل نیاز به حافظه هستند، ضروری است. این را می‌توان با کاهش تعداد پارامترها و عملیات قابل آموزش با حفظ دقت بالا به دست آورد و اطمینان حاصل کرد که مدل‌ها می‌توانند بر روی دستگاه‌های تعبیه شده با قدرت محاسباتی و حافظه محدود مستقر شوند که عملکردی قابل اعتماد و سریع را نیز ارائه می‌دهند. مدل پیشنهادی دارای اندازه مدل کوچک و راندمان محاسباتی بالایی است که آن را برای استقرار بلادرنگ کاربردی و مناسب می‌سازد. با افزایش تعداد پارامترها و عملیات در مدل‌های یادگیری ماشین، مدل، پیچیده‌تر می‌شود و به منابع محاسباتی بیشتری برای آموزش نیاز دارد. بنابراین، برای اطمینان از دقت بالا و زمان پاسخ کم، لازم است بین پیچیدگی و عملکرد، متعادل شود. هدف، توسعه یک سیستم سبک و کارآمد است که از تکنیک‌های یادگیری عمیق مانند شبکه عصبی کانولوشنی و مهندسی ویژگی‌های پیشرفته برای بهبود دقت و در عین حال کاهش تعداد پارامترهای مورد نیاز برای آموزش و استنتاج استفاده می‌کند. این سیستم در سناریوهای دنیای واقعی با منابع محاسباتی کم به طور موثر عمل خواهد کرد. آزمایش‌ها نشان می‌دهد که مدل پیشنهادی مبتنی بر شبکه عصبی کانولوشنی بسیار مؤثر و قابل اعتماد است و از نظر دقت با چندین تکنیک پیشرفته مانند VGG16، MobileNetV2 و ResNetV2 قابل مقایسه است. دقت، فراخوانی و امتیاز F1 در مجموعه داده GTSRB همانطور که در جدول ۲ نشان داده شده است، مدل پیشنهادی به نتایج تشخیص و طبقه‌بندی مطلوبی برای علائم ترافیکی با اشکال، اندازه‌ها و رنگ‌های مختلف دست یافت که استحکام و سازگاری آن را در شرایط دنیای واقعی

نشان می‌دهد. علاوه بر این، مدل پیشنهادی، سبک وزن و از نظر محاسباتی کارآمد است و آن را برای برنامه‌های بلادرنگ که استفاده از منابع، یک محدودیت است، مناسب می‌کند.

مراجع

- [1] L. Jiao et al., "A Survey of Deep Learning-Based Object Detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.
- [2] Li, J.; Wang, Z. Real-time traffic sign recognition based on efficient CNNs in the wild. *IEEE Trans. Intell. Transp. Syst.* 2018, 20, 975–984.
- [3] Zhu, Y.; Yan, W.Q. Traffic sign recognition based on deep learning. *Multimed. Tools Appl.* 2022, 81, 17779–17791.
- [4] Shustanov, A.; Yakimov, P. CNN design for real-time traffic sign recognition. *Procedia Eng.* 2017, 201, 718–725.
- [5] Bangquan, X.; Xiong, W.X. Real-time embedded traffic sign recognition using efficient convolutional neural network. *IEEE Access* 2019, 7, 53330–53346.
- [6] Zaibi, A.; Ladgham, A.; Sakly, A. A lightweight model for traffic sign classification based on enhanced LeNet-5 network. *J. Sensors* 2021, 2021, 8870529.
- [7] Sokiprialia, J.; Orike, S. Traffic sign classification comparison between various convolution neural network models. *Int. J. Sci. Eng. Res.* 2021, 12, 165–171.
- [8] Mishra, J.; Goyal, S. An effective automatic traffic sign classification and recognition deep convolutional networks. *Multimed. Tools Appl.* 2022, 81, 18915–18934.
- [9] Chen, L.; Zhao, G.; Zhou, J.; Kuang, L. Real-time traffic sign classification using combined convolutional neural networks. In *Proceedings of the 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, Nanjing, China, 26–29 November 2017; pp. 399–404.
- [10] Haque, W.A.; Arefin, S.; Shihavuddin, A.; Hasan, M.A. DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst. Appl.* 2021, 168, 114481.
- [11] Fang, H.F.; Cao, J.; Li, Z.Y. A small network MicronNet-BF of traffic sign classification. *Comput. Intell. Neurosci.* 2022, 2022, 3995209.
- [12] Sarku, E.; Steele, J.; Ruffin, T.; Gokaraju, B.; Karimodini, A. Reducing Data Costs-Transfer Learning Based Traffic Sign Classification Approach. In *Proceedings of the SoutheastCon 2021*, Atlanta, GA, USA, 10–13 March 2021; pp. 1–5.
- [13] Cao, J.; Song, C.; Peng, S.; Xiao, F.; Song, S. Improved traffic sign detection and recognition algorithm for intelligent vehicles. *Sensors* 2019, 19, 4021.
- [14] Fu, H.; Wang, H. Traffic Sign Classification Based on Prototypes. In *Proceedings of the 2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Chengdu, China, 26–28 November 2021; pp. 7–10.
- [15] Sichkar, V.; Kolyubin, S. Effect of various dimension convolutional layer filters on traffic sign classification accuracy. *Sci. Tech. J. Inf. Technol. Mech. Opt.* 2019, 19, 546–552.
- [16] Agarwal, S.; X, C.; Kumar, R. Convolutional Neural Network for Traffic Sign Classification. *Int. J. Inf. Technol. Proj. Manag.* 2022, 9.
- [17] Youssouf, N. Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4. *Heliyon* 2022, 8, e11792.
- [18] Gökberk, A.; Durdu, A.; Nesimioğlu, B.S. Accuracy Comparison of CNN Networks on GTSRB Dataset. *J. Artif. Intell. Data Sci.* 2022, 2, 63–68.
- [19] Kuros, S.; Kryjak, T. Traffic Sign Classification Using Deep and Quantum Neural Networks. 2022.
- [20] Pradana, A.I.; Rustad, S.; Shidik, G.F.; Santoso, H.A. Indonesian Traffic Signs Recognition Using Convolutional Neural Network. In *Proceedings of the 2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia, 17–18 September 2022; pp. 426–430.
- [21] Bhatt, N.; Laldas, P.; Lobo, V.B. A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN. In *Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES)*, India, June 2022; pp. 1354–1358.