

دسته‌بندی ابرنقاط سه‌بعدی با استفاده از یک معماری عمیق توسعه‌یافته

ایمان احمدی^۱، حسن ختن‌لو^۲ و معین پیغمبرزاده^۳

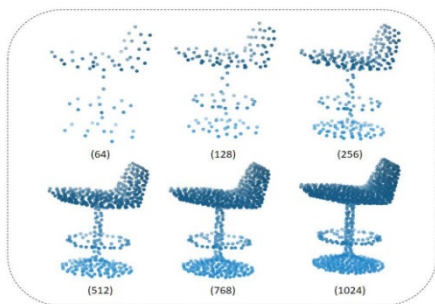
چکیده

با گذر زمان و بیشتر شدن ابزارهای دریافتگر داده‌های سه‌بعدی، میزان دسترسی به داده‌هایی همچون ابرنقاط افزایش یافت. ابرنقاط سه‌بعدی از جمله داده‌هایی است که اجسام را در فضای سه‌بعد بازنمایی می‌کند. این افزایش داده‌ها در کنار پیشرفت فراوان شبکه‌های عصبی عمیق امکان حل مسائل بسیاری را فراهم کرد. دسته‌بندی اجسام سه‌بعدی، قطعه‌بندی آن‌ها و تخمین جریان حرکتی اجسام از جمله مسائلی هستند که با فراهم آمدن شرایطی که پیشتر گفته شد، توجه پژوهشگران بسیاری را به خود جلب کرده‌اند. در این مقاله به منظور بهبود دقت دسته‌بندی ابرنقاط، روشی ارائه شده که با استفاده از نسخه‌ای تغییر یافته از تابع KNN لایه‌هایی از همسایگی شکل می‌گیرد. ویژگی‌های هر یک از این لایه‌های همسایگی استخراج می‌شود و پس از آن با الحاق شدن به ویژگی‌های سراسری برآمده از نقاط منحصر به فرد، از بردار تجمیع شده برای عمل دسته‌بندی استفاده می‌گردد. نتایج به دست آمده از اجرای روش ارائه‌شده بر روی مجموعه‌های دادگان ModelNet40 و ScanObjectNN گواهی می‌دهد که این سازوکار توانسته در مقایسه با دیگر روش‌ها به نتایج و کارایی قابل توجهی دست یابد.

کلید واژه‌ها

ابرنقاط، دسته‌بندی، پردازش داده سه‌بعدی، شبکه‌های عصبی عمیق

ضعف‌ها را فراهم کرده است.



شکل (۱) نمایش ابرنقطه یک صندلی از مجموعه دادگان ModelNet-40 با تعداد نقاط متفاوت [۱]

ابرنقاط یکی از این داده‌های نوظهور است. این نوع داده قابلیت بازنمایی سطوح اجسام در دو و سه بعد را داراست. برای تولید آن کافی است مجموعه‌ای از نقاط به صورت منسجم در کنار یکدیگر قرار گیرند تا سطح یک جسم را نمایش دهند. هر ابرنقطه می‌تواند نمایانگر یک جسم منفرد یا مجموعه‌ای از اجسام و پدیده‌ها باشد. رشد سریع حسگرهای سه‌بعدی و دسترسی پذیری بیشتر آن‌ها در سال‌های اخیر، توجه پژوهشگران را به پردازش ابرنقاط افزایش داده است. در همین راستا، تجهیز تلفن‌های همراه به دوربین‌های

۱- مقدمه

افزایش روزافزون کاربردهای پردازش تصاویر و داده‌های سه‌بعدی در حوزه‌های متنوعی نظیر وسایل نقلیه خودران، رباتیک، سنجش از دور، واقعیت مجازی و بسیاری از زمینه‌های دیگر، توجه پژوهشگران و متخصصان صنعتی را به این شاخه از بینایی ماشین به طور قابل توجهی جلب کرده است [۲، ۳].

اگرچه تصاویر دو بعدی در گستره‌ی بسیار وسیعی مورد استفاده قرار می‌گیرند اما ضعف‌هایی را نیز شامل می‌شوند. از جمله این ضعف‌ها از دست دادن بسیاری از اطلاعات محیطی است. بخش زیادی از جزئیات محیط توسط تصاویر ثبت نمی‌شوند. اطلاعاتی که از دست رفتنشان منجر به کاهش توان پردازشی می‌شود. ظهور داده‌های نوین و شیوه‌های بازنمایی جدید آنچنان که در شکل ۲ قابل مشاهده است امکان برطرف کردن این

این مقاله در ۱۹ تیر ۱۴۰۴ دریافت و در ۱۸ تیر ۱۴۰۵ پذیرفته شد

^{۱،۲،۳} گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه بوعلی

سینا، همدان، ایران

^۱ رایانامه: iman.ai.1372@gmail.com

^۲ رایانامه: khotanlou@basu.ac.ir

^۳ رایانامه: smmoeinp@gmail.com

تورواره‌های دارای عمق مکعبی^۶ نمایش داده می‌شوند. چنین نمایش‌هایی امکان به‌کارگیری شبکه‌های عصبی استاندارد را فراهم می‌کنند و از پیچیدگی‌های پردازش مستقیم ابرنقاط می‌کاهند.

روش‌های ترکیبی^۹: این رویکردها در تلاش‌اند تا مزایای هر دسته را ترکیب کرده و از بخش‌های کارآمد آن‌ها در قالب مدل‌های جدید بهره‌گیرند.

روش‌های مبتنی بر پردازش مستقیم ابرنقاط^{۱۰}: این دسته، که مهم‌ترین گروه محسوب می‌شود، بدون هیچ‌گونه واسطه یا تبدیل داده، مستقیماً با ابرنقاط سه‌بعدی کار می‌کند. این رویکرد چندین زیرشاخه داشته و مدل‌های متنوعی را در بر می‌گیرد.

به‌کارگیری ابزارهای متداول پردازش تصویر در داده‌های ابرنقطه‌ای با دشواری‌های متعددی همراه است. برای نمونه، شبکه‌های عصبی کانولوشنی که یکی از روش‌های قدرتمند در پردازش تصاویر دوبعدی محسوب می‌شوند، به‌صورت مستقیم قابل استفاده در

ابرنقاط نیستند و نیاز به تغییرات ساختاری دارند. تعریف فیلترهای کانولوشنی در فضای ابرنقاط ساده نیست، زیرا برخلاف تصاویر، نقاط فاقد نظم شبکه‌ای هستند و باید همسایگی‌ها به‌طور جداگانه تعریف شوند [۷].

در نتیجه، پژوهشگران برای بهره‌گیری از روش‌های کانولوشنی به جای فیلترهای ثابت از الگوریتم‌های متنوع همسایگی استفاده می‌کنند. الگوریتم‌هایی مانند K-Nearest Neighbors (KNN)، همسایگی شعاعی و همسایگی استوانه‌ای از جمله رویکردهایی هستند که به این منظور به‌کار گرفته شده‌اند [۱۰].

سهم اصلی این پژوهش در پردازش ابرنقاط به شرح زیر است:

- ارائه روشی مبتنی بر شبکه‌های عصبی عمیق برای بهبود دقت در دسته‌بندی ابرنقاط.
- طراحی و معرفی یک بخش برای استخراج ویژگی‌ها در همسایگی هر نقطه.
- به‌کارگیری این بخش به صورت سلسله‌مراتبی در شبکه‌های عصبی چندلایه به منظور استخراج ویژگی‌های نسبی.
- ترکیب ویژگی‌های سراسری استخراج‌شده توسط شبکه‌های عصبی چندلایه با ویژگی‌های محلی حاصل از سطوح متفاوت همسایگی.

ساختار مقاله به صورت زیر تنظیم شده است:

در بخش دوم، کارهای پیشین مرور می‌شود. بخش سوم به

عمق‌سنج مبتنی بر زمان پرواز و نیز توسعه حسگرهایی مانند Tele-15، Leica BLK360 و Kinect V2 دسترسی به داده‌های ابرنقطه‌ای را تسهیل کرده است [۴-۶].

ابرنقاط سه‌بعدی در ساده‌ترین حالت با استفاده از^۱ مختصات x,y,z ایجاد می‌شوند. این سه شاخص موقعیت مکانی هر نقطه در فضا را مشخص می‌کنند (شکل ۱). به این مختصات می‌توان ویژگی‌های دیگری مانند رنگ، بردار نرمال و شدت نور را^۲ نیز افزود که به‌عنوان ویژگی‌های مکمل در پردازش استفاده می‌شوند [۷]. برخلاف تصاویر که ساختاری منظم و شبکه‌ای دارند، ابرنقاط مجموعه‌ای نامنظم و پراکنده‌اند. نقاط در آن‌ها تراکم یکسانی ندارند و در فضای سه‌بعدی به‌طور غیریکنواخت توزیع می‌شوند. این ناهمگنی باعث می‌شود بخش‌های مختلف یک شیء با کیفیت متفاوتی بازنمایی شود و طراحی الگوریتم‌های پردازشی دشوار گردد. هرچند روش‌هایی مانند نمونه‌برداری تصادفی، نمونه‌برداری تورواره^۱ و نمونه‌برداری با معیار دورترین فاصله^۲ برای کاهش این مشکل به کار رفته‌اند، اما ناهمگنی تراکم همچنان یکی از چالش‌های اصلی در پردازش ابرنقاط محسوب می‌شود.

علاوه بر تفاوت در تراکم نقاط، ویژگی دیگری که پردازش ابرنقاط را دشوار می‌سازد، بدون ترتیب بودن نقطه‌ها است. در یک ابرنقطه هیچ نظم یا توالی مشخصی میان نقاط وجود ندارد [۸]. در حالی‌که در تصاویر، پیکسل‌ها به صورت منظم چیده شده‌اند و روند پردازش آن‌ها روشن است، یک ابرنقطه می‌تواند با چینش‌های مختلف ذخیره شود و همچنان بازنمایی یکسانی از جسم ارائه دهد. این ویژگی نیز طراحی الگوریتم‌های پردازشی را با چالش روبه‌رو می‌کند. مجموعه این مشکلات، همراه با چالش‌های دیگر، سبب شده است که روش‌های رایج پردازش تصاویر دوبعدی به سادگی در مورد ابرنقاط قابل استفاده نباشند. در سال‌های اخیر، روش‌های متعددی برای پردازش ابرنقاط ارائه شده که می‌توان آن‌ها را در پنج دسته اصلی تقسیم‌بندی کرد [۹]:

روش‌های مبتنی بر مشبک‌ها^۳: این مدل‌ها تلاش می‌کنند با اعمال ساختار مشبک به داده‌ها، انسجامی مصنوعی ایجاد کرده و از پیچیدگی پردازش مستقیم ابرنقاط بکاهند.

روش‌های مبتنی بر داده‌های انتقال‌یافته^۴: در این رویکرد، ابرنقاط ابتدا به شکل داده‌ای دیگر (مانند تصویر) تبدیل می‌شوند و سپس ویژگی‌های استخراج‌شده از آن داده برای وظایفی همچون دسته‌بندی استفاده می‌گردد.

روش‌های مبتنی بر داده‌های حجمی^۵: در این دسته، داده‌ها به صورت ساختارهایی مانند تورواره‌های سه‌بعدی^۶ از

⁶ 3D grids

⁷ Cubic voxels

^۸ در ادبیات مدل‌های سه بعدی عموماً به تورواره‌های دارای عمق و سه بعدی Voxel گفته می‌شود. تورواره خود ترجمه‌ای از واژه grid است.

⁹ Hybrid methods

¹⁰ Raw point-based methods

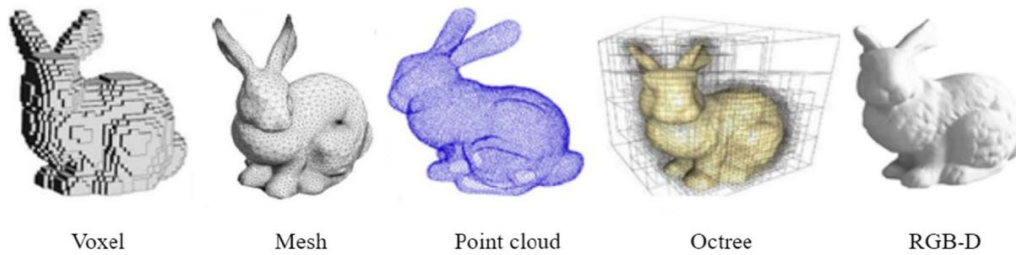
¹ Grid sampling

² Farthest point sampling

³ Mesh-based methods

⁴ Projection-based methods

⁵ Volumetric-based methods



شکل (۲) مدل Stanford bunny که در قالب های گوناگون نمایش داده شده است [۱۱].

می‌دهد [۲۰]. در این رویکرد، هر جسم سه‌بعدی با دو وضوح از پیش تعیین شده بازنمایی و پردازش می‌شود.

دسته‌ای از روش‌ها به صورت ترکیبی عمل می‌کنند [۹]. در این رویکرد، یادگیری مبتنی بر داده‌هایی غیر از ابرنقاط و یادگیری مستقیم از ابرنقاط در قالب یک چارچوب یکپارچه تلفیق می‌شوند و این امر پیشرفت‌های قابل توجهی در طبقه‌بندی ابرنقاط به همراه داشته است. روش PointGrid یکی از مدل‌های شاخص این دسته است که بازنمایی نقطه‌ای و توروارهای^۳ را ترکیب می‌کند [۲۱]. [۲۲] نیز از جمله روش‌های این دسته است که در آن از ابرنقاط و تصاویر ماهواره‌ای چند طیفی با وضوح بالا^۴ در کنار یکدیگر استفاده شده است.

مهم‌ترین دسته از روش‌ها، مدل‌های مبتنی بر پردازش مستقیم ابرنقاط هستند که خود به چندین زیرشاخه تقسیم می‌شوند. بخشی از این روش‌ها بر پایه‌ی پرسپترون چندلایه بنا شده‌اند. مدل PointNet به عنوان یکی از نخستین رویکردهای این دسته مطرح است [۲۳]. در این مدل، هر نقطه به طور مستقل پردازش می‌شود و بنابراین، اطلاعات ساختاری محلی میان نقاط در نظر گرفته نمی‌شود. همین محدودیت سبب شد تا مدل PointNet++ معرفی شود [۲۴]. در PointNet++ ناحیه‌ی محلی هر نقطه به صورت یک کره به مرکزیت نقطه‌ی x تعریف می‌شود و ویژگی‌های این ناحیه به صورت سلسله‌مراتبی استخراج می‌گردند. یانان سانگ و همکارانش روشی ارائه داده‌اند که با بهره از ویژگی‌های محلی و نیز استخراج ویژگی‌های غیر محلی با استفاده از روشی مشابه مکانیزم توجه می‌کوشد دقت را افزایش دهد [۲۵]. در روش Point-BLS هزینه‌ی محاسباتی کاهش پیدا کرده و دقت بهبود نسبتی یافته است [۲۶]؛ با توجه به گستردگی مدل‌های مبتنی بر پردازش مستقیم ابرنقاط، این مدل‌ها خود به چندین زیرشاخه دسته‌بندی شده‌اند، یکی از این زیرشاخه‌ها روش‌های کانولوشنی بر روی ابرنقاط است. برای نمونه، مدل Pointwise-CNN یک شیوه‌ی خاص برای تعریف هسته‌های کانولوشنی ارائه می‌دهد [۲۷]. در این مدل، ابرنقاط سه‌بعدی به توروارهای یکنواخت تبدیل شده و وزن‌هایی به تمامی نقاط موجود در هر واحد تورواره تخصیص

معرفی روش پیشنهادی اختصاص دارد. در بخش چهارم نتایج آزمایش‌ها و ارزیابی‌ها ارائه می‌گردد و در نهایت، نتیجه‌گیری در بخش پنجم بیان خواهد شد.

۲- مروری بر کارهای پیشین

در آغاز، به روش‌های مبتنی بر مشبک‌ها می‌پردازیم. یکی از رویکردهای این دسته، MeshNet است که در آن یک واحد سطحی تعریف می‌شود و این واحد به عنوان پایه‌ای برای انجام عملیات کانولوشن مورد استفاده قرار می‌گیرد [۱۲]. در روش MeshWalker پردازش با پیمایش سطح مشبک و بررسی سطوح و رؤس صورت می‌گیرد [۱۳]. از دیگر روش‌های ارائه‌شده در این دسته می‌توان به CurveNet و PolyNet اشاره کرد که هر یک تلاش می‌کنند ویژگی‌های هندسی نهفته در ساختار مشبک را استخراج نمایند [۱۴، ۱۵].

دسته‌ی دیگر، رویکردهای مبتنی بر تصویرسازی هستند. یکی از روش‌های پیشگام در این گروه MVCNN است [۱۶]. در این رویکرد، نماهای مختلفی از ابرنقاط به صورت تصاویر دویعدی تولید می‌شود و سپس از این تصاویر برای استخراج ویژگی‌ها استفاده می‌گردد. در ادامه، برخی روش‌ها کوشیده‌اند روابط میان نماها را نیز در نظر بگیرند. برای نمونه، در GVCNN نماها بر اساس امتیازهای تمایزبخش به گروه‌هایی تقسیم می‌شوند تا ارتباط میان آن‌ها در بهبود نتایج مورد بهره‌برداری قرار گیرد [۱۷]. دیپش کومار و همکارانش به تازگی روشی برای دسته‌بندی ارائه داده‌اند که طی آن ابرنقاط دریافتی توسط رادارهای موج میلی‌متری^۱ به تصاویر تبدیل می‌شود. این تصاویر در مرحله‌ی بعد با استفاده از مدل‌های ResNet و CNN دسته‌بندی می‌شوند [۱۸].

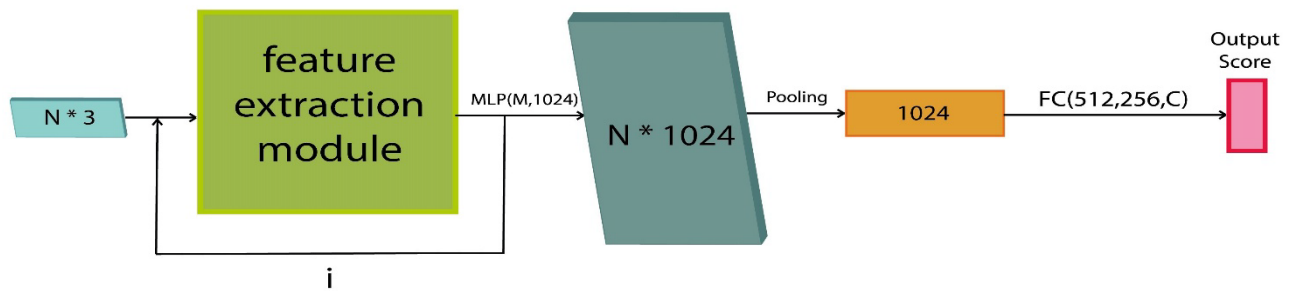
در میان روش‌های مبتنی بر داده‌های حجمی، شبکه‌ی 3D ShapeNet یکی از نخستین مدل‌هاست که بر اساس شبکه‌های باور عمیق طراحی شده است [۱۹]. این شبکه توزیع احتمالاتی نقاط را بر روی توروارها مدل می‌کند. روش MRCNN نیز از یک چارچوب مشبک‌سازی^۲ چندسطحی استفاده می‌کند که داده‌ها را به صورت شبکه‌های اشغال دودویی در دو سطح نمایش

^۳ Point and grid representation

^۴ Very High Resolution (VHR) multi-spectral satellite imagery

^۱ mmWave Radar

^۲ Voxelization



شکل (۳) معماری روش پیشنهادی که در آن N نمایانگر تعداد نقاط، i نمایانگر تعداد دفعات استفاده از لایه‌ی همسایگی، M ابعاد ویژگی‌های حاصل از اعمال مازول پیشنهادی و C برابر است با تعداد کلاس‌ها.

آن هر نقطه از ابرنقاط به تمامی نقاط دیگر متصل است. در این روش، ماتریس لاپلاسی گراف در هر لایه به‌روزرسانی می‌شود و تابع هزینه شامل یک مؤلفه‌ی هموارسازی سیگنال گراف است تا قابلیت مقایسه‌پذیری ویژگی میان رئوس نزدیک به یکدیگر تقویت شود [۳۸]. در مدل PointGCN، گراف بر اساس k نزدیک‌ترین همسایه ساخته می‌شود و وزن هر یال با استفاده از هسته‌ی گوسی محاسبه می‌گردد [۳۹].

۳- روش پیشنهادی

ساختار کلی روش ارائه‌شده با الهام از مدل PointNet طراحی شده است. آن چنان که در شکل ۳ قابل مشاهده است در گام نخست و در بخش استخراج ویژگی اولیه، می‌توان با هر بار اعمال این بخش، ویژگی‌های مبتنی بر هر لایه‌ی همسایگی را استخراج کرد اگر از این بخش استفاده نکنیم در انتهای پردازش تنها ویژگی‌های سراسری ابرنقاط را به کار گرفته‌ایم و با هر بار استفاده از این بخش، با نادیده گرفتن بخشی از نقاط، سطحی از ویژگی‌های غیر سراسری استخراج می‌شود. این ویژگی‌های غیر سراسری استخراج‌شده، با قرارگرفتن در کنار ویژگی‌های سراسری استخراج‌شده در گام‌های بعدی، بردار ویژگی کامل‌تری می‌سازند. در گام دوم از شبکه‌های پرسپترون چندلایه استفاده می‌شود. می‌دانیم که یکی از شاخص‌های چالش‌برانگیز ابرنقاط بدون ترتیب بودن آن است. برای برطرف کردن این چالش و رسیدن به یک مدل تغییرناپذیر نسبت به جایگشت، از پرسپترون چندلایه با وزن‌های اشتراکی استفاده می‌شود. این وزن‌های اشتراکی باعث می‌شود مواجهه شبکه عصبی با همه نقاط یکسان باشد و تغییر ترتیب نقاط، تغییری در کلاس پیش‌بینی شده برای آن‌ها ایجاد نکند. در حقیقت با اعمال چنین روشی ویژگی‌های هندسی هر نقطه مستقلاً نسبت به کل شکل جسم محاسبه می‌شود.

داده می‌شود. در ادامه، روشی برای دستیابی به تغییرناپذیری در برابر چرخش^۱ معرفی شد که با استفاده از عملگر RConv و خوشه‌بندی ویژگی‌های سطح پایین عملیات کانولوشن را انجام می‌دهد [۲۸]. مدل Point-PlaneNet نیز شبکه‌ای عصبی ارائه کرده است که با در نظر گرفتن فاصله‌ی میان نقاط و صفحات، همبستگی‌های مکانی-محلی را به‌طور مؤثرتری استخراج می‌کند [۲۹].

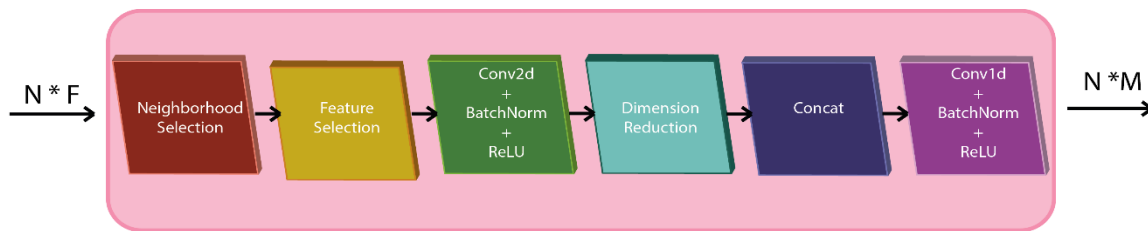
روش KPConv با بهره‌گیری از مجموعه‌ای از نقاط قابل یادگیری، دو نوع هسته‌ی سخت و انعطاف‌پذیر تعریف می‌کند که عملیات کانولوشن بر اساس آن‌ها انجام می‌گیرد [۱۰]. از دیگر روش‌های مطرح این دسته می‌توان به PointConv، SpiderCNN و PCNN اشاره کرد [۳۰-۳۲].

روش‌های گراف‌محور نیز زیرمجموعه‌ای از رویکردهای پردازش مستقیم ابرنقاط محسوب می‌شوند. در مدل PointWeb، که بر پایه‌ی PointNet++ طراحی شده است، یک گراف در فضای ویژگی‌ها ایجاد می‌شود. این گراف پس از هر لایه به‌صورت پویا به‌روزرسانی می‌گردد [۳۳]. مدل DGCNN نیز گرانی در فضای ویژگی‌ها تولید می‌کند، اما از یک شبکه‌ی پرسپترون چندلایه (MLP) برای یادگیری ویژگی‌ها در یال‌های گراف استفاده می‌کند [۳۴]. [۳۵] نشان داده که DGCNN در مقایسه با PointNet دقت بیشتر و پارامترهای کمتری دارد اما کندتر است.

روش M-GCN در دو مقیاس با دو تراکم متفاوت ویژگی استخراج می‌کند. مقیاس اول تراکم بیشتر و مقیاس دوم پراکندگی بیشتری دارد [۳۶]. مدل GF-DGCNN با تکیه بر ویژگی‌های سراسری از جمله روش‌هاییست که تحت تأثیر روش پیشین ارائه شده است. این روش علاوه بر استفاده از نقاطی یکسان برای استخراج ویژگی‌های محلی، با استفاده از یک بخش استخراجی ویژگی سراسری اضافی سعی می‌کند دقت دسته‌بندی را بهبود دهد [۳۷]. در مقابل، مدل RGCNN گرانی ایجاد می‌کند که در

¹ Rotation invariant

Feature Extraction Module



شکل (۴) ماژول استخراج ویژگی که در آن N نمایانگر تعداد نقاط، F نشاندهنده ویژگی‌های ورودی و M نشاندهنده ویژگی‌های خروجی است.

نقطه را انتخاب کرد. بعد از انتخاب این نقاط به عنوان ناحیه مورد نظر، کانولوشن‌های دوبعدی بر آن اعمال می‌شود تا ویژگی‌های آن استخراج شود. معماری بخش استخراج ویژگی در شکل ۴ و ساختار تابع تغییر یافته knn در ادامه آمده است.

$$N_{HKNN}(y_0, P, k, h) = \{x_{k,h} \in P, |y_0 - x_{k,h-1}| \leq |y_0 - x_{k,h-1}|, |y_0 - x_{k,h}| \leq |y_0 - x_{k,h-1}|, k, h \in N\} \quad (1)$$

در این روش، k تعداد همسایه‌ها، h لایه همسایگی، y_0 نقطه مورد نظر و P مجموعه ابرنقاط هستند. همسایگی h -ام شامل k همسایه برای نقطه y_0 است. خروجی، مجموعه‌ای از نقاط خواهد بود که بر اساس فاصله از هر نقطه انتخاب می‌شوند.

۴- ارزیابی نتایج

در این بخش به بررسی و ارزیابی نتایج به دست آمده از روش پیشنهادی می‌پردازیم. برای اینکار ابتدا پایگاه داده را معرفی کرده، سپس از معیارهای ارزیابی مورد استفاده سخن می‌گوییم. جزئیات پیاده‌سازی و نیز مقایسه با دیگر روش‌ها در انتهای این بخش آمده است.

۴-۱- پایگاه داده

در این پژوهش از دو مجموعه داده ModelNet40 و ScanObjectNN استفاده شده است. پایگاه داده ModelNet40 شامل ۱۲۳۱۱ ابرنقطه است که در ۴۰ کلاس تقسیم شده‌اند. از این تعداد ۹۸۴۳ ابرنقطه برای آموزش و ۲۴۶۸ ابرنقطه برای آزمون به کار گرفته می‌شود. داده‌های این مجموعه مصنوعی هستند و همگی از روی مشبک نمونه‌برداری شده‌اند. هیچگونه انسدادی در این پایگاه داده وجود ندارد و فاقد پس‌زمینه می‌باشد. پایگاه داده ScanObjectNN شامل ۱۴۸۰۲ شی است که در ۱۵ دسته با ۲۹۰۲ نمونه شی منحصر به فرد طبقه بندی شده‌اند. این پایگاه داده از اسکن سه بعدی اجسام در دنیای واقعی ایجاد شده است. به همین دلیل انسداد و نیز ناهنجاری‌های ممکن در هنگام دریافت

عملی که به زبانی دیگر کانولوشن نقطه‌ای^۱ خوانده می‌شود. پس از آن و در گام سوم ویژگی‌های به دست آمده توسط پرسپترون چندلایه به یک تابع متقارن سپرده می‌شود. در اینجا تابع متقارن عملگر ادغام بیشینه است. هدف از اعمال عملگر ادغام انتخاب قوی‌ترین ویژگی‌ها از میان همه ویژگی‌های استخراج شده است. خروجی این گام یک بردار ویژگیست که در گام آخر به لایه‌های تماماً متصل داده می‌شود تا با به کار بردن تابع softmax، احتمال هر کلاس، محاسبه شده و عمل پیش‌بینی صورت گیرد. بخش استخراج ویژگی در ادامه توضیح داده شده است.

۳-۱- بخش استخراج ویژگی

در حوزه پردازش تصویر برای استخراج ویژگی‌های محلی از روش‌های کانولوشنی استفاده می‌شود. با مبنا قرارگرفتن هر پیکسل، بافت‌های حاصل از پیکسل‌های مجاور آن بررسی شده، از آن ویژگی استخراج می‌شود. از راه حل‌های ابتدایی برای بررسی بافت‌های محلی گسترده‌تر، بزرگ‌تر کردن فیلترهای کانولوشنی است. این راه حل علیرغم دریافت اطلاعات بیشتر با توجه به هزینه‌ی زیاد محاسباتی چالش‌آفرین است. در گذر زمان روش‌هایی همچون Atrous convolution ارائه شدند تا موازنه‌ای میان بار محاسباتی و استخراج ویژگی برقرار کنند [۴۰].

این چالش در حوزه پردازش ابر نقاط نیز وجود دارد. صورت استفاده از همسایگی‌های گسترده‌تر به منظور استخراج ویژگی‌های محلی، بار محاسباتی زیادی به سیستم تحمیل می‌شود. به همین منظور لازم است به جای بزرگ‌تر کردن اندازه‌ی همسایگی‌ها، از سازوکاری استفاده کنیم که ضمن از دست نرفتن اطلاعات، هزینه‌ی محاسباتی کمتری به مدل تحمیل کند. روش ارائه‌شده با الهام از روش Atrous convolution و با هدف بررسی بافت‌های با تراکم یکسان اما در موقعیت‌های متفاوت، می‌تواند ویژگی‌های غیرسراسری و نسبی را بدون تحمیل هزینه‌ی اضافه، محاسبه کند. به این منظور سازوکار الگوریتم KNN با کمی تغییر، با نیاز روش ارائه‌شده سازگار شده است. با استفاده از این نسخه‌ی تغییر یافته می‌توان در هر بار فراخوانی تابع، k همسایه‌ی h ام هر

^۱ Pointwise convolution

۵-۴- تحلیل روش ارائه شده

در تصاویر دو بعدی فیلترهای کانولوشنی ابزاری برای استخراج بافت‌های معنادار یا معنابخش تصاویر هستند. هر یک از این فیلترها با استفاده از معیار همبستگی و نیز با نمایندگی کردن قالبی خاص توان بیرون کشیدن الگوهای متفاوتی را از درون تصویر دارا هستند. الگوهایی که در نهایت به عنوان ویژگی‌های استخراج شده به فهم و تقویت توان جداکنندگی تصاویر توسط مدل کمک می‌کنند. در فضای ابر نقاط و با توجه به ویژگی‌هایی که پیشتر عنوان شد طراحی فیلترهایی همچون آن‌ها که در تصاویر مورد استفاده قرار می‌گیرد دشوار است.

بنابراین چاره‌ای نیست جز این که با در نظر گرفتن محدودیت‌های موجود و نیز تجربه‌ی کارکردهای فیلترهای کانولوشنی در حوزه پردازش تصاویر، اقدام به طراحی روش‌هایی کنیم که بتواند در فضای ابرنقاط ویژگی استخراج کند. Dilated convolution روشیست که در فضای پردازش تصاویر و پیکسل‌ها به کانولوشن چند مقیاسی می‌پردازد [۴۰، ۴۲]. فیلترها در عین حفظ ابعاد خود به جای برداشتن گام بیرونی و غلتیدن بر روی تصویر، میان خانه‌های خود فاصله ایجاد می‌کنند تا بافت مورد نظر خود را در مقیاسی دیگر جست‌وجو کنند. به عنوان مثال همانطور که در شکل ۶ مشاهده می‌کنید یک فیلتر ۳*۳ به جای اینکه بر ۹ خانه‌ی همجوار اعمال شود، با نرخ یک پیکسل گسترش پیدا می‌کند و با مقیاسی دیگر بر ۹ پیکسل دیگر اعمال می‌شود. با چنین ایده‌ای این روش قادر است با نرخ‌های گسترش متفاوت به ویژگی‌هایی فراتر از ویژگی‌های محلی دست یابد. ویژگی‌هایی که ما ویژگی‌های نسبی نامیدیمشان در حقیقت بافت‌هایی هستند که از پیکسل‌های ناهم‌جوار قابل استخراجند.

متناسب با آن در فضای ابرنقاط و در روش ارائه‌شده تلاشی برای به دست آوردن بافت‌های ناهم‌جوار هر نقطه صورت گرفته است. اگر نه همچون پیکسل‌ها، اما نقاط نیز با در کنار یکدیگر قرار گرفتن خود، به هر شکل و هر میزان تراکم و فاصله، نمایانگر بخشی از ویژگی‌های جسمی هستند که آن را بازنمایی می‌کنند. بنابراین یافتن بافت‌ها و ویژگی‌هایی که از هر جایگشت متفاوت از نقاط، شکل گرفته اصلی‌ترین منبع ما برای استخراج و دریافت ویژگی است. حال هر چه محدودیت‌های عملیاتی در فضای ابرنقاط کاهش یابد امکان استخراج ویژگی‌های معنادار افزایش می‌یابد.

در روش ارائه شده کوشیده شده با به کارگیری ابزارهای موجود گامی در جهت رسیدن به ویژگی‌های مورد نظر برداشته شود و آنچنان که نتایج گواهی می‌دهند علیرغم محدودیت‌هایی که نوع داده ایجاد می‌کند این روش توانسته به فهم بهتر داده‌ها کمک کند. شکل‌های ۷ تا ۱۰ نشان‌دهنده نمونه‌هایی از کیفیت دسته‌بندی مدل بر اساس داده‌های مجموعه دادگان ModelNet40 است. تشخیص‌های اشتباه به گونه‌ایست که حتی با چشم انسانی نیز احتمال خطا بسیار زیاد است.

ابرنقاط را نیز شامل می‌شود. همانند مجموعه دادگان پیشین، این مجموعه نیز با نسبت ۸۰ به ۲۰ برای آموزش و آزمون جدا سازی شده است.

۲-۴- معیارهای ارزیابی

برای ارزیابی مسائل مختلف در حوزه‌ی ابرنقاط، معیارهای گوناگونی تعریف شده است. در مورد مسئله دسته‌بندی ابرنقاط این معیارها عبارتند از Overall accuracy و Mean class accuracy. OA به عنوان نخستین معیار، عبارت است از دقت میانگین بر روی همه‌ی نمونه‌های تست و mAcc عبارت است از میانگین دقت بر روی همه کلاس‌ها.

$$mAcc = \frac{1}{|class|} \sum_{i \in class} \frac{TOTAL CORRECT_i}{TOTAL SEEN INSTANCE_i} \quad (2)$$

$$OA = \frac{TOTAL INSTANCE}{TOTAL SEEN INSTANCE} \quad (3)$$

که در دو عبارت فوق صورت کسرها به معنای آن دسته از نمونه‌هایی است که به درستی تشخیص داده شده‌اند و نیز مخرج کسر به معنای همه نمونه‌های بررسی شده‌اند. تعداد کلاس‌ها به صورت $|class|$ نشان داده شده است.

۳-۴- جزئیات پیاده‌سازی

برای پیاده‌سازی این پژوهش از زبان برنامه‌نویسی پایتون ۳/۹ و بر بستر Pytorch استفاده شده است. این نسخه‌ی پیاده‌سازی شده بر روی دو سیستم مورد آزمایش قرار گرفته است. نخستین سیستم دارای پردازنده‌ی گرافیکی ۳۰۷۰ Ti با میزان حافظه‌ی اصلی ۸ گیگابایت و دومین سیستم دارای پردازنده گرافیکی ۱۰۷۰ Ti و با حافظه‌ی اصلی ۱۶ گیگابایت بوده است.

برای آموزش این شبکه از بهینه‌ساز ADAM استفاده شده است [۴۱]. آموزش، در بسته‌های ۱۶، ۲۴ و ۳۲ تایی، با نرخ یادگیری ۰/۰۰۱ و نیز با نرخ کاهشی ۰/۰۰۰۱ صورت پذیرفته است. از طرفی تعداد همسایه‌های گوناگونی از میان آن‌ها با توجه به نتیجه‌های به دست آمده، تعداد ۳۲ همسایه برای هر لایه در نظر گرفته شده است. ۲۵۰ بار روند آموزش تکرار گشته و در نهایت به نتیجه‌ی ارائه‌شده همگرا شده است.

۴-۴- مقایسه با دیگر روش‌ها

نتایج حاصل از روش ارائه‌شده در جدول ۱، ۲، ۳ و ۴ آمده است. آزمایش‌ها نشان می‌دهد استفاده از دو لایه‌ی همسایگی در مقایسه با یک لایه‌ی همسایگی به نتیجه‌ی بهتری منجر می‌شود. و همینطور با افزودن سومین لایه همسایگی، دقت بهبود می‌یابد. نتایج آزمایش‌ها همچنین نشان می‌دهد که روش ارائه‌شده نسبت به بسیاری از روش‌های مطرح این حوزه عملکرد بهتری دارد. این درحالیست که تعداد پارامترهای روش ارائه شده نسبت به مطرح‌ترین روش‌ها به شکل قابل توجهی کمتر است.

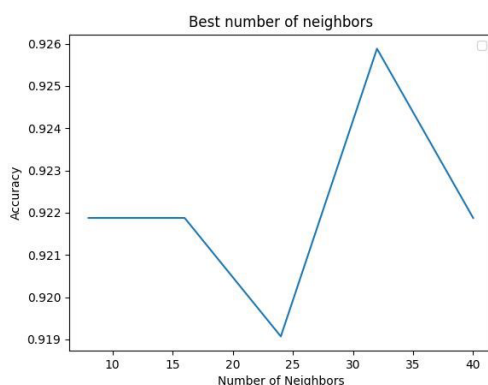
جدول (۴) مقایسه مدل‌ها بر اساس تعداد پارامتر و نیز دقت دسته‌بندی بر مبنای مجموعه دادگان ModelNet40

روش	تعداد پارامتر	دقت میانگین (%)	دقت کل (%)
DGCNN	$\sim 1/8m$	۹۰/۲	۹۲/۹
PointNet++	$\sim 1/7m$	-	۹۰/۷
M-GCN	$\sim 1/22m$	۹۰/۱	۹۳/۱
LNLCL-Net	$\sim 1/1m$	۹۱/۱	۹۳/۶
Ours(Three layer)	$m(1134656) \sim 1/1$	۹۰/۷	۹۳/۰

۴-۴- بررسی میزان تراکم نقاط در هر لایه

در ارائه‌ی فرضیه‌ی این پژوهش بنا بر این بود که برای فهم، شناخت و یادگیری یک جسم سه‌بعدی توسط مدل شبکه عصبی نباید تنها به ویژگی‌های ساختاری ایجاد شده از نقاط نزدیک هر نقطه بسنده کرد و چه بسا بتوان از روابط دور دست هر نقطه به ویژگی‌هایی دست یافت که توان پردازشی شبکه در تشخیص اجسام را بهبود دهد و در نهایت منجر به بهبود دقت تصمیم‌گیری شبکه شود. با توجه به ابزارهای موجود برای استخراج ویژگی عاملی که می‌تواند ما را به این خواسته نزدیک کند استفاده از تراکم نقاط است.

پس از این فرضیه یکی از پرسش‌هایی که ذهن را به خود مشغول می‌کند این است که ما تا چه مقدار می‌توانیم از عنصر تراکم استفاده کنیم و چه میزان اثرگذاری بر روی عملکرد مدل خواهد داشت. برای رسیدن به پاسخ این پرسش، میزان اثرگذاری لایه‌های همسایگی با مقادیر متفاوتی از تعداد همسایه‌ها آزموده شده است. مقادیری که عمدتاً مضارب عدد ۸ هستند. بنا بر نتایج به دست آمده، از میان تعداد همسایه‌های مورد بررسی قرار گرفته، بهترین تعداد همسایه برای انتخاب در هر لایه همسایگی عدد ۳۲ می‌باشد. شکل ۵ میزان اثرگذاری تعداد همسایه‌ها را در معماری دولایه نشان می‌دهد.



شکل (۵) اثر تعداد همسایه‌ها بر دقت دسته‌بندی

جدول (۱) مقایسه و ارزیابی مدل بر مبنای پایگاه داده ModelNet40

روش	mAcc-دقت میانگین (%)	OA-دقت کل (%)
PointNet	۸۶/۲	۸۹/۲
PointNet++	-	۹۰/۷
Crossmoco[43]	-	۹۱/۴۹
Kd-Network[44]	۸۸/۵	۹۱/۸
PointPlaneNet	۹۱/۰	۹۲/۰
PointCNN[45]	-	۹۲/۲
PointsToPatches[46]	-	۹۲/۶
Improved Non-local Neural network[25]	۸۸/۹	۹۲/۶
DGCNN	۹۰/۲	۹۲/۹
M-GCN[36]	۹۰/۱	۹۳/۱
GF-DGCNN[37]	۹۱/۰	۹۳/۳
LNLCL-Net [47]	۹۱/۱	۹۳/۶
Point-BLS[26]	-	۸۷/۴۸
Ours(one layer)	۸۹/۶	۹۲/۳
Ours(Two layer)	۹۰/۵	۹۲/۶
Ours(Three layer)	۹۰/۷	۹۳/۰

جدول (۲) مقایسه و ارزیابی مدل بر مبنای پایگاه داده

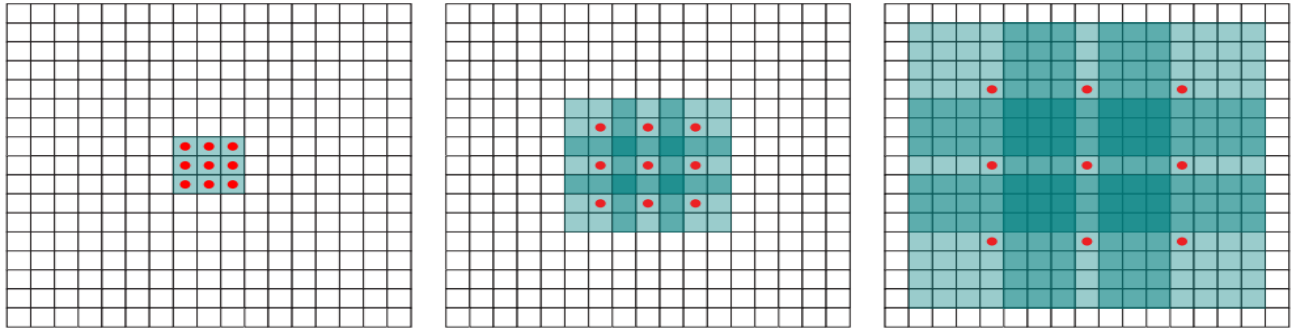
ScanObjectNN

روش	mAcc-دقت میانگین (%)	OA-دقت کل (%)
PointNet	۶۳/۴	۶۸/۲
PointNet++	۷۵/۴	۷۷/۹
DGCNN	۷۳/۶	۷۸/۱
PointCNN	۷۵/۱	۷۸/۵
BGA-DGCNN[48]	۷۵/۷	۷۹/۷
LNLCL-Net	۷۹/۷	۸۲/۵
Ours(one layer)	۷۵/۶	۷۷/۳
Ours(Two layer)	۷۹/۹	۸۱/۰
Ours(Three layer)	۸۰/۱	۸۱/۵

جدول (۳) مقایسه مدل‌ها بر اساس تعداد پارامتر و نیز دقت دسته‌بندی

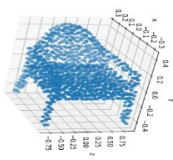
بر مبنای مجموعه دادگان ScanObjectNN

روش	تعداد پارامتر	دقت میانگین (%)	دقت کل (%)
DGCNN	$\sim 1/8m$	۷۳/۶	۷۸/۱
PointNet++	$\sim 1/7m$	۷۵/۴	۷۷/۹
LNLCL-Net	$\sim 1/1m$	۷۹/۷	۸۲/۵
Ours(Three layer)	$\sim 1/1m$	۸۰/۱	۸۱/۵

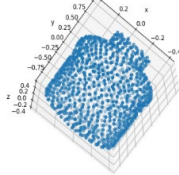


شکل (۶) نمونه‌ای از کانولوشن گسترش یافته در فضای تصویر [۴۰]

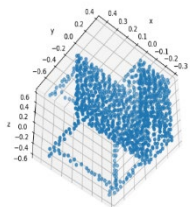
چشمگیرتر بوده و دقت از ۷۷/۳ درصد به ۸۱/۵ درصد افزایش یافته است. این موضوع نشان می‌دهد که استخراج ویژگی‌های غیرمحلی در لایه‌های عمیق‌تر به بهبود عملکرد کمک می‌کند، به‌ویژه در داده‌های پیچیده‌تر که نیاز به درک زمینه‌های گسترده‌تری دارند.



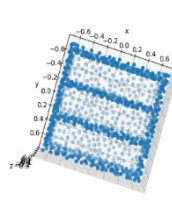
شکل (۸) تشخیص اشتباه مبلی به جای نیمکت



شکل (۷) تشخیص صحیح بطری



شکل (۱۰) تشخیص اشتباه میز به جای نیمکت



شکل (۹) تشخیص صحیح کتابخانه

۴-۷-۲ تحلیل پیچیدگی محاسباتی

تعداد پارامترها و حجم مدل

روش پیشنهادی با ۱,۱۳۴,۶۵۶ پارامتر قابل یادگیری، در دسته مدل‌های سبک قرار می‌گیرد و حجم مدل ذخیره‌شده تنها ۴/۳۳ مگابایت در حالت (FP32) است. این تعداد پارامتر در مقایسه با بسیاری از روش‌های مبتنی بر گراف و یادگیری زمینه‌ای، به مراتب کمتر است. به عنوان مثال، روش پیشنهادی نسبت به DGCNN با حدود ۱/۸ میلیون پارامتر، حدود ۳۷ درصد کاهش پارامتر داشته و نسبت به PointNet++ با حدود ۱/۷ میلیون پارامتر، حدود ۳۳ درصد کاهش نشان می‌دهد. در مقایسه با

۴-۷ تحلیل نتایج صحت و پیچیدگی زمانی

۴-۷-۱ تحلیل نتایج صحت

روش پیشنهادی بر روی دو مجموعه داده استاندارد ModelNet40 و ScanObjectNN مورد ارزیابی قرار گرفته است. مجموعه داده ModelNet40 شامل داده‌های تمیز و منظم آزمایشگاهی است، در حالی که ScanObjectNN شامل داده‌های دنیای واقعی با نویز، نقاط پرت و تغییرات شیء می‌باشد که چالش بسیار بیشتری را برای مدل‌های طبقه‌بندی ایجاد می‌کند.

در مجموعه داده ModelNet40، روش پیشنهادی با معماری سه‌لایه به دقت کلی ۹۳/۰ درصد و دقت میانگین ۹۰/۷ درصد دست یافته است. این نتایج در مقایسه با روش‌های پایه مانند PointNet با دقت ۸۹/۲ درصد و PointNet++ با دقت ۹۰/۷ درصد، بهبود قابل توجهی را نشان می‌دهد. همچنین در مقایسه با روش‌های هم‌رده مانند DGCNN با دقت ۹۲/۹ درصد و LNLCL-Net با دقت ۹۳/۶ درصد، روش پیشنهادی با اختلاف اندکی از بهترین روش فاصله دارد، اما این اختلاف ناچیز در حالی رخ داده که هزینه محاسباتی بسیار کمتری را تحمیل کرده است.

در مجموعه داده ScanObjectNN که داده‌های آن از دنیای واقعی جمع‌آوری شده و دارای پیچیدگی‌های فراوانی هستند، روش پیشنهادی سه‌لایه موفق به ثبت دقت کلی ۸۱/۵ درصد و دقت میانگین ۸۰/۱ درصد شده است. این نتایج نسبت به DGCNN با دقت ۷۸/۱ درصد و PointNet++ با دقت ۷۷/۹ درصد، برتری محسوسی دارد و نشان‌دهنده مقاومت بالای روش پیشنهادی در برابر نویز و نقاط پرت است. نکته قابل توجه، شکاف عملکردی کمتر این روش بین داده‌های تمیز و نویزی است. در حالی که این شکاف برای DGCNN حدود ۱۴/۸ درصد و برای PointNet++ حدود ۱۲/۸ درصد است، روش پیشنهادی این شکاف را به ۱۱/۵ درصد کاهش داده که نشان از تعمیم‌پذیری بهتر مدل دارد.

بررسی تأثیر تعداد لایه‌های شبکه نشان می‌دهد که افزایش تعداد لایه‌ها از یک به سه لایه، روند صعودی پایداری در دقت ایجاد کرده است. در مجموعه داده ModelNet40، دقت از ۹۲/۳ درصد در معماری یک‌لایه به ۹۳/۰ درصد در معماری سه‌لایه رسیده است. در مجموعه داده ScanObjectNN این بهبود بسیار

ویژگی‌های میانی است. در مقایسه با DGCNN با حدود ۶۵۰ مگابایت، افزایش تنها حدود ۵/۵ درصد بوده که تقریباً قابل چشم‌پوشی است. اما در مقایسه با PointTransformer با حدود ۲۵۰۰ مگابایت، مصرف حافظه حدود ۷۲ درصد کاهش یافته که دستاورد بسیار ارزشمندی محسوب می‌شود. همچنین نسبت به LNLCL-Net با حدود ۷۰۰ مگابایت، مصرف حافظه روش پیشنهادی کمتر است.

با توجه به این نتایج، می‌توان نتیجه گرفت که معماری پیشنهادی از نظر مصرف حافظه نیز کارآمد بوده و حتی بر روی پردازنده‌های گرافیکی با ظرفیت حافظه متوسط (۴ گیگابایت) نیز قابلیت آموزش و استقرار دارد. این ویژگی، مدل را برای سیستم‌های تعبیه‌شده و محیط‌های دارای محدودیت منابع محاسباتی بسیار مناسب می‌سازد.

در مجموع، نتایج حاصل از تحلیل صحت و پیچیدگی محاسباتی نشان می‌دهد که مدل پیشنهادی با حفظ دقت ۹۳ درصد بر روی دیتاست ModelNet40 و ۸۱/۵ درصد بر روی ScanObjectNN، توانسته است توازن مناسبی میان تعداد پارامترها، زمان آموزش، سرعت استنتاج و مصرف حافظه GPU برقرار کند. اگرچه زمان آموزش نسبت به روش‌های ساده‌تر مانند PointNet افزایش یافته است، اما این افزایش هزینه در مقابل بهبود قابل‌توجه دقت و مقاومت در برابر نویز کاملاً توجیه‌پذیر است. همچنین در مقایسه با مدل‌های هم‌رده از نظر دقت، روش پیشنهادی با تعداد پارامتر کمتر و مصرف حافظه رقابتی، عملکردی کارآمد و متعادل ارائه داده است.

این ویژگی‌ها، مدل را به گزینه‌ای مناسب برای کاربردهای بلادرنگ، سامانه‌های تعبیه‌شده و محیط‌های دارای محدودیت منابع محاسباتی تبدیل می‌کند، بدون آنکه کاهش قابل توجهی در عملکرد طبقه‌بندی نسبت به روش‌های متداول مشاهده شود.

۵- نتیجه‌گیری

نتایج گواه بر این امر است که با استخراج روابط میان نقاط و استفاده از ویژگی‌های منحصربه‌فرد هر نقطه، مدل می‌تواند به فهم بهتری نسبت اشیاء برسد و با تکیه بر آن دقت تصمیم‌گیری را افزایش دهد. از کارهای آینده بررسی و طراحی مدل‌هاییست که قادر به استخراج بهتر روابط میان نقاط باشد و ویژگی‌های نسبی را به شکل بهینه‌تری به کار گیرد تا الگوهای پیچیده‌تر نیز کشف شود و دقت تصمیم‌گیری بهبود یابد.

مراجع

- [1] Z. Wu et al., "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015: IEEE, pp. 1912–1920.
- [2] S. D. Pendleton et al., "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

PointNet با حدود ۰/۸ میلیون پارامتر، افزایشی حدود ۴۲ درصدی مشاهده می‌شود که این افزایش با بهبود قابل‌توجه ۳/۸ درصدی در دقت همراه بوده است. همچنین در مقایسه با مدل‌های سنگین‌تر مانند PointTransformer با حدود ۸ میلیون پارامتر، کاهش چشمگیر ۸۶ درصدی در تعداد پارامترها رخ داده که با وجود تنها ۰/۷ درصد کاهش دقت، دستاورد بسیار ارزشمندی محسوب می‌شود. تعداد پایین پارامترها علاوه بر کاهش هزینه ذخیره‌سازی مدل، موجب کاهش سربار محاسباتی و حافظه موردنیاز در مراحل آموزش و استنتاج نیز شده است.

زمان آموزش و استنتاج

از نظر کارایی زمانی، میانگین زمان اجرای Forward Pass برابر با ۰/۲۳۱۲ ثانیه و میانگین زمان استنتاج در هر دوره برابر با ۳۴/۵۵ ثانیه است که نشان می‌دهد مدل از سرعت مناسبی در فرآیند استنتاج برخوردار است. همچنین، میانگین زمان هر دوره آموزشی برابر با ۲۷۵/۴۶ ثانیه و زمان کل آموزش برابر با ۷۷۵۰۲/۷۴ ثانیه (حدود ۲۱/۵ ساعت) بیانگر آن است که فرآیند آموزش، با وجود استخراج و یادگیری ویژگی‌های هندسی ابرنقاط، از نظر زمانی در محدوده‌ای قابل قبول قرار دارد.

در مقایسه با روش‌های دیگر، زمان آموزش نسبت به PointNet که هر دوره را حدود ۸۰ ثانیه پردازش می‌کند، حدود ۳/۴ برابر افزایش یافته است. این افزایش عمدتاً به دلیل جستجوی همسایگی با ۳۲ همسایه و محاسبه ویژگی‌های غیرمحلی در هر سه لایه است. با این حال، در مقایسه با DGCNN که هر دوره را حدود ۲۰ تا ۳۰ ثانیه انجام می‌دهد، زمان آموزش حدود ۱۰ برابر بیشتر شده است. اما از سوی دیگر، نسبت به مدل‌های پیچیده‌تر مانند LNLCL-Net با حدود ۳۵۰ ثانیه در هر دوره و PointTransformer با حدود ۴۰۰ ثانیه در هر دوره، زمان آموزش روش پیشنهادی کمتر است. نکته مهم این است که سرعت استنتاج ۰/۲۳ ثانیه به ازای هر نمونه، معادل حدود ۴ فریم بر ثانیه، برای بسیاری از کاربردهای بلادرنگ قابل قبول است و این ویژگی، مدل را برای کاربردهای عملی مناسب می‌سازد.

مصرف حافظه GPU

از منظر مصرف حافظه، اندازه‌گیری مستقیم اوج مصرف حافظه GPU نشان داد که مصرف حافظه در پایان مرحله Forward برابر با ۵۱۵/۵ مگابایت بوده و در مرحله Backward به بیشینه مقدار ۶۸۶/۱۶ مگابایت افزایش یافته است. این افزایش عمدتاً ناشی از تخصیص حافظه برای گرادینانها و متغیرهای میانی موردنیاز در فرآیند پس‌انتشار است و عدم افزایش بیشتر حافظه پس از مرحله به‌روزرسانی پارامترها، نشان می‌دهد که سربار حافظه ناشی از بهینه‌ساز ناچیز است.

در مقایسه با روش‌های دیگر، مصرف حافظه نسبت به PointNet با حدود ۳۰۰ مگابایت، حدود ۲/۳ برابر افزایش یافته که این افزایش به دلیل پیچیدگی بیشتر معماری و ذخیره‌سازی

- and Deep Learning Models," in *2025 7th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2025: IEEE, pp. 141–145 .
- [19] Z. Wu *et al.*, "3d shapenets: A deep representation for volumetric shapes ",in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920 .
- [20] G. Sambit, Y. Xian, B. Aditya, S. Soumik, and K. Adarsh, "Multi-level 3D CNN for learning multi-scale spatial features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019 .
- [21] T. Le and Y. Duan, "Pointgrid: A deep network for 3d shape understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9204–9214 .
- [22] M. Fawzy, M. Dowajy, T. Lovas, and A. Barsi, "Urban Land Cover Classification Using Deep Neural Networks Based on VHR Multi-Spectral Image and Point Cloud Integration," in *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium :۲۰۲۴* , IEEE, pp. 5377–5381 .
- [23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–66 .
- [24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.
- [25] Y. Song, X. Liu, W. Shen, Y. Gao, X. Zhou, and P. Lu, "An effective point cloud classification method based on improved non-local neural networks," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2022: IEEE, pp. 665–670 .
- [26] Y. Chen, M. Fu, and K. Shen, "Point-BLS: 3D point cloud classification combining deep learning and broad learning system," in *2022 34th Chinese Control and Decision Conference (CCDC)*, 2022: IEEE, pp. 2810–2815 .
- [27] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 984–993 .
- [28] A. Montanaro, D. Valsesia, and E. Magli, "Rethinking the compositionality of point clouds through regularization in the hyperbolic space," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33741–33753, 2022.
- [29] S. M. Peyghambarzadeh, F. Azizmalayeri, H. Khotanlou, and A. Salarpour, "Point-PlaneNet: Plane kernel based convolutional neural network for point clouds analysis," *Digital Signal Processing*, vol. 98, p. 102633, 2020.
- [30] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621 . ۹۶۳۰–
- [31] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 87–102 .
- [3] J. W. Weingarten, G. Gruener, and R. Siegwart, "A state-of-the-art 3D sensor for robot navigation," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, 2004, vol. 3: IEEE, pp. 2155–2160 .
- [4] TelemetriQ. "Tele-15 Sensor." TelemetriQ Inc. <https://www.telemetriq.com/tele-15> (accessed 28 Sep. 2025).
- [5] L. Geosystems. "Leica BLK360 Imaging Laser Scanner." Leica Geosystems AG. <https://leica-geosystems.com/products/laser-scanners/scanners/blk360> (accessed 28 Sep. 2025).
- [6] Microsoft. "Kinect for Windows v2 Sensor." Microsoft Corporation. <https://www.microsoft.com/en-us/download/details.aspx?id=44561> (accessed ۲۸ Sep. 2025).
- [7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.
- [8] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Deep learning on 3D point clouds," *Remote Sensing*, vol. 12, no. 11, p. 1729, 2020.
- [9] S. Sarker *et al.*, "A comprehensive overview of deep learning techniques for 3D point cloud classification and semantic segmentation," *Machine Vision and Applications*, vol. 35, no. 4, p. 67, 2024.
- [10] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420 .
- [11] G. Turk, "The Stanford bunny, 2000,[Online; accessed May 14, 2007]," ed.
- [12] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3d shape representation," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, no. 01, pp. 8279–8286 .
- [13] A. Lahav and A. Tal, "Meshwalker: Deep mesh understanding by random walks," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–13, 2020.
- [14] A. Muzahid ,W. Wan, F. Sohel, L. Wu, and L. Hou, "CurveNet: Curvature-based multitask learning deep networks for 3D object recognition," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 6, pp. 1177–1187, 2020.
- [15] M. Yavartanoo, S.-H. Hung, R. Neshatavar, Y. Zhang, and K. M. Lee, "Polynet: Polynomial neural network for 3d shape recognition with polyshape representation," in *2021 international conference on 3D vision (3DV)*, 2021: IEEE, pp. 1014–1023 .
- [16] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953 .
- [17] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "Gvcnn: Group-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 264–272 .
- [18] D. Kumar, R. Sharma, A. Mohan, H. K. Meena, P. Mahajan, and A. Srivastava, "A Point Cloud-Based Multi-class Car Classification by Using mmWave Radar

benchmark dataset and classification model on real-world data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1588–1597.



ایمان احمدی دانش آموخته‌ی کارشناسی ارشد هوش مصنوعی از دانشگاه بوعلی سینا همدان است. او هم‌اکنون در واحد تحقیق و توسعه پتروشیمی دماوند مشغول فعالیت بر روی سامانه‌های هوشمند صنعتی می‌باشد. از زمینه‌های مورد علاقه ایشان می‌توان به بینایی ماشین، رباتیک و سیستم‌های خودمختار، پردازش تصویر و ویدیو و نیز یادگیری ماشین اشاره کرد.



حسن ختن‌لو استاد گروه مهندسی کامپیوتر دانشگاه بوعلی سینا، در سال ۱۳۸۷ دکترای مهندسی کامپیوتر خود را از دانشگاه پیر و ماری کوری اخذ و تا به حال به عنوان عضو هیئت علمی گروه مهندسی کامپیوتر، مشغول به فعالیت می‌باشد و زمینه‌های تحقیقاتی مورد علاقه ایشان، پردازش تصویر و ویدیو، پردازش تصاویر و سیگنال‌های پزشکی، سیستم‌های فازی، شناسایی الگو، یادگیری ماشین و یادگیری عمیق است.



معین پیغمبرزاده پژوهشگر و فارغ التحصیل حوزه ژئوانفورماتیک از دانشگاه پلیتکنیک میلان است. فعالیت‌های او بر پردازش و تحلیل داده‌های سه‌بعدی، ابرنقاط (Point Cloud)، سنجش‌ازدور و بینایی ماشین متمرکز است. علاقه‌مندی پژوهشی او توسعه روش‌های نوین برای استخراج اطلاعات از داده‌های مکانی و سه‌بعدی و به‌کارگیری آن‌ها در حل مسائل واقعی است. او در پروژه‌های پژوهشی و کاربردی، بر تلفیق فناوری‌های نوین با داده‌های مکانی برای دستیابی به تحلیل‌های دقیق‌تر و تصمیم‌گیری آگاهانه‌تر تمرکز دارد.

- [32] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *arXiv preprint arXiv:1803.10091*, 2018.
- [33] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5565–5573.
- [34] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
- [35] C. W. Sia, K. H. Lim, and J. T. S. Phang, "Comparative Study of Point Cloud Classification Using Deep Learning Neural Networks," in *2023 International Conference on Digital Applications, Transformation & Economy (ICDATE)*, 2023: IEEE, pp. 1–5.
- [36] J. Hu, X. Wang, Z. Liao, and T. Xiao, "M-gcn: Multi-scale graph convolutional network for 3d point cloud classification," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, 2023: IEEE, pp. 924–929.
- [37] Z. Gong, C. Kang, A. Wang, J. Wang, Y. Hu, and J.-G. Lu, "3D Point Clouds Classification and Segmentation Network Based on Global Features," in *2021 China Automation Congress (CAC)*, 2021: IEEE, pp. 1344–1348.
- [38] G. Te, W. Hu, A. Zheng, and Z. Guo, "Rgcnn: Regularized graph cnn for point cloud segmentation," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 746–754.
- [39] Y. Zhang and M. Rabbat, "A graph-cnn for 3d point cloud classification," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018: IEEE, pp. 6279–6283.
- [40] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
- [41] K. Diederik, "Adam: A method for stochastic optimization," (*No Title*), 2014.
- [42] I. K. Hassani, T. Pellegrini, and T. Masquelier, "Dilated convolution with learnable spacings," *arXiv preprint arXiv:2112.03740*, 2021.
- [43] S. Paul, Z. Patterson, and N. Bouguila, "Crossmoco: multi-modal momentum contrastive learning for point cloud," in *2023 20th Conference on Robots and Vision (CRV)*, 2023: IEEE, pp. 273–280.
- [44] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3d point cloud models," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 863–872.
- [45] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.
- [46] A. Berg, M. Oskarsson, and M. O'Connor, "Points to patches: Enabling the use of self-attention for 3d shape recognition," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022: IEEE, pp. 528–534.
- [47] N. Ye, K. Feng, and S. Lin, "Local-non-local complementary learning network for 3D point cloud analysis," *Scientific Reports*, vol. 15, no. 1, p. 480, 2025.
- [48] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new