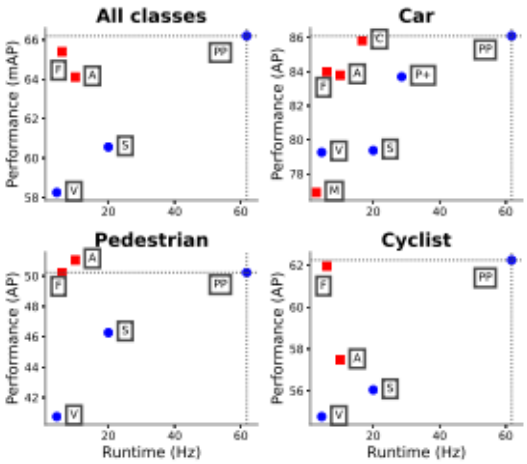
چکیده:

تشخیص شیء در ابرهای نقطه جنبه مهم بسیاری دارد از جمله برنامه های کاربردی رباتیک مانند رانندگی خودمختار در این مقاله ، مسئله رمزگذاری یک ابر نقطه در قالب مناسب برای دریافت خط لوله شناسایی را در نظر می گیریم

راهکار های اخیر دو نوع رمزگذار را نشان می دهد.رمزگذارهای ثابت تمایل دارند که سریع باشند ، اما دقت را قربانی می کنند ، در حالی که رمزگذارهایی که از داده ها آموخته اند دقیق تر هستند ، اما کندتر تر. در این جا ، ما PointPillars را پیشنهاد می دهیم ،که رمزگذار جدیدی است و از PointNets استفاده میکند که برای یادگیری از نمای ابرهای نقطه ای که در ستون های عمودی (ستون ها) سازماندهی شده اند استفاده میکند . در حالی که از ویژگی های رمزگذاری شده با معماری conv2d استفاده میکند ، ما یک شبکه کوچک را پیشنهاد میکنیم آزمایش گسترده نشان می دهد که PointPillars از رمزگذارهای قبلی با توجه به سرعت و دقت تا حد زیادی بهتر است علیرغم اینکه فقط از لیدار استفاده می کند ، خط لوله شناسایی کامل ما به طرز چشمگیری از حالت صنعتی وحتی در بین روش های ادغامی نیز فراتر است . با توجه به معیارهای D3 و چشم انداز پرندگان KITTIدر حالی که . این عملکرد آشکارسازی در صنعت با سرعت 62 هرتز کار می کند درpoint pillar بهبود زمان 2 - 4 برابرسریعترنسخه های دیگر با 105 هرتز مطابقت دارد.این معیارها نشان می دهد که PointPillars رمزگذاری مناسبی برای تشخیص شیء در ابرهای نقطه است.

مقدمه:

استقرار وسایل نقلیه خودمختار (AV) در محیط های شهری یک چالش تکنولوژیکی دشوار را ایجاد می کند. از جمله وظایف دیگر ، AV باید به ردیابی اشیاء متحرک مانند وسایل نقلیه ، پیاده و دوچرخه سوار در زمان واقعی است .برای رسیدن به این هدف ، وسایل نقلیه خودمختار به چندین حسگر متکی هستند که لیدار مهمترین آنها از این نظر است. لیدار برای اندازه گیری فاصله محیط از اسکنر لیزر استفاده میکند ، بنابراین نمای ابرنقاط کم تراکم و پراکنده ایجاد می کند.



شکل1

شکل 1. عملکرد چشم پرنده در مقابل سرعت پیشنهادی ما برای PointPillars , PP روش در مجموعه تست KITTI [5]

روش های Lidar تنها به عنوان دایره های آبی ترسیم شده اند.

روشهای lidar و بینایی به صورت مربع قرمز ترسیم شده است.

همچنین روش های برتر KITTI به عنوان liderboard یا تابلو ها رسم شده اند M : MV3D [2], A AVOD : [11], C : ContFuse [15], V : VoxelNet [33], F : Frustum PointNet [21], S : SECOND [30], P+ PIXOR++ [31].

PointPillars از لحاظ سرعت و همچنین دقت در یک حد بزرگی ، از سایر روشهای فقط لیداربهتر است. همچنین از همه روشهای ادغامی عملکرد بهتری دارد به جز عابران پیاده. عملکرد مشابه در متریک سه بعدی حاصل می شود (جدول 2) به دنبال پیشرفت های چشمگیر در روش های یادگیری عمیق برای بینایی رایانه ، تعداد زیادی از منابع پژوهشی تحقیق کرده اند که تا چه میزان می توان از این فناوری برای شناسایی شیء از ابرهای نقطه استفاده کرد [33 ، 31، 32، 11، 2، 21، 15، 30 ، 26 ، 25]. در حالی که شباهت های زیادی بین روش ها وجود دارد ، دو تفاوت کلیدی وجود دارد: 1) ابر نقطه یک نمایش پراکنده است ، در حالی که یک تصویر متراکم است و 2) ابر نقطه سه بعدی است ، در حالی که تصویر D2 است. در نتیجه ، تشخیص شیء از ابرهای نقطه ای با جزئیات به خطوط لوله های تصفیه کننده تصویر image convolutional pipelines استاندارد منتقل نمی شود.

در ابتدا خیلی از پژوهشگران بر روی استفاده از 3D convolutions [3] یا طراحی ابر نقطه در تصویر متمرکز شده اند[14] روشهای اخیر تمایل به مشاهده ابر نقطه lidar [14] از منظره چشم پرنده (BEV) دارند [2 ، 11 ، 33 ، 32].

این چشم انداز بالا چندین مزیت را ارائه می دهد. ابتدا ، BEV مقیاس شی را حفظ می کند. دوم ،حلقه ها در BEV اطلاعات مربوط به محدوده محلی را حفظ می کند.. اگر به جای آن ، حلقه هایی را در نمای تصویر انجام دهید ، اطلاعات مربوط به عمق را تار می کند (شکل 3 در [28]).

با این حال ، دید چشم پرنده بسیار پراکنده است که باعث می شود کاربرد مستقیم convolutional neural networksغیر عملی و ناکارآمد باشدراه حل مشترک یک راه حل متداول برای این مشکل ،این است که سطح زمینه را به یک شبکه منظم مثلاً 10 در 10 سانتی متر تقسیم کنید و سپس یک روش ویژگی رمزگذاری دستی را در نقاط انجام دهید

در هر سلول شبکه [2 ، 11 ، 26 ، 32]. با این حال ، چنین روشهایی ممکن است زیر حد مطلوب باشد زیرا روش استخراج ویژگی سخت کد شده ممکن است بدون تلاشهای مهندسی قابل توجهی به پیکربندیهای جدید تعمیم ندهد. برای پرداختن به این موضوعات ، و ساختن بر روی طراحی PointNet که توسط Qi و همکاران تهیه شده است. VoxelNet [33] یکی از اولین روش هایی بود که به درستی یادگیری سراسری را در این حوزه انجام داد VoxelNet فضا را تقسیم می کندبه voxels ، یک PointNet را برای هر voxelاعمال می کند VoxelNet [33] یکی از اولین روش هایی بود که به درستی یادگیری سراسری را در این حوزه انجام داد. VoxelNet فضا را تقسیم می کندبه voxels ، یک PointNet را برای هر وکسل اعمال می کند ، و پس از آن یک لایه میانی 3D convolutional برای ادغام محورعمودی ، پس از آن یک معماری تشخیص2D convolutional اعمال می شود. با این حال عملکرد VoxelNet قوی است ،زمان استنتاج ، در 4.4 هرتز ، بسیار کند است برای اجرای آن در زمان واقعی . اخیراًالگوریتم SECOND , [30] سرعت استنباطVoxelNet را بهبود بخشید اما پیچیدگی های سه بعدی همچنان یک تنگنا محسوب می شود .در این کار یا مقاله ، ما PointPillars را پیشنهاد می کنیم: روشی برای کشف اشیاء به صورت سه بعدی که یادگیری سرتاسری را تنها با لایه های دو 2D convolutionalامکان پذیر می کند.PointPillars از رمزگذار جدیدی استفاده می کند که ویژگی های موجود در ستون ها (ستون های عمودی) ابر نقطه را برای پیش بینی جعبه های سه بعدی گرا را برای اشیاء می آموزد. چندین مزیت از این رویکرد وجود دارد. اول ، با یادگیری ویژگی ها به جای تکیه بر رمزگذار ثابت ،PointPillars می تواند اطلاعات کاملی را که توسط ابر نقاط شده است ، به دست آورد. بدون نیاز به تنظیم جعبه های محاطی دستی BINNIG که در جهت عمودی وجود دارد . کار آموزش را انجام دهد. .سرانجام ، pillarsها سریع هستند زیرا تمام عملیات های کلیدی می تواند به صورت2D convolutions فرمول بندی شود که

بسیار زیاد در محاسبه GPU کارآمد است. یک مزیت دیگر از ویژگی های یادگیری این است که PointPillars برای استفاده از پیکربندی های ابر نقطه مختلف مانند چندین اسکن لیدر یا حتی ابرهای نقطه رادار ، نیازی به تنظیم دستی ندارد. ما شبکه PointPillars را در چالش های تشخیص KITTI عمومی که نیاز به تشخیص اتومبیل ، پیاده و دوچرخه سوار در BEV یا D3 دارد ارزیابی کردیم. [5] در حالی که شبکه PointPillars ما فقط با استفاده از ابرهای نقطه lidar آموزش داده می شود در وضعیت فعلی در صنعت به صورت روش هایی که از لیدار و تصاویر استفاده می کنند نیز هست. بنابراین سازماندهی استاندارد های جدیدی را برای عملکرد درهردو تشخیص چشم پرندگان BEV و سه بعدی D3 ایجاد می کند (جدول 1 و جدول 2).در همین زمان ، PointPillars با سرعت 62 هرتز کار می کند ، که 2-4 برابر سریعتر نسبت حالت قبلی صنعت است (شکل 1).

PointPillars رقابت بین سرعت و دقت بیشتر را امکان پذیر می کند. در یک تنظیم ، ما با عملکرد صنعتی بیش از 100 هرتز مطابقت داریم (شکل 5). ما همچنین کد 1 (کد مورد نظر در گیت هاب)را برای تولید نتایج خود منتشر کرده ایم.

1.1. کار مرتبط

1.1.1 کشف اشیاء با استفاده از CNN

شروع کار با Girshick و همکاران شروع می شود. [6] ، مشخص شد که معماری های شبکه عصبی حلقوی (CNN) برای تشخیص تصاویر جز آخرین پیشرفت های علمی محسوب می شوند. سلسله مقالاتی که [24 ، 7] دنبال شدند ، از رویکرد دو مرحله ای برای این مسیله دفاع می کنند. مرحله اول ، یک شبکه پیشنهادی منطقه ای . در مرحله اول ، یک شبکه پیشنهادی منطقه ای (RPN)نامزد های پیشنهادی را پیشنهاد می کند ،که قبل از طبقه بندی توسط شبکه دو مرحله ایی ، برش داده می شوند و اندازه آن قابل تغییر است. روشهای دو مرحله ای بر مجموعه داده های مهم معیار بینایی نظیر COCO [17] بر معماری های تک مرحله ایی که ابتدا توسط لیو و همکاران ارائه شده بودند ، حاکم شدند. [18]

. در یک معماری تک مرحله ای ، مجموعه ای متراکم از جعبه های لنگر ، ردیف شده و در یک مرحله به مجموعه ای از پیش بینی ها ارائه می شوند که معماری سریع و ساده ای را ارائه می دهد. اخیراً ، لین و همکاران. [16] با اطمینان اظهار داشت که با تابع lossمتمرکز پیشنهادی خود ، یک روش واحد از نظر دقت و زمان اجرا نسبت به مدلهای دو مرحله ای برتر است. در این مقاله از یک روش تک مرحله ای استفاده می کنیم

1.1.2 تشخیص شی در ابرهای نقطه برجستهlidar

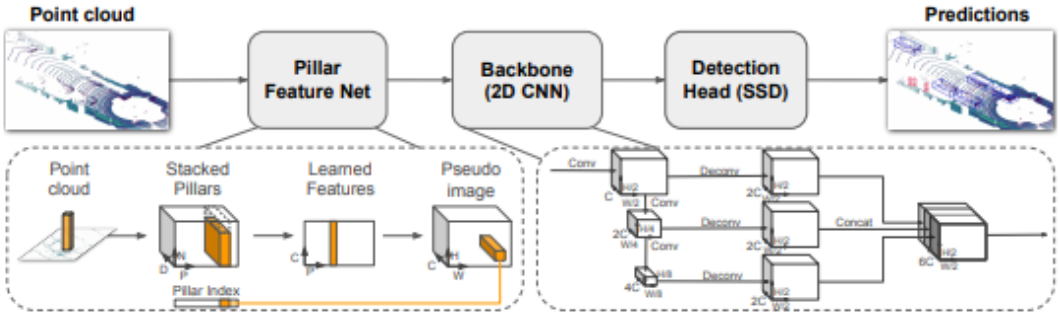
تشخیص شیء در ابرهای نقطه ای یک مسیله ذاتاً سه بعدی است. به همین ترتیب ، طبیعی است که برای شناسایی ، یک شبکهD convolutional3 را مستقر کنید ، که الگوی چندین مقاله اولیه است [3 ، 13].این روشها ضمن ارائه معماری ساده ، کند هستند. [3] برای استنتاج روی یک ابر واحد به 0.5 ثانیه نیاز دارند.اکثر روشهای اخیرزمان اجرا را بهبود می بخشد. با پیش بینی ابر نقطه سه بعدی یا روی صفحه زمین [11 ، 2] یا صفحه تصویر ،[14] .

. در رایج ترین نمونه ، ابر نقطه در وکسل ها سازماندهی می شود و مجموعه ای از وکسل ها در هر ستون عمودی به یک رمزگذاری از ویژگی های با طول ثابت ، دستکاری و رمزگذاری می شوند تا یک تصویر مجازی تشکیل می شود که می تواند بامعماری تشخیص تصویر استاندارد پردازش شود. برخی از آثار قابل ذکر شامل [2] MV3D،

[11] AVOD، PIXOR [32]و YOLO Complex [26] که از الگوی رمزگذاری ثابت یکسانی به عنوان گام اولیه از معماریهای شان استفاده می کنند

دو روش اول علاوه بر این ویژگیهای لیدار را با ویژگی های تصویر ترکیب می کند تا یک آشکار کننده یا کشف کننده چند وجهی ایجاد کند. مرحله تلفیقی که در MV3D و AVOD استفاده می شود ، آنها را مجبور می کند از خطوط لوله ردیابی دو مرحله استفاده کنند ، در حالی که PIXOR و Complex YOLO از خطوط لوله تک مرحله ای استفاده می کنند.

در کار اصلی آنها Qi et al. [22 ، 23] یک معماری ساده مطرح میشود ، PointNet ، برای یادگیری از مجموعه های نقاط نامرتب ، که مسیری برای یادگیری کامل سراسری ارائه می داد ، پیشنهاد کرد. VoxelNet [33] یکی از اولین روش های مستقر در PointNets برای تشخیص اشیاء در ابرهای برجسته است. در روش آنها ، PointNets به voxels اعمال می شود که سپس توسط مجموعه ای از لایه های convolutional 3D وبه دنبال آن ستون فقرات D2 و یك head detection انجام می شود. و بعد از آن یادگیری به پایان می رسد ، اما مانند کار قبلی که به غلبه های سه بعدی متکی بود ، VoxelNet کند است ، و برای یک ابر نقطه ای نیاز به ms225 زمان استنتاج (4.4 هرتز) دارد. روش اخیر دیگر ، Frustum PointNet [21] ، از PointNets برای تقسیم بندی و طبقه بندی ابر نقطه درذره ایجاد شده از طرح ریزی تشخیص در یک تصویر به صورت سه بعدی استفاده می کند.



نمای کلی از شبکه پیشنهادی ما 1 : feature net 2: ستون فقرات 3: head detection

Frustum PointNet در مقایسه با سایر روشهای تلفیقی عملکرد معیار بالایی را به دست آورد ، اما طراحی چند مرحله ای آن یادگیری سراسری را غیر عملی می کند. اخیراً SECOND [30] یک سری پیشرفت در VoxelNet ارائه داد که منجر به عملکرد قوی تر و سرعت بسیار بهبود یافته 20 هرتز می شود. با این حال ، آنها قادر به حذف لایه های محکم 3 بعدی گران قیمت نبودند.

1.2 دستاورد ها

• ما یک رمزگذار و شبکه ابر نقطه جدید ، PointPillars ، پیشنهاد می کنیم که روی ابر نقطه کار کند تا بتواند آموزش سراسری یک شبکه تشخیص سه بعدی اشیاء را فعال کند.• ما نشان می دهیم که چگونه می توان محاسبات روی ستون ها را در معرض متراکم convolutions 2Dکه استنباط را در 62 هرتز امکان پذیر می کند ، عاملی که 2-4 برابر سریعتر از سایر روشها.

• ما آزمایشاتی را در مجموعه داده های KITTI انجام می دهیم و وضعیت نتایج صنعتی خودروها ، پیاده ها و دوچرخه سواران را در هر دو معیار BEV و D3 نشان می دهیم

.• ما چندین مطالعه موشکفانه از کلی به جزی را برای بررسی هدفمان انجام می دهیم

2. شبکه PointPillars

PointPillars ابرهای نقطه را به عنوان ورودی قبول می کند و محور جعبه های سه بعدی را برای اتومبیل ها ، پیاده ها و دوچرخه سواران تخمین می زند. از سه مرحله اصلی تشکیل شده است (شکل 2)

(1) یک شبکه رمزگذار ویژگی که یک ابر نقطه را به یک pseudoimage پراکنده تبدیل می کند

(2) backbone convolutional 2D برای پردازش pseudoimage به نمای سطح بالا.

(3) head detects که جعبه های سه بعدی را ردیابی کرده و از آن استفاده می کند.

2.1. Pointcloud به Pseudo-Image

برای استفاده از یک معماری D convolutional2 ، ابتدا ابر نقطه را به یک pseudo-image تبدیل می کنیم . ما با یک نقطه را در ابر نقطه ای با مختصات x، y و z بیان می کنیم.

در مرحله اول ، ابر نقطه در یک شبکه مسطح در صفحه x-y گسسته می شود و مجموعه ای از ستون های P را با | B | = P

توجه داشته باشید که یک ستون یک وکسل با فضای نامحدود در جهت z است و از این رو دیگر نیازی به یک پارامتر هایپر برای کنترل برآمدگی در بعد z نیست.نقاط در هر ستون سپس با r، x_c، y_c، z_c، x_p، y_p که در آن Rبازتاب (تزئینات) قرار دارد تزئین شده است ، انیس c فاصله را با میانگین حسابی تمام نقاط ستون نشان می دهد ، واندیس p نشان دهنده انحراف ازمرکز ستون x، y (برای جزئیات طراحی به Sec 7.3 مراجعه کنید). نقطه تزئین شده درنقاط لیدار ^I اکنون D = 9 بعدی است.

در حالی که ما روی ابرهای نقطه تمرکز داریم ، ابرهای نقطه دیگری مانند رادار یا RGB-D [27] می توانند با تغییر دکوراسیون برای هر نقطه از PointPillars استفاده شوند. مجموعه ستون ها به دلیل کمبود ابر نقطه عمدتاً خالی خواهند بود و ستون های غیر خالی به طور کلی تعداد کمی نقطه در آنها خواهند بود به عنوان مثال ، در مربع های 0.162 متر مربع ابر نقطه از HDL-64E Velodyne lidar دارای ستون های k-9k6 خالی در محدوده ای است که معمولاً در KITTI درای کمبود 97٪ است .

این کمبود با تحمیل یک حد هم بر تعداد ستونهای غیر خالی در هر نمونه (P) و هم بر تعداد نقاط در هر ستون (N) برای ایجاد یک تنسور متراکم از اندازه (D، P، N) مورد سوء استخراج قرار می گیرد. اگر یک نمونه یا ستون داده های زیادی را برای قرار گرفتن یا یادگرفتن در این تنسور در اختیار داشته باشد ، داده ها به طور تصادفی نمونه برداری می شوند. برعکس ، اگر یک نمونه یا ستون از داده های خیلی کمی برای جمع کردن تانسور برخوردار باشد ، از لایه صفر استفاده می شود.

در مرحله بعد ، ما از یک نسخه ساده از PointNet استفاده می کنیم که برای هر نقطه ، یک لایه خطی به کار می رود و به دنبال آن BatchNorm [10] و ReLU [19] برای تولید یک تنسور به اندازه (C، P، N) استفاده می شود.

این کار با حداکثر عملکرد در کانالها انجام می شود تا یک تانسور خروجی اندازه (C، P) ایجاد شود. توجه داشته باشید که لایه خطی می تواند به صورت یک convolution 1x1 در سراسر تنسور تدوین وفرموله بندی شود . و در نتیجه محاسبات بسیار کارآمد انجام می شود. پس از رمزگذاری ، ویژگی ها به مکان های ستونی اصلی پراکنده می شوند تا یک pseudo-image اندازه (C، H، W) ایجاد شود که در آن H و W ارتفاع و عرض بومcanvas را نشان می دهد. توجه داشته باشید که انتخاب ما در استفاده از ستون ها به جای وکسل به ما اجازه می دهد تاd convolutions3 که دارای هزینه زیادی دارد رد شویم

backbone2.2(ستون فقرات)

ما از ستون فقرات مشابهی مثل (voxel net)[33] استفاده می کنیم و ساختار در شکل 2 نشان داده شده است. ستون فقرات دارای دو شبکه فرعی است: یکی شبکه از بالا به پایین که ویژگیهایی با وضوح مکانی خیلی کوچک را تولید می کند و شبکه دوم که باعث بالا رفتن و جمع شدن ویژگی های ازجز به کل می شود.

ستون فقرات جز به کل را می توان با مجموعه ای از بلوک های بلوک (S، L، F) مشخص کرد. . هر بلوک در گام S کار می کند (نسبت به ورودی اصلی pseudo-image اندازه گیری می شود). یک بلوک دارای L لایه ی 3X3 2D conv-layers با کانال های خروجی F است که هر کدام دارای BatchNorm و ReLU است.

اولین convolution در داخل لایه دارای گام S / Sin است تا اطمینان حاصل شود که بلوک پس از دریافت یک قطعه کوچک از ورودی از Sin ، در مسیر S کار می کند. همه convolutions های بعدی در یک بلوک دارای گام 1اند.

ویژگی های نهایی از هر بلوک از بالا به پایین از طریق بیشینه سلزی و جمع بندی به عنوان دنباله شرح زیر ترکیب می شوند.

در مرحله اول ، این ویژگی ها با استفاده از یک ترانهادهconvolution 2Dبا ویژگی های F نهایی ، از یک گام اولیه sin تا گام نهایی SoUT (با اندازه گیری مجدد wrt. تصویر اصلی شبهoriginal pseudo-image)بیشینه شده است. در مرحله بعد ، BatchNorm و ReLU روی ویژگی های بیشینه اعمال می شوند.

2.3. Detection Head

ما از تنظیمات Single Shot Detector (SSD) [18] برای انجام تشخیص شیء سه بعدی استفاده می کنیم. اگر کسی به یک کار متفاوت + (مثلاً تقسیم بندی) علاقه داشته باشد ، فقط نیاز به تعویض head detection برای head مختص برای کار مورد نظر دارد. مشابه SSD ، ما با استفاده از جعبه های محاطی و ground truth (IoU) را محسبه می کنیم [4].

جزئیات اجرا

3.1 شبکه:

به جای اینکه از قبل شبکه های ما را آموزش دهیم ، همه وزن ها بطور تصادفی با استفاده از توزیع یکسان مانند [8] در ابتدا تنظیم شدند. شبکه رمزگذاردر C = 64 دارای ویژگی های خروجی است

ستون فقرات اتومبیل و پیاده / دوچرخه سوار همان است به جز قدم اول بلوک (S = 2 برای ماشین ، S = 1 برای پیاده / دوچرخه سوار). هر دو از شبکه متشکل از سه بلوک Block1(S, 4, C), Block2(2S, 6, 2C), و Block3(4S, 6, 4C). هر بلوک با استفاده از مراحل زیر بالا بردن زیر زیر مجموعه: Up1(S, S, 2C), Up2(2S, S, 2C) , Up3(4S, S, 2C). سپس ویژگی های Up1، Up2 و Up3 با هم جمع می شوند تا ویژگی های 6c برای head detection ایجاد شود.

3.2. Loss

ما از همان توابع از دست دادن استفاده شده در SECOND استفاده می کنیم [30]..اساس جعبه هاو لنگرها توسط (x، y، z، w، l، h، θ) تعریف می شوند. مانده های رگرسیون محلی سازی بین اساس حقیقت و لنگرها توسط این موارد تعریف شده است:

L :طول w :عرض h : ارتفاع D قطر

$$\Delta x = \frac{x^{gt} - x^a}{d^a}, \Delta y = \frac{y^{gt} - y^a}{d^a}, \Delta z = \frac{z^{gt} - z^a}{h^a}$$
$$\Delta w = \log \frac{w^{gt}}{w^a}, \Delta l = \log \frac{l^{gt}}{l^a}, \Delta h = \log \frac{h^{gt}}{h^a}$$
$$\Delta \theta = \sin\left(\theta^{gt} - \theta^a\right),$$

$$d^a = \sqrt{(w^a)^2 + (l^a)^2}.$$

که در آن x^a x^{gt} به ترتیب جعبه های حقیقت زمین و لنگر هستند

که تابع کلی loss محلی هست:

$$\mathcal{L}_{loc} = \sum_{b \in (x,y,z,w,l,h,\theta)} \text{SmoothL1}\left(\Delta b\right)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \qquad (3)$$

از آنجا که زاویه تابع محلی loss نمی تواند جعبه های لنگر را تشخیص دهد ، طبقه بندی softmax classification loss یادگرفته می شود, l_{dir} از روی راه هاو مسیر های کوچک آموخته می شود [30]. تابع هزینه طبقه بندی شی استفاده می کند از تابع lossکانونی یا متمرگز شده [16]

$$\mathcal{L}_{cls} = -\alpha_a \left(1 - p^a\right)^{\gamma} \log p^a,$$

که p^a احتمال کلاس یک لنگر است تنظیمات اصلی مقاله $\alpha = 0.25$ و $\gamma = 2$. بنابراین تابع LOSS کل :

$$\mathcal{L} = \frac{1}{N_{pos}} \left(\beta_{loc}\mathcal{L}_{loc} + \beta_{cls}\mathcal{L}_{cls} + \beta_{dir}\mathcal{L}_{dir}\right),$$

جایی که N_{pos} تعداد لنگرهای مثبت و $\beta_{loc} = 2$ است $\beta_{cls} = 1$ و $\beta_{dir} = 0.2$

تابع LOSS FUNNCTION با استفاده از adam با میزان یادگیری اولیه $2*10^{-4}$ است که با ضریب کاهشی 0.8 در هر 15 EPOCH است . تعدادEPOC 160 و 320، BATCH SIZE 2 و 4 به ترتیب برای Val و Test است

3. راه اندازی آزمایشی

3.1. Datase

همه آزمایشات از مجموعه داده های معیار تشخیص شیء KITTI استفاده می کنند [5] که شامل نمونه هایی است که دارای ابرهای برجسته و تصاویر هستند. همه آزمایشات از مجموعه داده های معیار تشخیص شیء KITTI استفاده می کنند [5] ، که شامل نمونه هایی است که دارای ابرهای برجسته و تصاویر هستند. ما فقط روی ابرهای نقطه آموزش می دهیم ، اما با روش های تلفیقی که هم lidar و هم از تصاویر استفاده می شود ، هم مقایسه می کنیم. نمونه ها در ابتدا به 7481 آموزش و 7518 نمونه آزمایش تقسیم شده اند

برای مطالعات آزمایشی، مجموعه رسمی آموزش را به 3712 نمونه آموزش و 3769 نمونه validation تقسیم کردیم ، در حالی که برای ارسال آزمون ما یک مجموعه mini-val از 784 نمونه از مجموعه validationایجاد کردیم و روی

6733 نمونه باقی مانده آموزش دادیم. معیار KITTI نیاز به ردیابی خودروها ، پیاده ها و دوچرخه سواران دارد. معیار KITTI نیاز به ردیابی خودروها ، پیاده ها و دوچرخه سواران دارد. از آنجا که اشیاء داده مرجع فقط درصورتیکه در تصویر قابل مشاهده باشند ، حاشیه نویسی می شوند ، ما از قرداد های استاندارد [2 ، 33] پیروی می کنیم که فقط از نقاط برجسته استفاده می کند که در تصویر قرار می گیرند. با دنبال کردن روش نوشتجات استاندارد در KITTI [11 ، 33 ، 30] ، ما یک شبکه را برای اتومبیل ها و یک شبکه برای عابران پیاده و دوچرخه سواران آموزش می دهیم

3.2. تنظیمات

مگر اینکه به صراحت در یک مطالعه تجربی متفاوت، استفاده می کنیم وضوح XY: 0.16 متر، حداکثر تعداد ستون :(P) 12000، و حداکثر تعداد نقاط در هر ستون 10 :(N)

ما از همان لنگرها و استراتژی مطابق با [33] استفاده می کنیم. هر کلاس لنگر شرح داده شده با یک طول و عرض و ارتفاع و مرکز z هست که دارای دو زاویه 0 و 90 درجه عملی میشود لنگرها با داده های مرجع مطابقت می شوند با استفاده از iou 2d با قوانین زیر:

جعبه های محاطی با بیشرین امنتیاز در ننظر گرفته میشوند

تمام جعبه های دیگر در تابع loss نادیده گرفته می شوند. در زمان استنتاج ،ما محورترازشده (NMS)یا non maximum suppression را با آستانه همپوشانی IoU 0.5 اعمال می کنیم.

این عملکرد مشابهی را در مقایسه با NMS rotational ارائه می دهد ، اما بسیار سریعتر است

ماشین. دامنه x، y، z [(0 ، 70.4) ، (-40 ، 40) ، (-3 ، 1)]متر به ترتیب لنگر اتومبیل دارای عرض ، طول و

ارتفاع (1.6 ، 3.9 ، 1.5) متر با مرکز z از -1 متر. تطابق از آستانه های مثبت و منفی 0.6 و 0.45 استفاده می کند

پیاده و دوچرخه سوار. دامنه x، y، z به ترتیب [(0 ، 48) ، (-20،20) ، (-2.5 ، 0.5)] متر است. لنگر پیاده دارای عرض ، طول و ارتفاع (0.6 ، 0.8 ، 1.73) متر با مرکز0.6- = z متر است ، در حالی که لنگر دوچرخه سواری دارای عرض ، طول و ارتفاع (0.6 ، 1.76 ، 1.73) متر با مرکز0.6- = z متر. تطبیق از آستانه های مثبت و منفی 0.5 و 0.3 استفاده می کند

افزودن داده ها تقویت داده ها برای عملکرد خوب در معیار KITTI بسیار مهم است [30، 32، 2]. ابتدا ،پیرو SECOND [30] ، ما یک جدول جستجوی جعبه های داده های اساسی 3بعدی برای همه کلاس ها و ابرهای نقطه ارتباطی ایجاد می کنیم که در داخل این جعبه های سه بعدی قرار دارند.

سپس برای هر نمونه ، به طور تصادفی 15 ، 0 ، 8 داده های مرجع را انتخاب می کنیم نمونه ها برای اتومبیل ها ، پیاده ها و دوچرخه سواران به ترتیب وآنها را در ابر نقطه فعلی قرار دهید. ما اینها را پیدا کردیم تنظیمات برای انجام بهتر از تنظیمات پیشنهادی [30]

در مرحله بعد ، همه جعبه های داده مرجع به صورت جداگانه تقویت می شوند. هر جعبه چرخانده شده است (به طور یکنواخت از [− 20 / π، 20 / π] کشیده شده است)

و ترجمه شده است (x، y ، و z به طور مستقل از N (0 ، 0.25) گرفته شده) برای غنی سازی مجموعه آموزش. سرانجام ، ما دو مجموعه تقویت را انجام می دهیم که به طور مشترک در ابر نقطه و کلیه جعبه ها اعمال می شوند. ابتدا ، ما به صورت تصادفیmirroring flip در امتداد محور x [32] ، و سپس چرخش و scaling[30 ، 33] استفاده می کنیم. سرانجام ، ما یک افزایش داده با x، y، z را از N (0 ، 0.2) گرفته شده برای شبیه سازی نویز محلی سازی اعمال می کنیم.

Method	Modality	Speed (Hz)	mAP	Car			Pedestrian			Cyclist		
			Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D [2]	Lidar & Img.	2.8	N/A	86.02	76.90	68.49	N/A	N/A	N/A	N/A	N/A	N/A
Cont-Fuse [15]	Lidar & Img.	16.7	N/A	88.81	85.83	77.33	N/A	N/A	N/A	N/A	N/A	N/A
Roarnet [25]	Lidar & Img.	10	N/A	88.20	79.41	70.02	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN [11]	Lidar & Img.	10	64.11	88.53	83.79	77.90	**58.75**	**51.05**	**47.54**	68.09	57.48	50.77
F-PointNet [21]	Lidar & Img.	5.9	65.39	88.70	84.00	75.33	58.09	50.22	47.20	75.38	61.96	54.68
HDNET [31]	Lidar & Map	20	N/A	89.14	**86.57**	78.32	N/A	N/A	N/A	N/A	N/A	N/A
PIXOR++ [31]	Lidar	35	N/A	**89.38**	83.70	77.97	N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet [33]	Lidar	4.4	58.25	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55
SECOND [30]	Lidar	20	60.56	88.07	79.37	77.95	55.10	46.27	44.76	73.67	56.04	48.78
PointPillars	Lidar	**62**	**66.19**	88.35	86.10	**79.83**	58.66	50.23	47.19	**79.14**	**62.25**	**56.00**

Table 1. Results on the KITTI test BEV detection benchmark.

Method	Modality	Speed (Hz)	mAP	Car			Pedestrian			Cyclist		
			Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D [2]	Lidar & Img.	2.8	N/A	71.09	62.35	55.12	N/A	N/A	N/A	N/A	N/A	N/A
Cont-Fuse [15]	Lidar & Img.	16.7	N/A	82.54	66.22	64.04	N/A	N/A	N/A	N/A	N/A	N/A
Roarnet [25]	Lidar & Img.	10	N/A	**83.71**	73.04	59.16	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN [11]	Lidar & Img.	10	55.62	81.94	71.88	66.38	50.80	42.81	40.88	64.00	52.18	46.61
F-PointNet [21]	Lidar & Img.	5.9	57.35	81.20	70.39	62.19	51.21	**44.89**	40.23	71.96	56.77	50.39
VoxelNet [33]	Lidar	4.4	49.05	77.47	65.11	57.73	39.48	33.69	31.5	61.22	48.36	44.37
SECOND [30]	Lidar	20	56.69	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
PointPillars	Lidar	**62**	**59.20**	79.05	**74.99**	**68.30**	**52.08**	43.53	**41.49**	**75.78**	**59.07**	**52.92**

Table 2. Results on the KITTI test 3D detection benchmark.

4. نتایج

آنالیزمقدار

تمام نتایج تشخیص با استفاده از معیارهای رسمی ارزیابی KITTI اندازه گیری می شود که عبارتند از: چشم پرنده (BEV)، D 3، D2 و میانگین شباهت جهت گیری (AOS).

تشخیص D2 در صفحه تصویر انجام می شود و شباهت جهت گیری میانگین تشابه (اندازه گیری شده در BEV) را برای تشخیص D2میانگین جهت ارزیابی می کند. مجموعه داده KITTI در آسان ، متوسط و سخت طبقه بندی شده است ، و رتبه بندی رسمی KITTI با عملکرد بر روی محدوده ایی رتبه بندی می شود

همانطور که در جدول 1 و جدول 2 نشان داده شده است ، PointPillars از تمام روشهای منتشر شده با توجه به میانگین متوسط دقت (mAP)^2 بهتر است.

در مقایسه با روشهای lidar-only، PointPillars در تمام کلاسها و اقشار دشوار به جز طبقه بندی ماشین آسان ، نتایج بهتری کسب می کند. همچنین از روشهای مبتنی بر ادغامی در اتومبیل ها و دوچرخه سواران بهتر است. در حالی که PointPillars جعبههای سه بعدی را پیش بینی می کند ، اندازه گیری های BEV و D3 جهت گیری را در نظر نمی گیرند. جهت یابی با استفاده از AOS [5] ارزیابی می شود ، که نیاز به طرح ریزی جعبه سه بعدی در تصویر ، انجام تطبیق تشخیص D2 و سپس ارزیابی جهت گیری تطبیق می دهد .

عملکرد PointPillars در AOS در مقایسه با تنها دو روش تشخیص سهبعدی [11 ، 30] که جعبه های جهت دار را پیش بینی می کنند ، به طور قابل توجهی از همه اقشار فراتر می رود (جدول 3).

به طور کلی ، روش های تنها تصویر در تشخیص D2 بهترین عملکرد را دارند زیرا که طرح سه بعدی جعبه ها در تصویر می توانند بسته به حالت سه بعدی جعبه های بی قاعده ایجاد کنند.

. با وجود این، PointPillars متوسط AOS دوچرخه سوار از 68.16 بهتر از بهترین روش مبتنی بر تصویر است [29].

Figure 3. Qualitative analysis on KITTI. We show a bird's eye view of the lidar point cloud (top), as well as the 3D bounding boxes projected into the image for clearer visualization. Note that our method *only* uses lidar. We show ground truth (gray) and predicted boxes for car (orange), cyclist (red) and pedestrian (blue). The box orientation is shown by a line from the bottom center to the front of the box.

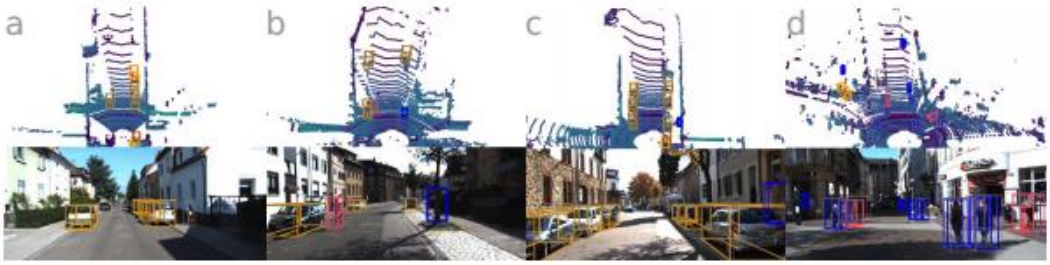


Figure 4. Failure cases on KITTI. Same visualize setup from Figure 3 but focusing on several common failure modes.

تحلیل کیفی.

در شکل 3 و 4 نتایج کیفی ارائه می دهیم. در حالی که ما فقط در ابرهای نقطه نقطه آموزش می دهیم ، برای سهولت در تفسیر ، پیش بینی های جعبه محدود 3 بعدی را از منظر BEV و تصویر تجسم می کنیم. . شکل 3 با جعبه های محدود 3 بعدی نتایج تشخیص ما را نشان می دهد.

پیش بینی های مربوط به اتومبیل ها بخصوص دقیق هستند و حالت های خرابیfalse negatives رایج شامل منفی های دروغین بر روی نمونه های دشوار (اشیاء جزئی مسدود یا دور) یا مثبت کاذبfalse positives در کلاسهای مشابه (وانت یا تراموا).

تشخیص عابر پیاده و دوچرخه سواران چالش برانگیز تر است و به برخی از حالت های جالب شکست منجر می شود. عابران پیاده و دوچرخه سواران معمولاً مانند سایرین طبقه بندی می شوند (شکل a4 را برای یک نمونه استاندارد و شکل d4 برای ترکیب عابر پیاده و جدول که به عنوان دوچرخه سواری طبقه بندی شده اند ، ببینید).

تشخیص عابر پیاده و دوچرخه سواران چالش برانگیز تر است و به برخی از حالت های جالب شکست منجر می شود. عابران و دوچرخه سواران معمولا به جای دیگیری اشتباه گرفته میشوند و دارای طبقهبندی نادرست اند (شکل A4 را برای یک مثال استاندارد و شکل برای ترکیبی از عابر پیاده و جدول طبقه بندی به عنوان یک دوچرخه سوار D4).

علاوه بر این ، عابران پیاده به راحتی با ویژگی های عمودی باریک محیط مانند ستون ها یا تنه های درخت اشتباه می گیرند (شکل b4 را ببینید). در بعضی موارد ، ما اشیاء را که در یادداشت های داده های مرجع گم شده اند ، به درستی تشخیص می دهیم (شکل c4 را ببینید).تو شکل آدمه پشت درخته ولی تشخیص داده شده است .

5. نتیجه زمان واقعی

همانطور که از نتایج ما نشان داده شده است (جدول 1 ، شکل 1 و شکل 5) ، PointPillars از نظر زمان اجرا ی استنتاج پیشرفت چشمگیری را نشان می دهد.در این بخش زمان اجرا را تجزیه می کنیم و گزینه های مختلف طراحی را که باعث افزایش این سرعت می شود در نظر می گیریم.ما روی شبکه اتومبیل تمرکز می کنیم ، اما شبکه عابر پیاده و دوچرخه با سرعت مشابهی اجرا می شود زیرا محدوده کوچکتر اثر ستون فقرات را که در مراحل پایین تر کار می کند لغو می کند.

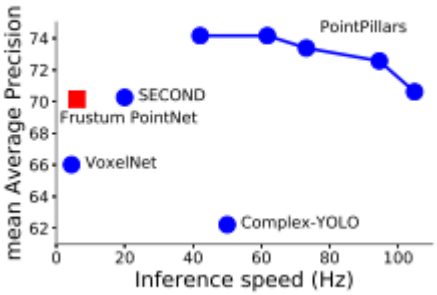
ما روی شبکه اتومبیل تمرکز می کنیم ، اما شبکه عابر پیاده و دوچرخه با سرعت مشابهی اجرا می شود زیرا محدوده کوچکتر اثر ستون فقرات را که در مراحل پایین تر کار می کند خنثی می کند.

تمام زمان های اجرا بر روی یک دسک تاپ با یک CPU Intel i7 و یک پردازنده گرافیکی ti1080 اندازه گیری می شود. مراحل اصلی استنباط به شرح زیر است. ابتدا ابر نقطه بر اساس دامنه و شفافیت در تصاویر (1.4 میلی ثانیه) بارگیری و فیلتر می شود. سپس ، نقاط در ستون ها ترتیب داده شده و تزئین می شوند (2.7 ms). در مرحله بعد ، تانسور PointPillar در (2.9 ms) GPU ،بارگذاری می شود رمزگذاری شده (1.3 میلی ثانیه) ، به صورت شبه تصویر (0.1 میلی ثانیه) پراکنده می شود و توسط ستون فقرات و سر های تشخیص (7.7 ms) پردازش می شود.
سرانجام NMS بر روی CPU (0.1 میلی ثانیه) برای کل زمان اجرا 16.2 میلی ثانیه اعمال می شود.
. رمزگذاری: طراحی اصلی برای فعال کردن این زمان ، رمزگذاری PointPilar است. طراحی اصلی برای فعال کردن زمان اجرا ، رمزگذاری PointPilar است.
اخیراً ، SECOND یک نسخه سریعتراز رمزگذار VoxelNet 50 میلی ثانیه را برای کل مدت زمان اجرای یک شبکه پیشنهاد دادآنها آنالیز زمان اجرا را ارائه ندادند ، اما از آنجا که بقیه معماری آنها شبیه به ماست ، نشان می دهد که رمزگذار هنوز هم بطور قابل توجهی کندتر است. در اجرای implementation3 ، رمزگذار به 48 میلی ثانیه نیاز دارد
جزئیات طراحی :
. ما دریافتیم که استفاده از پارامترهای کمتر بر عملکرد تشخیص تأثیر نمی گذارد. ما زمان استفاده PyTorch را با استفاده از یک PointNet در رمزگذار مان ، به جای 2 نقطه متوالی همانند [33] ، به 2.5 میلی ثانیه کاهش می دهیم
.ابعاد بلوک اول برای مطابقت با اندازه خروجی رمزگذار به 64 کاهش یافته است ، که زمان اجرا را به 4.5 میلی ثانیه کاهش می دهد. سرانجام ، با کاهش ابعاد خروجی لایه های ویژگی upsampled به نصف به 128 ، 3.9 ms دیگر را صرفه جویی می کنیم .
.TensorRT
در حالی که تمام آزمایشات ما در PyTorch انجام شده است [20] ، هسته نهایی GPU برای رمزگذاری استفاده می شود ، backbone and detection headبا استفاده از NVIDIA TensorRT ساخته شده است ، که یک کتابخانه برای استنتاج بهینه GPU است. انتقال به TensorRT ازPyTorch pipeline که با سرعت 42.4 هرتز کار می کند ، 45.5 درصد سرعترا افزایش می دهد .
نیاز به سرعت: همانطور که در شکل 5 مشاهده می شود ، PointPillars با کم شدن LOSS می تواند به 105 هرتز برسد در حالی که می توان استدلال کرد که چنین زمان اجرا بیش ازاندازه مفرط زیرا یک لیدار معمول با سرعت 20 هرتز کار می کند ، دو نکته مهم وجود دارد که باید در نظر داشته باشید. در مرحله اول ، با توجه به مصنوعات حاشیه نویسی داده های مرجع KITTI ، تنهابه صورت تقریبی 10٪از ابر نقاط اند که امتیازهای نمایش داده شده در تصویر جلویی استفاده می شود. با این حال ، یک عملیات AV باید محیط کامل را مشاهده کرده و ابر کامل نقطه را پردازش کند ، به طور قابل توجهی زمان اجرا را افزایش می دهد. دوم ، اندازه گیری زمان بندی در نوشتجات به طور معمول در یک پردازنده گرافیکی با قدرت بالا انجام می شود. با این حال ، یک عملیات AV ممکن است در عوض استفاده از GPU های جاسازی شده یا محاسبه گر جاسازی شده که احتمالاً دارای توان کمتری است انجام شود

6. مطالعات جزئی

7.1 رزولوشن فضایی :

متغیر بودن در اندازه ی سایه بان فضایی spatial binning، مبادله ای بین سرعت و دقت ایجاد می کند. ستون های کوچکتر امکان localization دقیق تر را ایجاد می کنند و منجر به ویژگی های بیشتری می شوند ، در حالی که ستون های بزرگتر به دلیل کمتر بودن ستون های غیر خالی (سرعت بخشیدن به رمزگذار) و تصویر شبه کوچکتر (سرعت بخشیدن به ستون فقرات CNN) سریعتر هستند.



نشان می دهد که اندازه صندوق bin بزرگتر منجر به شبکه های سریعتر می شود. در 0.28 ^ 2 در عملکرد مشابه با روشهای قبلی به 105 هرتزمی رسیم .
کاهش عملکرد عمدتاً به دلیل کلاس های پیاده و دوچرخه سواری بود ، در حالی که عملکرد خودرو در اندازه های جعبه یا صندوق bin پایدار بود.

7.2 افزودن داده در هر جعبه:
هر دو VoxelNet [33] و SECOND [30] افزایش هر جعبه را گسترده توصیه می کنند. با این حال ، در آزمایشات ما ،حداقل افزایش جعبه عمل کرد بهتری دارد . به طور خاص ، عملکرد آشکارسازی برای عابران پیاده به طور قابل توجهی با افزایش جعبه بیشتر کاهش یافته است. فرضیه ما این است که معرفی نمونه برداری از داده های مرجع ، نیاز به افزایش گسترده هر جعبه را کاهش می دهد
7.3 تزئین نقطه:
رمزگذار بازده های خام لیدار را بدست می آورد: x، y، z، وانعکاس r، دلتا ها را به روی مرکزخوشه اضافه می کند ستون نقطه شکل 5.
عملکرد تشخیص BEV (MAP) در مقابل سرعت (هرتز) بر روی مجموعه یKITTI[5] VAL در سراسر پیاده، دوچرخه و اتومبیل. دایره های آبی فقط روش های lidar یا برجسته را نشان می دهند ، مربع های قرمز نشان دهنده روش استفاده از lidar و vision یا بینایی هستند. .
نقاط کاربردي مختلف با استفاده از اندازه هاي شبکه ستون بدست آمدm2 { 0.282 , 0.242 , 0.2 2 , 0.162 , 0.122}
و حداکثر ستونها به ترتیب 16000 ، 12000 ، 12000 ، 8000 ، 8000.
(cxc، Δyc، czc) (همانطور که در VoxelNet [33] انجام شد) و فاصله از مرکز ستون (pxp، Δyp) (سهم ما).
انحراف ستون ها (Δyp، xp Δ) محل نقطه را در سیستم مختصات محلی هر ستون رمزگذاری می کند. آنها از نقاط دیگر مستقل هستند و بنابراین فضای محلی نقاط را به شکلی استاندارد می کنند که مکمل2D convolutionsدر x و y باشد.
ما ستون z را منحرف نکردیم زیرا این یک انحراف ثابت برای تمام نقاط است
در حالی که جبران خسارات خوشه ای (Δ xc، Δyc، czc) روش دیگری برای استاندارد سازی محتوا ی محلی نقاط ارائه می دهد ، نیاز به محاسبه آماری خلاصه دارد و از این رو وابستگی بین نقاط ایجاد می شود.
افزایش داده ها و نمونه برداری از نقاط در یک ستون ، مرکز خوشه را تغییر می دهد ، این امر منجر به واریانس بالاتر در هنگام آموزش فقط با انحراف های خوشه ای می شود و نه انحراف ستون را جبران می کند.
7.4. Encoding

x, y, z	r	x_c, y_c, z_c	x_p, y_p	BEV mAP	Δ mAP
✓				66.6	-6.0
✓	✓			70.5	-2.1
✓	✓	✓		70.4	-2.2
✓	✓		✓	71.4	-1.2
✓	✓	✓	✓	**72.6**	0.0

Table 4. Ablation study for encoder point decorations. The lidar sensor outputs the spatial location, x, y, z, and reflectance r, of each lidar return. This can be supplemented with the cluster center offset (Δx_c, Δy_c, Δz_c) or pillar center offset (Δx_p, Δy_p). The best detection performance uses all this information.

Encoder	Type	0.16^2	0.20^2	0.24^2	0.28^2
MV3D [2]	Fixed	72.8	71.0	70.8	67.6
C. Yolo [26]	Fixed	72.0	72.0	70.6	66.9
PIXOR [32]	Fixed	72.9	71.3	69.9	65.6
VoxelNet [33]	Learn	**74.4**	**74.0**	**72.9**	71.9
PointPillars	Learn	73.7	72.6	**72.9**	**72.0**

Table 5. Encoder performance evaluation. To fairly compare encoders, the same network architecture and training procedure was used and only the encoder and xy resolution were changed between experiments. Performance is measured as BEV mAP on KITTI val. Learned encoders clearly beat fixed encoders, especially at larger resolutions.

برای ارزیابی تأثیرگذاری رمزگذاری PointPillar به صورت مجزا ، چندین رمزگذار را در پایگاه رسمی SECOND اجرا کردیم [30]. برای جزئیات بیشتر در مورد هر کدگذاری ، ما به مقالات اصلی مراجعه می کنیم.

همانطور که در جدول 5 نشان داده شده است ، یادگیری رمزگذار ویژگی کاملاً نسبت به رمزگذار ثابت در همه دقت ها برتر است. انتظار می رود این امر به عنوان موفقیت آمیزترین معماریهای یادگیری عمیق سراسری انجام شود. علاوه بر این ، اختلافات با اندازه صندوق بزرگتر افزایش می یابد که در آن فقدان قدرت بیانگر رمزگذار ثابت به دلیل وجود ابر بزرگتر در هر ستون ، برجسته می شود.

در میان رمزگذارهای آموخته شده VoxelNet از PointPillars قوی تر است. با این حال ، هنگامی که مقایسه برای مدت استنتاج مشابه انجام شود ، مشخص است که PointPillars یک نقطه کار بهتری را ارائه می دهد (شکل 5).

چند نکته جالب در جدول 5 وجود دارد. اول ، علیرغم یادداشت هایی که در مقالات اصلی وجود دارد که رمزگذار آنها فقط روی اتومبیل کار می کند ، دریافتیم که رمزگذارهای MV3D [2] و PIXOR [32] می توانند عابران پیاده و دوچرخه سواران را به خوبی یاد بگیرند.

. دوم ، پیاده سازی های ما نتایج منتشر شده مربوطه را با حاشیه زیادی جستجو میکنند (1 - 10 نقشه). اگرچه این یک مقایسه مستقیم نیست ، زیرا ما فقط از رمزگذارهای مربوطه استفاده کرده ایم و نه از معماری های کامل شبکه ، تفاوت عملکرد قابل توجه است. ما چندین دلیل احتمالی را می بینیم. در مورد VoxelNet و SECOND ما گمان می کنیم که افزایش عملکرد ناشی از افزایش بیش از حد data augmentation hyperparameters پارامترهای تقویت داده است ، همانطور که در بخش 7.2 بحث شده است. . در میان رمزگذارهای ثابت ، تقریباً نیمی از افزایش عملکرد را می توان با معرفی نمونه برداری از پایگاه داده دادهای مرجع توضیح داد [30] ، که ما پیدا کردیم که mAP را تقریباً 3٪ mAP تقویت می کند.

اختلافات باقیمانده به دلیل ترکیبی از هایپرپارامرهای متعدد از جمله طراحی شبکه (تعداد لایه ها ، نوع لایه ها ، استفاده از هرم ویژگی feature pyramid) به احتمال زیاد ناشی می شود. طراحی جعبه لنگر (یا فقدان آن [32])؛ loss محلی سازی با توجه به D3 و زاویه؛ از دست دادن طبقه بندی؛ classification loss; گزینه های بهینه ساز (SGD در مقابل آدم ، اندازه دسته)؛ و بیشتر. با این حال ، مطالعه ای دقیق تر برای جداسازی هر علت و معلولی مورد نیاز است.

8- نتیجه گیری.

در این مقاله ، ما PointPillars ، یک شبکه عمیق و رمزگذار جدید را معرفی می کنیم که می تواند از طریق ابرهای نقطه لیدار به صورت سراسری آموزش یابد. ما نشان می دهیم که در چالش KITTI، PointPillars با ارائه عملکرد تشخیص بالاتر (BEV و نقشه سه بعدی) با سرعتی بیشتر بر تمام روشهای موجود حاکم است. نتایج ما نشان می دهد که PointPillars بهترین معماری را تا کنون برای کشف شیء D3 از lidar ارائه می دهد.

.