دانشکده فنی مهندسی

پایان‌نامه‌ی پروژه دوره کارشناسی

در رشته مهندسی کامپیوتر

گرایش نرم افزار

عنوان پایان‌نامه

تشیخص حرکت با Deep Learning

استاد راهنما:

جناب آقای دکتر رضا روحانی


بسم الله الرحمن الرحيم

دانشکده فنی مهندسی

پایان‌نامه‌ی پروژه دوره کارشناسی

در رشته مهندسی کامپیوتر

گرایش نرم افزار

عنوان پایان‌نامه

برآورد انسانی با یادگیری عمیق

استاد راهنما:

جناب آقای دکتر رضا روحانی

توسط: علی امانی

آذر ماه ۱۳۹۸

پایان نامه آقای علی امانی جهت اخذ درجه کارشناسی رشته مهندسی کامپیوتر گرایش نرم افزار با عنوان: برآورد انسانی با یادگیری عمیق در تاریخ..... با حضور هیأت داوران زیر بررسی و با نمره..... مورد تصویب نهایی قرار گرفت.

۱- استاد راهنما جناب آقای دکتر رضا روحانی با مرتبه علمی استادیار

امضاء

۲. استاد داور با مرتبه علمی

امضاء

مسؤولیت کلیه مطالبی که در این پایان نامه آورده شده است به عهده نگارنده بوده و دانشکده فنی و مهندسی هیچ مسئولیتی را در این زمینه تقبل نمی‌نماید.

دکتر شهلا نعمتی

مدیر گروه مهندسی کامپیوتر

چکیده

یادگیری عمیق (Deep Learning) در طیف گسترده‌ای از صنایع شامل فناوری‌های پزشکی و محصولات مصرفی کاربرد دارد.در این متن می‌خواهیم یکی از روش‌های تشخیص برآورد انسانی مبتنی بر تصویر را بررسی کنیم. در این روش به وسیله‌ی دوربین Kinect از شخص مورد نظر دو ویدیو RGB گرفته می‌شود و به وسیله‌ی آن مفاصل[1] بدن شخص بدست آورده می شود.سپس با یادگیری عمیق مفاصل دو بعدی را از این دو ویدیو به دست می آوریم و با استریو[2] مفاصل سه بعدی به دست می آید.پس از آن با یادگیری عمیق مفاصل سه بعدی را به دست می آوریم. در آخر بین این دو روش در برآورد انسانی مقایسه ای انجام می گیرد و نتیجه ی حاصل از استریو با متلب رسم می شود.شایان ذکر است که در برخی نقاط حاصل از استریو خطا دیده می شود.

[1] keypoint
[2] sterio

فهرست

فهرست تصاویر

فهرست جداول

۱ فصل اول

مقدمه

هدف از این پروژه برآورد انسانی با روش یادگیری عمیق است.برآورد انسانی با پروژه ی FaceBookResearch VideoPose3D به صورت سه بعدی انجام می شود.برای اجرای این پروژه روی فیلم ها نیاز به نصب لینوکس،caffe و پروژه ی Detectron داریم.برآورد انسانی با پروژه ی OpenPose به صورت دو بعدی انجام می شود.این پروژه روی پردازنده های گرافیکی انویدیا و AMD اجرا شدنی است. از دو دوربین موازی برای برآورد انسانی به صورت دو بعدی استفاده می شود.هم چنین نمایش اسکلت بدن با استفاده از این نقاط دو بعدی و cv2 رسم می شود.سپس استریو به روش مثلث سازی انجام می شود تا بتوانیم نتیجه ی برآورد انسانی به صورت سه بعدی را نیز به دست آوریم.هم چنین مختصات نقاط حاصل از استریو با متلب رسم می شود.سپس نتیجه ی برآورد انسانی حاصل از این دو روش با هم مقایسه می شود.

۱.۱ مقایسه ی یادگیری عمیق با کینکت و سنسور

۱.۱.۱ سنسور

سنسورهای مورد استفاده برای به دست آوردن مفاصل بسیار دقیق ولی گران هستند.هم چنین نیاز به نصب این سنسور در هر مورد اجرای برآورد انسانی است.

۱.۱.۲ کینکت

دوربین کینکت ورژن 1 از فرو سرخ استفاده می کند.این امواج از پلاستیک رد نمی شوند.دقت پایین تری از یادگیری عمیق دارد و گاهی در برآورد تصاویر با پرش مواجه می شویم ولی پرش در اسکت انسانی پروژه ی سه بعدی وجود ندارد.کینکت 1 می تواند 2 نفر را رد گیری کند و کینکت 2 می تواند بیش از 6 نفر را رد گیری کند.فضای اطراف کینکت در برآورد اسکت بدن موثر است.محدود به فضای داخل است.کینکت 1فاصله ی 0.4-4.5 متر را ارزیابی می کند.کینکت 2 فاصله ی 0.5-8 متر را ارزیابی می کند.

۱.۲ مقایسه ی کینکت ۱ و کینکت ۲

در جدول زیر اطلاعات مر بوط به مقایسه ی دوربین های کینکت را مشاهده می کنید:

جدول ۱ مقایسه ی دوربین های کینکت

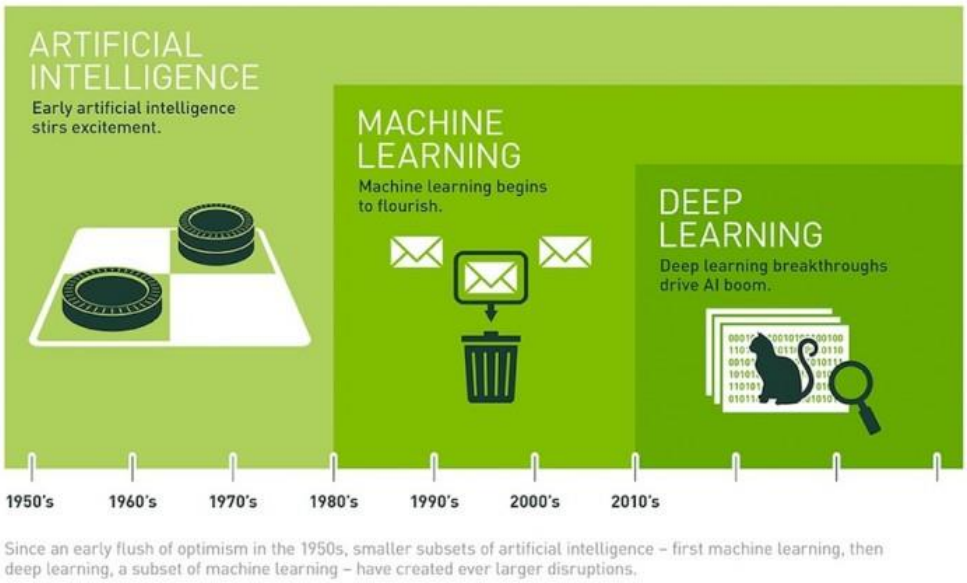
ویژگی	کینکت 1	کینکت 2
دوربین رنگی	640*480 30@fps	**1920*1080@30fps**
حداکثر عمق	~4.5 m	**~4.5 m**
حداقل عمق	40 cm	**50 cm**
محدوده ی دید افقی	57	**70**
محدوده ی دید عمودی	43	**60**
تعداد مفاصل قابل تعریف	20	**26**
تعداد اسکلت بدن قابل ردیابی	2	**6**
USB	2.0	**3.0**

۲ فصل دوم

معرفی یادگیری عمیق

۲.۱ مقدمه

یادگیری عمیق زیرمجموعه‌ای از یادگیری ماشین است. یادگیری عمیق با کمک الگوریتم‌های شبکه‌ی عصبی مصنوعی که از مغز انسان الهام گرفته شده است از مجموعه‌ای بزرگ از داده‌ها یاد می‌گیرد. یادگیری عمیق و تمام جنبه‌های هوش مصنوعی مدرن از داده‌ها استفاده می‌کنند تا تصمیم‌های مشابه هوش انسانی بگیرند. همچنین به رایانه‌ها آموزش می‌دهد که از داده‌ها برای یادگیری ساده بهره ببرند. کاربرد یادگیری عمیق را می‌توان در خودروهای بدون‌راننده برای تشخیص خودروهای دیگر، چراغ راهنمایی و علائم راهنمایی و رانندگی و حتی عابران پیاده دید. از سوی دیگر مدل‌های یادگیری عمیق در محصولات مصرفی همچون دستیاران هوشمند صوتی، تشخیص چهره، گوینده‌های هوشمند کاربرد دارند.

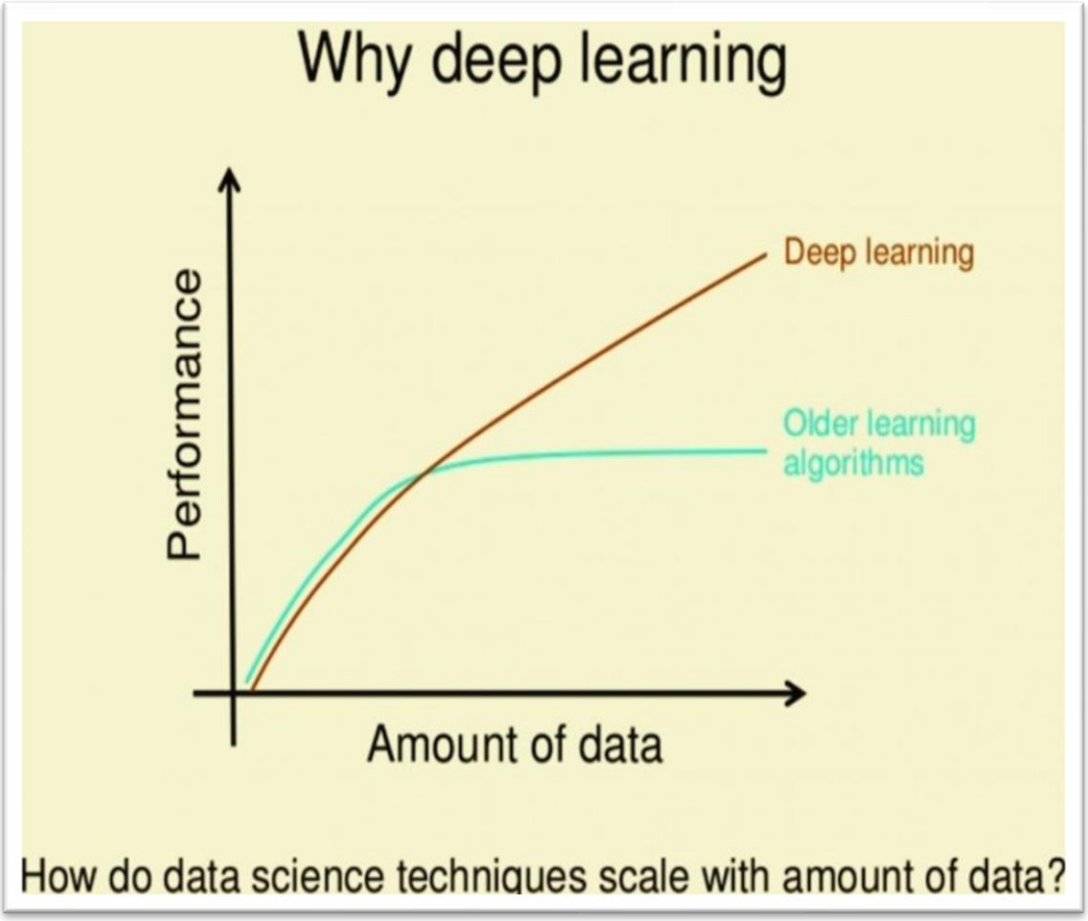


تصویر ۱ یادگیری عمیق

۲.۲ چرا اهمیت یادگیری عمیق رو به افزایش است؟

قلب یادگیری عمیق را می‌توان داده دانست. شما با کمک تمرین و تجربه مهارت‌های جدید می‌آموزید. مدل‌های یادگیری عمیق نیز کاری مشابه شما برای یادگیری انجام می‌دهند. یک خودروی بدون راننده را در نظربگیرید. یک مدل رایانه‌ای برای اینکه بتواند تابلوی ایست (STOP) را تشخیص بدهد باید با هزاران تابلوی ایست آموزش داده شود. مدل‌های رایانه‌ای یادگیری عمیق یاد

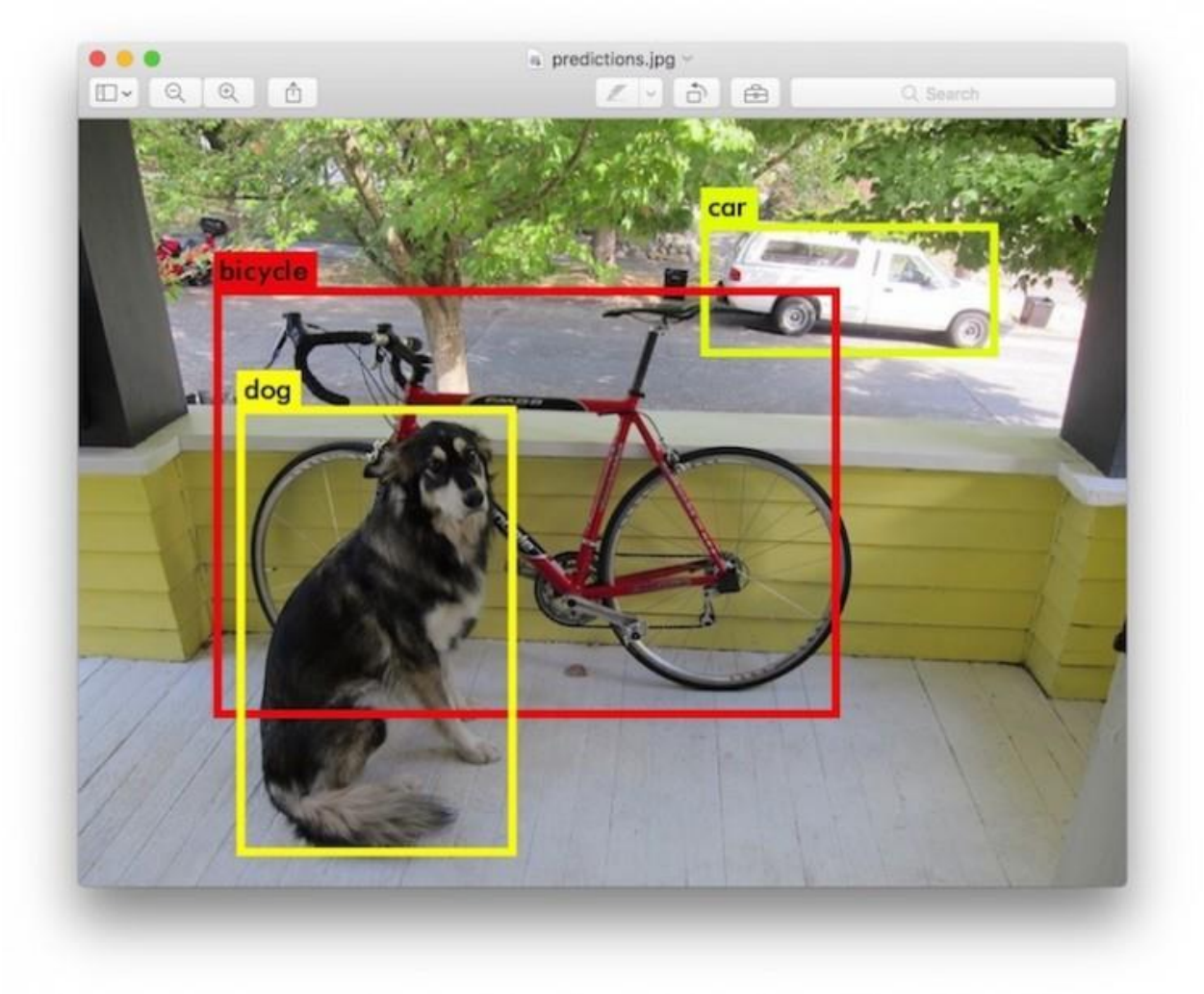
می‌گیرند تا وظایف طبقه‌بندی تصاویر، متن‌ها و یا صداها را به طور دقیق انجام دهند. این مدل‌ها همانند شبکه‌های عصبی مصنوعی آموزش می‌بینند که از مجموعه‌ای بزرگ از داده‌ها استفاده کنند.



تصویر ۲ کارایی

با مدل‌های یادگیری عمیق سطح دقت عملکرد الگوریتم‌ها بسیار زیاد خواهد شد به طوری‌که در وظایفی همچون طبقه‌بندی تصاویر، رایانه از انسان پیشی می‌گیرد. روش‌های یادگیری عمیق در شبکه‌های عصبی مصنوعی به کار می‌روند. شبکه‌ی عصبی مصنوعی مجموعه‌ای از الگوریتم‌ها است که تلاش می‌کند رابطه‌ی میان داده‌های ورودی و یک فرآیند را شناسایی کند. این روند بسیار مشابه عملکرد مغز انسان در پیدا کردن ارتباط میان داده‌های دریافتی از محیط است. شبکه‌های عصبی مصنوعی الگوهای موجود در داده‌های ورودی را استخراج و بهترین نتیجه را مشخص می‌کند.

یادگیری عمیق با ماشین‌ها این امکان را می‌دهد تا مسائل پیچیده را حتی با داده‌های بدون ساختار و متنوع حل کنند. هرچه بیشتر یادگیری عمیق یاد بگیرد عملکرد بهتری در حل مسئله خواهد داشت.



تصویر ۳ دقت یادگیری عمیق

روش‌های یادگیری عمیق از لایه‌های پنهان شبکه‌های عصبی مصنوعی بهره می‌برند. پیش از این شبکه‌های عصبی مصنوعی تنها ۲ تا ۳ لایه‌ی پنهان داشتند. هم‌اکنون تعداد لایه‌های پنهان یادگیری عمیق به ۱۵۰ لایه می‌رسد. پس از مدل کردن مسئله‌ی موردنظر، ابزار موردنیاز یادگیری عمیق است تا خروجی مدل همانند روند تصمیم‌گیری مغز انسان به‌دست آید. یادگیری عمیق بخشی از یادگیری ماشین است به طوری‌که هرلایه، ویژگی خاص و اطلاعات مفیدی را از داده‌ها استخراج می‌کند. دلیل استفاده از عبارت یادگیری عمیق این است که شبکه های عصبی لایه های مختلف یا عمیقی دارند که یادگیری را ممکن می سازد.

حجم این داده به لطف گسترش شبکه اینترنت و ابزارها و پلتفرم های ارتباطی هر روزه در حال افزایش است و اخیرا به ۲.۶ کوینتیلیون (۱۰به توان ۱۸) بایت در روز رسیده است. یادگیری عمیق به حجم داده‌ی بسیاری نیاز دارد تا بتواند از داده‌ها یاد

بگیرد. تولید انبوهی از داده‌ها یکی از دلایل رشد قابلیت‌های یادیگیری عمیق است. افزون بر این، پیشرفت توان محاسباتی نیز در به‌کارگیری هرچه بیشتر یادیگری عمیق موثر بوده است.

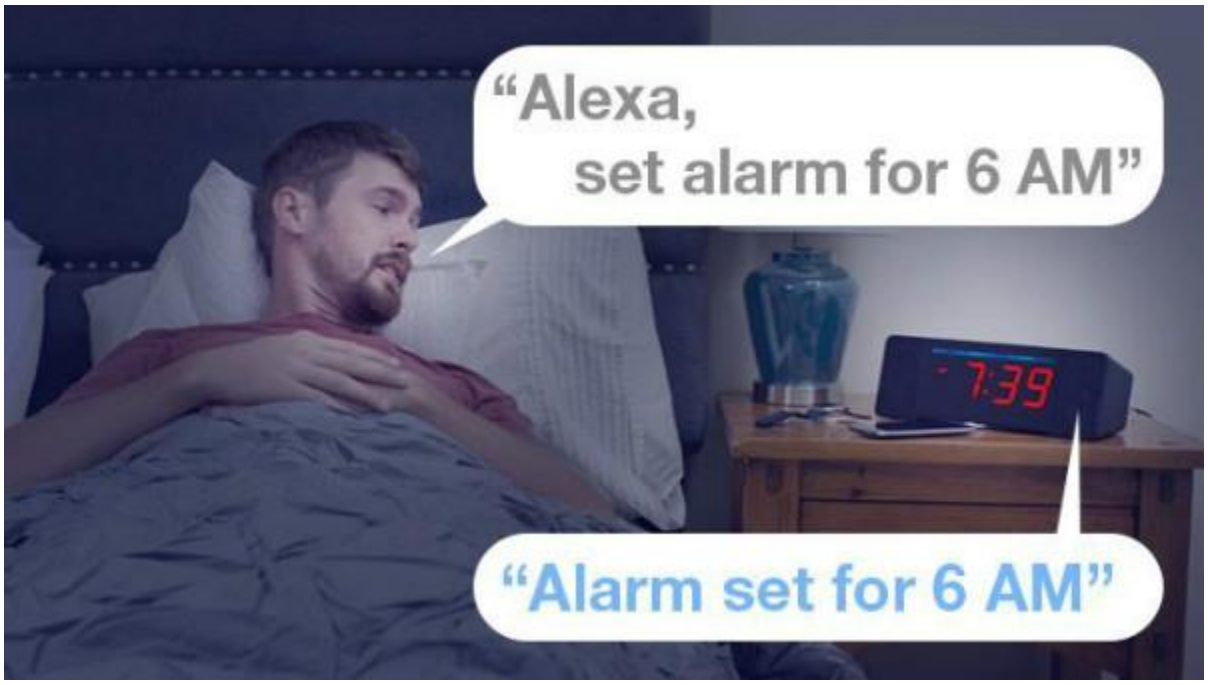
مغز انسان را می‌توان همانند یک شبکه عصبی شامل میلیون‌ها گره‌ی پردازشی ساده مدل کرد به‌گونه‌ای که با چگالی بسیاری در کنار یکدیگر قرار گرفتند. هر گره به تعدادی گره در لایه‌ی پیشین متصل است و داده را از آن‌ها دریافت می‌کند و همچنین برای ارسال داده به تعدادی گره در لایه‌ی بعدی متصل است.

۲.۳ کاربرد روش‌های یادگیری عمیق

مدل‌های یادگیری عمیق در دنیای امروز نفوذ کرده‌اند. از فناوری‌های الکترونیکی تا هوا و فضا و صنایع دفاعی کاربرد دارد. این مدل‌ها در برنامه‌های کاربردی ترجمه‌ی شنیداری و گفتاری استفاده می‌شوند. برنامه‌ها صدای شما را تشخیص می‌دهند و متناسب با صحبت شما پاسخ می‌دهند. در فناوری‌های پزشکی برای تشخیص سلول‌های سرطانی نیز کاربرد دارند. حتی برخی از صنایع با کمک پیشرفت یادگیری عمیق به بهبود کیفیت زندگی کارکنان خود می‌پردازند. آن‌ها مشخص می‌کنند کدام یک از کارکنان در هنگام استفاده از ماشین‌آلات سنگین در خطر آسیب هستند.

۲.۳.۱ دستیار مجازی

دستیارهای مجازی مانند الکسا، کورتانا و سیری برای درک گفتار و زبان انسان و تعامل با انسان از یادگیری عمیق بهره می‌برند.



تصویر ۴ الکسا

۲.۳.۲ مترجم زبانی

یادگیری عمیق زبان‌های مختلف را به یکدیگر ترجمه می‌کند. برنامه‌های کاربردی ترجمه برای گردشگران و مسافران بسیار کاربردی است. یکی از نمونه های خاص در این زمینه اپلیکیشن ترجمه گوگل است که قابلیت برگردان متون درون تصویر را هم دارد. مدیر واحد یادگیری عمیق گوگل دو سال پیش مدعی شد که در 10 سال آینده هدفون ها هر زبانی را ترجمه خواهند کرد.

۲.۳.۳ خودروهای بدون راننده تحویل‌دهنده، پهپادها و خودروهای خودران

توانایی درک موانع موجود در مسیر حرکت، علایم راهنمایی و رانندگی و دیگر خودروها توسط خودروهای خودران از یادگیری عمیق نتیجه می‌شود. الگوریتم‌های یادگیری عمیق هرچه بیشتر داده دریافت کنند عملکرد بهتری در پردازش اطلاعات خواهند داشت.

۲.۳.۴ ربات‌های گفت‌وگو (chat bot)

ربات‌های گفت‌وگو خدمات مشتری را برای بسیاری از شرکت‌ها فراهم می‌کنند. به لطف یادگیری عمیق، این چت‌بات‌ها می‌توانند به صورت هوشمندانه پاسخگوی پرسش‌های متنی و صوتی باشند.

تصویر ۵ خودروهای خودران

۲.۳.۵ تشخیص چهره

یادگیری عمیق با توانایی تشخیص چهره نه تنها در کاربردهای امنیتی استفاده می‌شود بلکه در شبکه‌های اجتماعی مانند فیسبوک نیز به کار می‌رود. در فیسبوک با الگوریتم‌های یادگیری عمیق نام افراد با کمک تشخیص چهره بر پست‌های فیسبوک برچسب زده می‌شود. آنچه که هنوز در موضوع تشخیص چهره برای یادگیری عمیق چالش به شمار می‌آید، شناخت چهره با تغییر مدل مو و ریش و حتی تصاویر با کیفیت پایین و در نور کم است.

۲.۳.۶ پزشکی و داروسازی

از تشخیص بیماری تا تولید دارو براساس ژنوم هر فرد از کاربردهای یادگیری عمیق در پزشکی و داروسازی است که بسیار موردتوجه شرکت‌های داروسازی و حوزه‌ی پزشکی قرار گرفته است.

تصویر ۶ داروی سرطان

۲.۳.۷ رنگ‌آمیزی تصویر

تبدیل تصویرهای سیاه و سفید به تصویرهای رنگی از توانایی های یادگیری عمیق است. الگوریتم‌های یادگیری عمیق با تشخیص زمینه‌ی تصویر و اشیای موجود در تصویر می‌توانند تصویر سیاه و سفید را با دقت بسیار به تصویر رنگی تبدیل کنند.

۲.۳.۸ خرید و تفریح شخصی‌سازی شده

تارنمای شرکت تولید و پخش مجموعه‌های تلویزیونی و فیلم‌های سینمایی با کمک یادگیری عمیق قادر است بر اساس علاقه‌های شخص به او فیلم سینمایی و سریال پیشنهاد دهد. تارنمای آمازون نیز به هر مشتری خود با توجه به علایق و نیازهای او محصولاتی را پیشنهاد خواهد داد.

۲.۳.۹ درک احساسات

ابزاری که بتواند حالت روحی و یا احساسات یک فرد را مبتنی بر ویدئوی چهره و یا صدای وی تشخیص دهد می‌تواند یک ابزار کاربردی در زمینه‌های مختلف باشد. یادگیری عمیق توانسته است تا حدودی این موضوع را به واقعیت نزدیک نماید. به عنوان مثالی از کاربرد این ابزار می‌توان به تشخیص احساسات افراد پس از مشاهده یک تبلیغ اشاره نمود که با پردازش آن می‌توان در مورد اثرگذاری تبلیغ اظهار نظر نمود.

۲.۳.۱۰ امنیت فضای سایبری

شرکت‌های بسیاری به دلیل ضعف‌های امنیتی و دسترسی هکرها به اطلاعات کاربرانشان هزینه‌های گزافی متحمل شده و یا به کلی از بین رفته‌اند. یادگیری عمیق در این زمینه می‌تواند با استفاده از تشخیص الگوی ویروس‌ها یا اصطلاحا امضای آن‌ها امنیت فضای سایبری را تضمین نماید. علت اینکه آنتی‌ویروس لازم است همواره به روز باشد نیز در واقع همین است که بتواند امضای ویروس‌های مختلف را به دست بیاورد و از این طریق امنیت فضای سایبری را بالا ببرد. در حال حاضر استارت‌آپ‌هایی وجود دارند که بر روی ساخت آنتی‌ویروس‌های زنده مبتنی بر یادگیری عمیق کار می‌کنند. این آنتی‌ویروس‌ها به گونه‌ای هستند که با استفاده از تشخیص الگو و بدون نیاز به اینترنت پایگاه داده امضای ویروس‌ها را به روز می‌کنند و در نتیجه حتی قادر خواهند بود امضای برخی از ویروس‌ها را پیش‌بینی نمایند و از این طریق امنیت فضای سایبری را بالا ببرند.

۲.۳.۱۱ اقتصاد

اگر بخواهیم به یکی از شرکت‌های فعال در این زمینه اشاره کنیم می‌توانیم به signalfire اشاره نماییم. این شرکت که قصد خرید تعدادی استارت‌آپ را داشت، به منظور تعیین آن‌ها از یک شبکه عصبی عمیق استفاده نمود و مبتنی بر تصمیم این شبکه استارت‌آپ‌ها را خریداری نمود.

۲.۴ روش های یادگیری عمیق

یادگیری عمیق بهترین راه برای پردازش حجم انبوهی از داده‌ها است. بیشتر اوقات، این فرایند از طریق «یادگیری تحت نظارت» انجام می‌شود که در این روش، تنها داده‌های از پیش مشخص شده مورد پردازش قرار می‌گیرند.

۲.۴.۱ یادگیری بدون نظارت

در این روش، افراد داده‌ها را دسته‌بندی نمی‌کنند. در عوض، متخصصین حجم انبوهی از داده‌ها را روانه سیستم‌ها می‌کنند و این سیستم‌ها از طریق پیدا کردن الگوها، می‌توانند این موارد را دسته‌بندی کنند.

۲.۴.۲ یادگیری تقویت شده

در این روش، سیستم اگر بتواند به هدف موردنظر دست پیدا کند پاداش می‌گیرد و در غیر این صورت مجازات می‌شود.
در این روش مثلا وقتی که یک سیستم موقعیت مقاله را در صفحه اصلی سایت مشخص می‌کند، در صورت کلیک کاربر بر مقاله مذکور، پاداش دریافت می‌کند و کلیک نکردن هم مساوی با مجازات سیستم است.به همین خاطر سیستم یاد می‌گیرد که انواع و اقسام مقاله‌ها در کدام بخش سایت قرار دهد.

2.4.3 شبکه‌های رقابتی

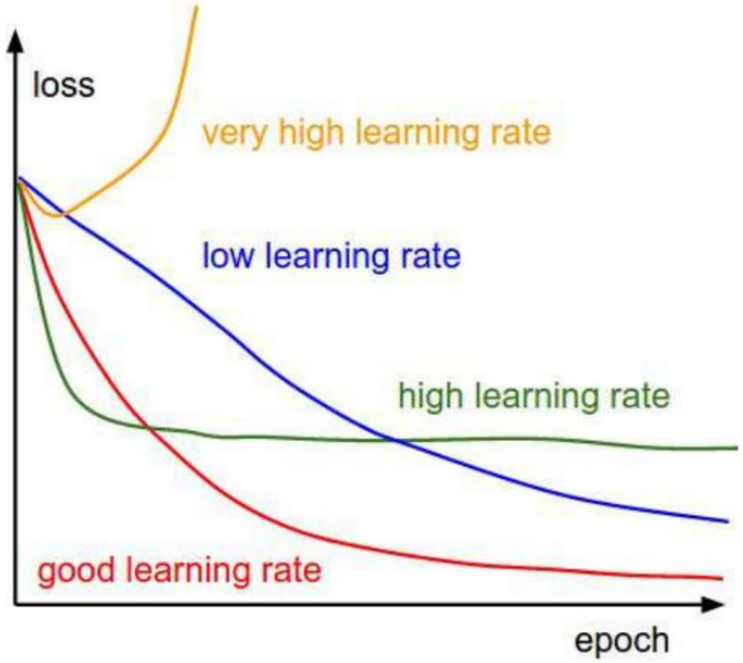
در این نوع سیستم‌ها، دو نوع هوش مصنوعی مختلف به رقابت با یکدیگر می‌پردازند و یکی از آن‌ها مرتبا داده‌های تقلبی ایجاد می‌کند و سیستم دیگر باید این داده‌های جعلی را تشخیص بدهد. به‌عنوان مثال، فرض کنید که یک الگوریتم ویدیوهای تقلبی از افراد مشهور را ایجاد می‌کند و یک الگوریتم دیگر وظیفه شناسایی این ویدیوهایی جعلی را برعهده دارد. هرچند چنین فرایندی می‌تواند هوش مصنوعی باهوش و خلاقی را به ارمغان بیاورد، اما پتانسیل زیادی هم برای سوءاستفاده دارد و مخاطرات آن نباید نادیده گرفته شود.

۲.۵ توضیح چند اصطلاح مهم در یادگیری عمیق

۲.۵.۱ نسبت یادگیری[۳]

سرعت کاهش هزینه‌ها همان نسبت یادگیری است. نسبت یادگیری باید به اندازه‌ای زیاد نباشد که حالت بهینه را رد کنیم و به اندازه‌ای کم نباشد که یادگیری شبکه زمان زیادی ببرد.

[3] Learning rate



تصویر ۷ نسبت یادگیری

۲.۵.۲ بسته[۴]

به جای ارسال کل ورودی، آن را به بسته‌های کوچک با اندازه‌ی یکسان تقسیم می‌کنیم.

۲.۵.۳ دوره[۵]

یک دوره برابر با یک رفت و برگشت ورودی در کل شبکه است. اگر دوره ها زیاد باشند،باعث دقت بیشتر شبکه می‌شود ولی زمان یادگیری شبکه هم افزایش می‌یابد.

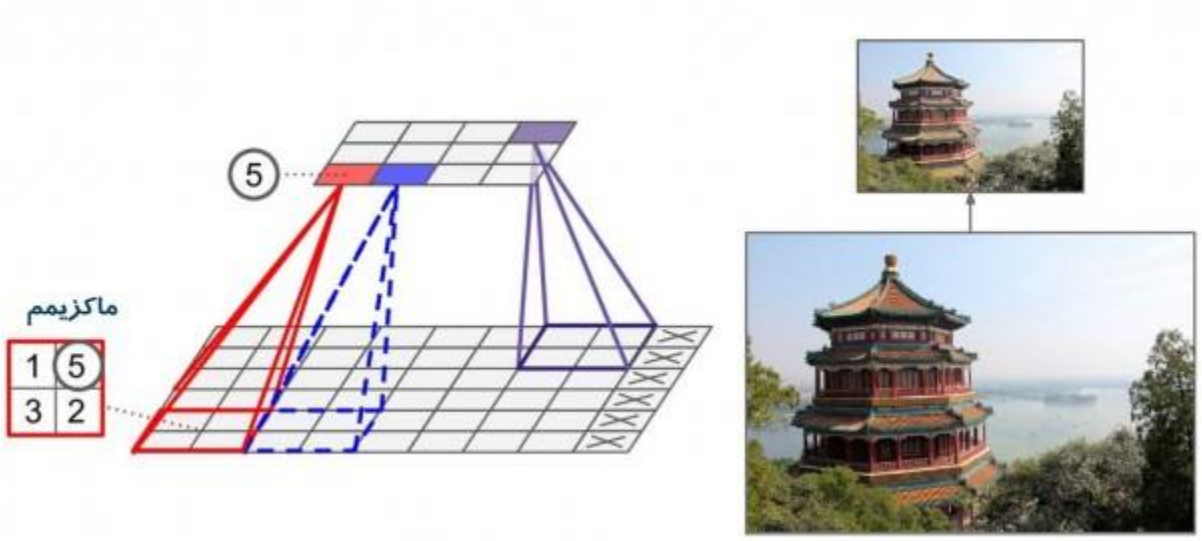
[4] batch
[5] epoch

dropout 2.5.4

در هنگام یادگیری، برخی از نورون‌ها به صورت شانسی رها می‌شوند.

Pooling 2.5.5

(کوچک کردن) تصویر ورودی به منظور کاهش بارمحاسباتی ، حافظه و تعداد پارامترها. برای لایه پولینگ شما چند تا مورد را باید مشخص کنید از جمله اندازه ، stride و نوع padding ی که مد نظر دارید.



تصویر ۸ pooling

Back propagation ۲.۵.۶

هنگامی که خروجی یک کار را می‌گیریم، می‌توانیم به کمک آن، مقدار خطای شبکه را محاسبه کنیم، سپس این مقدار را به همراه نمودار تابع هزینه به شبکه برگردانیم تا وزن‌های شبکه بروزرسانی شوند .

۳ فصل سوم

نصب موارد مورد نیاز برای اجرای پروژه های یادگیری عمیق

۳.۱ مقدمه

اجرای پروژه سه بعدی نیاز به نصب لینوکس،آناکوندا ،کودا تولکیت و نرم افزارهای دیگری دارد که در ادامه نحوه ی نصب آن ها بررسی می شود.هم چنین باید در محیطی که در آناکوندا ایجاد می شود،ورژن های به خصوصی از کتابخانه های پایتورچ نصب شود تا در هنگام اجرای پروژه با خطا مواجه نشویم.اجرای پروژه دو بعدی در ویندوز امکان پذیر است.

۳.۲ نصب لینوکس به صورت dual boot

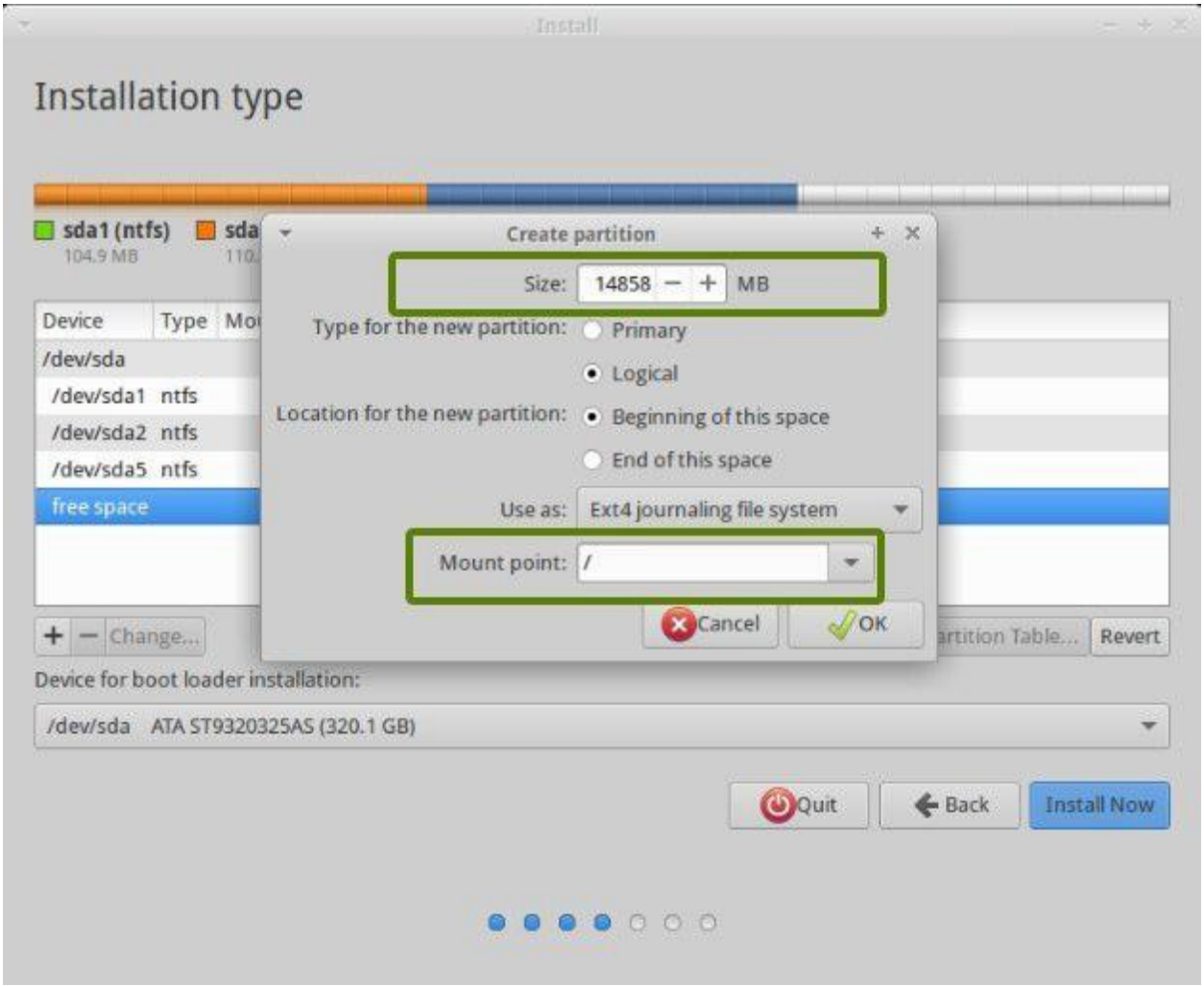
۳.۲.۱ ویندوز را برای Dual Boot شدن آماده کنید

از search ویندوز run را پیدا می کنیم.عبارت diskmgmt.msc را در آن وارد می کنیم.شما نمی توانید بیش از 4 پارتیشن به صورت primary داشته باشید.پس از یک پارتیشن پشتیبان تهیه می شود.سپس این درایور حذف و دوباره با همان نام ایجاد می شود.به این صورت پارتیشن به صورت logical ایجاد می شود.

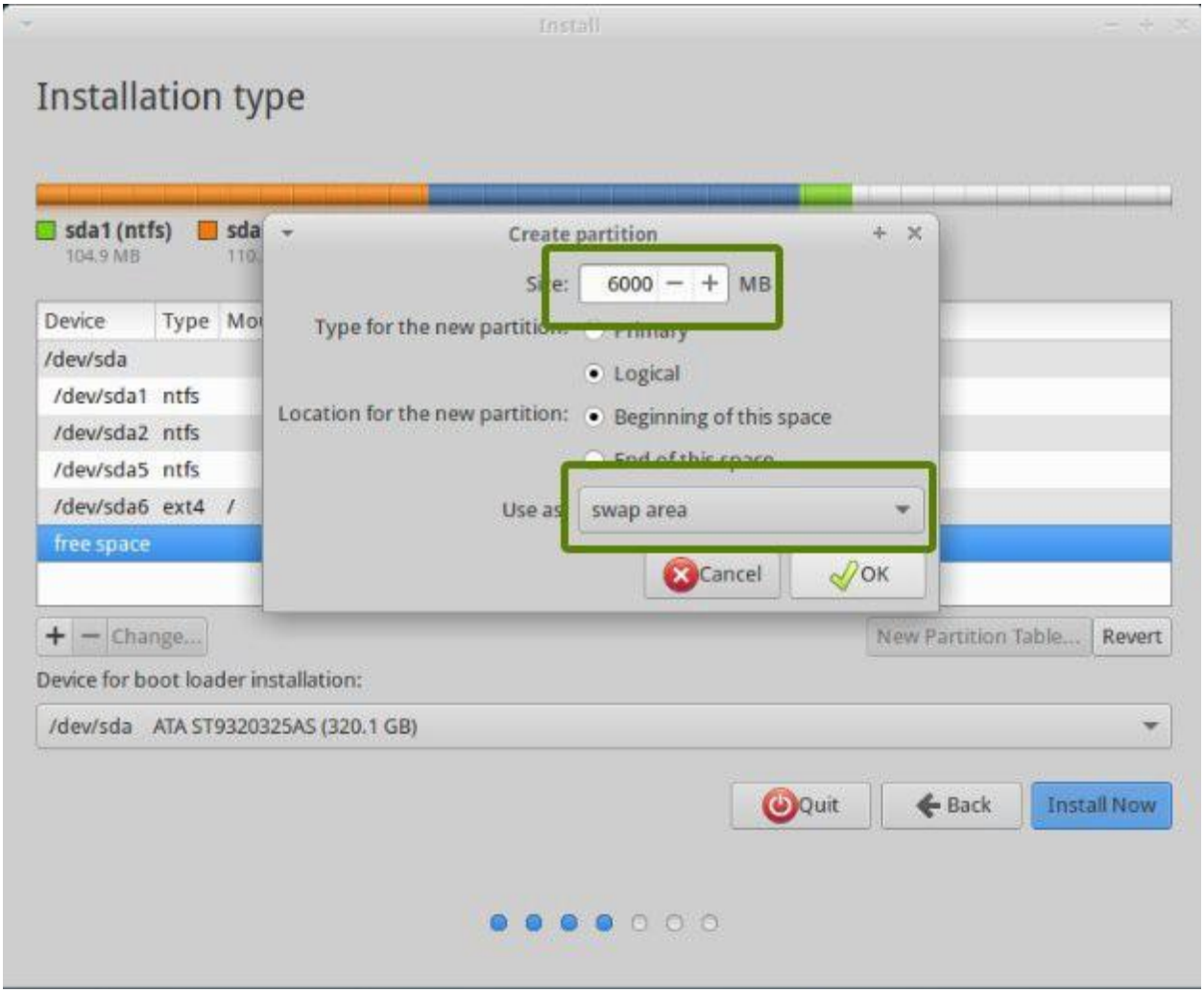
روی پارتیشن کلیک راست کرده و shrink volume را انتخاب می کنیم .سپس فضای لازم برای اوبونتو را انتخاب می کنیم.بهتر است این فضا 100-50 گیگابایت باشد. به محض اینکه تغییر سایز در فضای مورد نظر اعمال شد شما یک Unallocated Space روی سخت افزار مشاهده خواهید کرد.در صورتی که با داشتن 4 پارتیشن primary اقدام به shrink کردن کنید،فضای unallocated space غیر قابل استفاده خواهد بود.

۳.۲.۲ نصب اوبونتو

ایزوی اوبونتو دسکتاپ 18 را دانلود کنید.نرم افزار را Rufus برای bootable کردن آن دانلود کنید. سپس ویندوز را ریبوت کرده و گزینه ی something else را انتخاب می کنیم. پارتیشن بندی را به این صورت انجام می دهیم.بیشتر فضا را به / اختصاص می دهیم. به اندازه ی نصف سایز RAM را به swap اختصاص داده می شود.



تصویر ۹ فضای /



تصویر ۱۰ swap

۳.۳ نصب آناکوندا

فایل آناکوندا را از سایت آن دانلود کنید و با دستور زیر اقدام به نصب آن کنید:

```
sudo sh <AnacondaFileName>.sh
```

۳.۴ نصب درایور انویدیا

ابتدا بررسی کنید که سخت افزار GPU مناسب برای کار با cudatoolkit روی سیستم شما موجود باشد.

```
lspci | grep -i nvidia
```

کودا 10.0 نیاز به درایور انویدیا >=396 دارد.باید قبل از نصب مطابقت نسخه درایور قابل نصب روی سخت افزار با ورژن کودا بررسی شود.فایل کودا تولکیت شامل درایور انویدیا است ولی بهتر است با یکی از دستورهای زیر اقدام به نصب درایور کنید.

```
sudo ubuntu-drivers autoinstall
```

```
sudo apt install nvidia-435
```

با دستورات زیر از نصب درایور انویدیا مطمئن شوید:

```
lspci -k | grep -A 2 -i "VGA"
nvidia-smi
nvidia-settings
```

۳.۵ نصب کودا تولکیت

فایل cuda.run را از سایت انویدیا دانلود کنید.

```
sudo sh cuda_<version>_linux.run
```

در طی مراحل نصب بهتر است symlink را نیز برای نصب انتخاب کنید.متغیرهای path را به صورت زیر تنظیم کنید:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/cuda-10.0/lib64
export PATH=${PATH}:/usr/local/cuda-10.0/bin
```

با این دستور هم می توان کودا را نصب کرد:

```
sudo apt-get install cuda
```

فایل cudnn مربوط به نسخه ای از کودا که دانلود کرده اید را نیز دانلود کنید و با دستور زیر آن را نصب کنید.

```
sudo dpkg -i cudnn.deb
```

۳.۶ نصب VScode

```
sudo apt install --classic code
```

۳.۷ نصب git

```
sudo apt update
sudo apt install git
git –version
```

۳.۸ نصب ffmpeg و imagemagick

با این دستورات می توانید ffmpeg و imagemagick را نصب کنید و ویدیوها را به یکدیگر تبدیل کنید.ffmpeg برای اجرای پروژه روی ویدیو و imagemagick برای اجرای پروژه روی تصاویر مورد نیاز است.

```
sudo apt install ffmpeg
ffmpeg –version
```

sudo apt install imagemagick

ffmpeg -i input.mp4 output.avi

۳.۹ ایجاد محیطی در آناکوندا برای اجرای پروژه

۳.۹.۱ پروژه ۳d

با این دستور محیط جدیدی در آناکوندا ایجاد می شود:

conda create --name my_env python=3.7

محیط ایجاد شده را با این دستور فعال کنید:

conda activate my_env

تمام محیط های موجود را با این دستور مشاهده کنید:

conda info –envs

این دستور برای حذف محیط در آناکوندا است:

conda remove --name my_env –all

لیست تمام کتابخانه های نصب شده در یک محیط را با این دستور می توانید ببینید:

conda list

با این دستور پایتورچ ورژن GPU را نصب کنید:

conda install pytorch torchvision cudatoolkit=10.0 -c pytorch

موارد مورد نیاز دیگر را با این دستورات می توانید نصب کنید:

conda install matplotlib=3.0.3 numpy=1.16.1

۳.۹.۲ پروژه detectron

برای ایجاد محیط اجرای detectron از این دستورات استفاده کنید:

conda create --name my_env python=2.7

نصب **caffe** به این صورت انجام می شود:

conda install pytorch-nightly -c pytorch

بررسی کنید که نصب به درستی انجام شده باشد.

python -c 'from caffe2.python import core' 2>/dev/null && echo "Success" || echo "Failure"

```
python -c 'from caffe2.python import workspace; print(workspace.NumCudaDevices())'
```

پروژه COCOAPI را دریافت کنید.

```
git clone https://github.com/cocodataset/cocoapi.git
```

در مسیر مربوط به پروژه در مسیر PYTHONAPI دستور نصب را اجرا کنید.

```
make install
```

پروژه detectron را دریافت کنید.

```
git clone https://github.com/facebookresearch/detectron
```

موارد پیش نیاز اجرای پروژه را نصب کنید.

```
pip install -r $DETECTRON/requirements.txt
```

دستور نصب پروژه را در مسیر مربوط به پروژه اجرا کنید.

```
make
```

اطمینان حاصل کنید که پروژه به درستی نصب شده باشد.

```
python $DETECTRON/detectron/tests/test_spatial_narrow_as_op.py
```

فایل `inference/infer_video.py` را از پروژه `videopose3d` در پوشه `tools` کپی کنید.

۴ فصل چهارم

معرفی پروژه سه بعدی VideoPose3D

۴.۱ مقدمه

پروژه facebook research videoPose3D توسط فیس بوک برای تحقیقات انجام شده است و در CVPR 2019[۶] ارائه شده است.

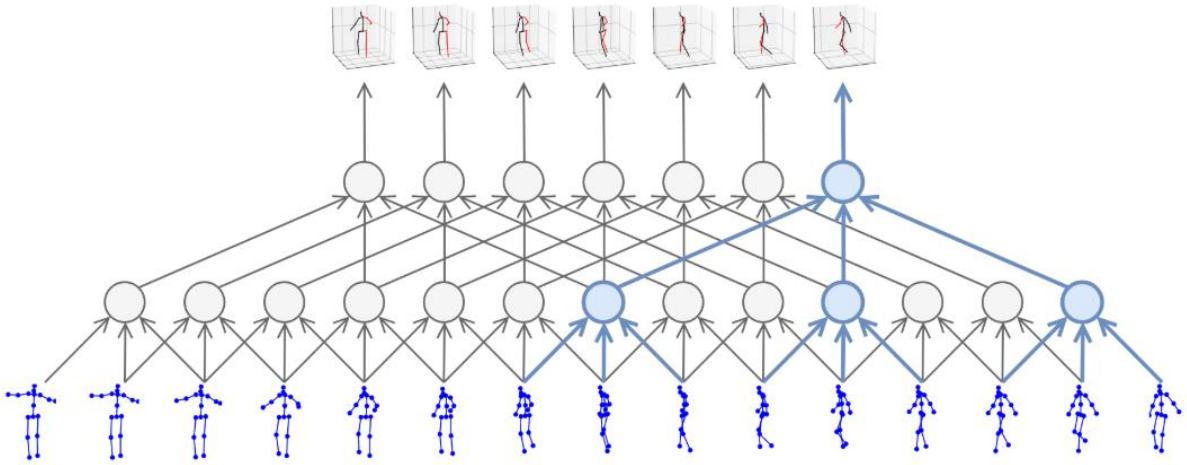
پروژه سه بعدی با استفاده از temporal convolution و روش نیمه نظارت شده برای training مفاصل را از ویدیو یا عکس استخراج می کند.

سپس نقاط سه بعدی به ورودی دوبعدی برگردانده می شود. (BACK PROJECTION)

مفاصل سه بعدی در ویدیو می تواند با یک مدل کانوولوشن بر اساس temporal dilated convolution روی نقاط دو بعدی به طور موثری به دست آید.هم چنین back projection به عنوان یک متد یادگیری ساده و موثر نیمه نظارت شده برای ویدیو های بدون برچسب معرفی می شود.این روش زمانی که داده های برچسب دار کمیاب هستند،مناسب است.

مدل کانوولوشن پردازش موازی چندین فریم را امکان پذیر می کند که با شبکه های recurrent امکان پذیر نیست.در مقایسه با رویکردهای مبتنی بر RNN[7] دقت بالاتر،سادگی و کارایی چه از نظر محاسبات وچه از نظر تعداد پارامترها به دست می آید.

[6] Conference on Computer Vision and Pattern Recognition
[7] Recurrent neural network



تصویر ۱۱ convolution

مدل temporal convolution توالی مفاصل دو بعدی را به عنوان ورودی دریافت می کند و تخمین سه بعدی را تولید می کند.برای به دست آوردن اطلاعات در یک توالی طولانی از dilated convolution استفاده می شود.

۴.۲ دیتاست Human ۳.۶m

برای جمع آوری این دیتاست از 15 سنسوراستفاده شده است که از قرار زیر می باشند:

- 4 digital video cameras
- 1 time-of-flight sensor
- 10 motion cameras

داده های این دیتاست از 15 حرکت برای برای آموزش تشکیل شده است مانند راه رفتن و تعدادی حرکات نامتقارن(راه رفتن با یک دست در جیب و راه رفتن با یک کیف روی شانه)،نشستن و انواع مختلفی از حالات صبر کردن و انواع دیگر حرکات.

ناحیه ی مورد نظر برای گرفتن ویدیو در حدود 5 متر در 6 متر بوده است و در داخل این ناحیه 4 متر در 3 متر فضای موثر وجود داشته است،جایی که نمونه ها در همه ی دوربین ها کاملا قابل مشاهده بودند.

برای قابلیت سازگاری و راحتی از 32 نقطه برای همه ی پارامترسازی ها استفاده می شود.در زمان تست کردن تعداد مفاصل مرتبط را کاهش می دهیم مثلا فقط یک نقطه برای هر دست و هر پا در نظر می گیریم.در این دیتاست مختصات دو بعدی نقاط نیز فراهم شده است.

4.3 مدل temporal dilated convolution

۴.۳.۱ نتایج بهتر نسبت به RNN

این پروژه از نقشه حرارتی[8] استفاده نمی کند.این مدل یک معماری کاملا بر اساس کانوولوشن با اتصالات باقیمانده است که یک توالی از حالات ورودی دو بعدی را می گیرد و از طریق temporal convolution آن ها را تبدیل می کند. مدل های کانوولوشن موازی سازی در دسته بندی و بعد زمان را ممکن می سازد در حالیکه RNN در زمان به طور موازی اجرا نمی شود. در مدل کانوولوشن مسیر بین گرادیان ورودی و خروجی یک طول ثابت صرف نظر از طول توالی دارد.به این ترتیب مسئله ی محو شدگی گرادیان[9] و انفجار گرادیان[10] که RNN را تحت تاثیر قرار می دهد،برطرف می شود.هم چنین در مدل کانوولوشن فیلد پذیرنده موقتی می تواند به دقت مورد کنترل قرار گیرد که برای مدل کردن وابستگی های زمانی برای تخمین سه بعدی مفید است.افزون بر این dilated convolution برای مدل کردن وابستگی های بلند مدت به کار رفته و هم زمان کارایی نیز حفظ شده است.

مدل های کانوولوشن معمولا لایه گذاری صفر انجام می دهند و تعداد ورودی و خروجی آن ها برابر است.در این پروژه کانوولوشن بدون لایه گذاری[11] و لایه گذاری توالی ورودی با کپی مقادیر مرزی در چپ و راست نتایج بهتری را نشان داده است.

۴.۳.۲ محو شدگی گرادیان و انفجار گرادیان

هرچه فاصله وابستگی بلند مدت بیشتر شود شبکه های عصبی RNN با مشکل بیشتری در یادگیری این وابستگی ها مواجه میشوند چرا که یا با مشکل محو شدگی گرادیان یا انفجار گرادیان برخورد میکنند.

در حین آموزش یک شبکه عمیق گرادیان ها پس از انتشار از انتهای شبکه به ابتدای شبکه،ضرب های متعددی را پشت سر می گذارند و رفته رفته مقادیر آن ها بسیار کوچک می شود(کمتر از 1) و آموزش متوقف می شود چون مقادیر ناچیز گرادیان ها تغییری در وزن ها صورت نمی دهند.این مشکل را محو شدگی گرادیان می گویند.

به همین ترتیب ممکن است مقادیر گرادیان به مرور آن قدر بزرگ شوند تا مدل دچار خطا گردد و سرریز در محاسبات رخ دهد.به این مسئله انفجار گرادیان گویند.یکی از علائم آن دریافت ارور NAN در پایتورچ می باشد.

۴.۴ رویکرد نیمه نظارت شده

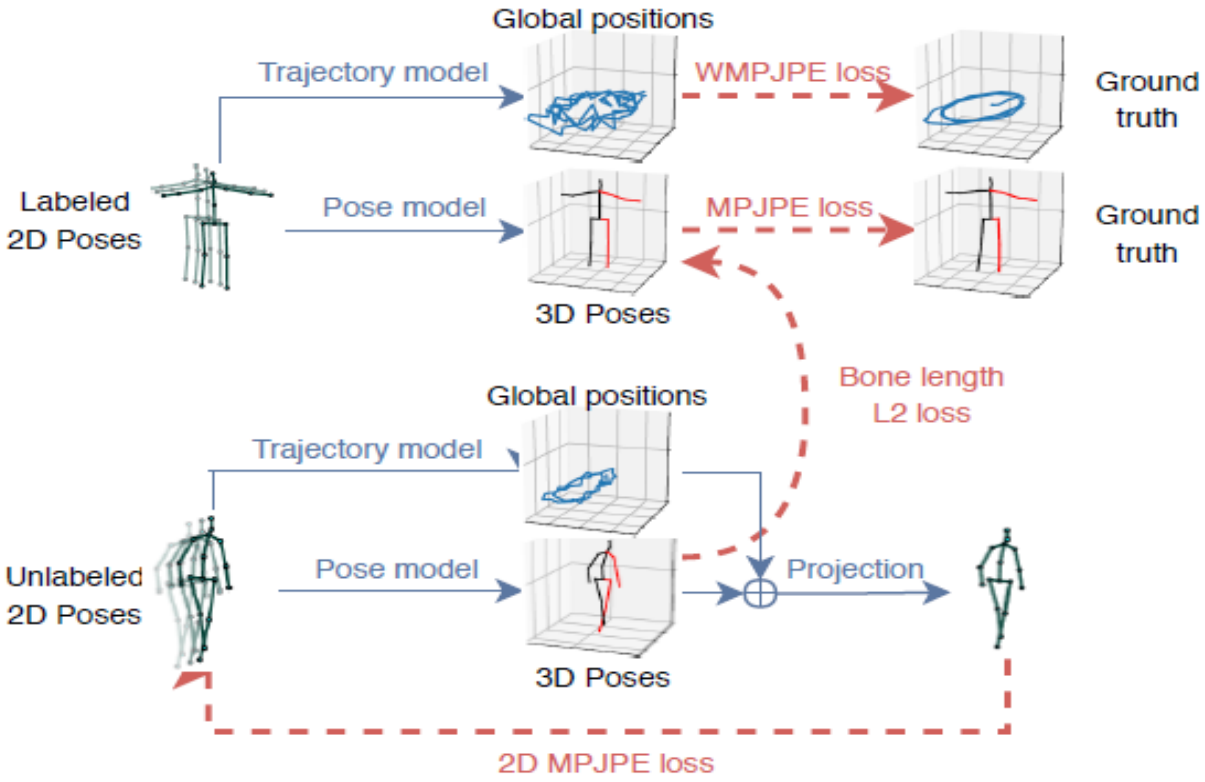
از روش نیمه نظارت شده برای افزایش دقت زمانی که داده های حالت های سه بعدی حقیقی[12] کمیاب باشد،استفاده می کنیم.تشخیص دهنده ی حالت[13] نقاط سه بعدی را از روی نقاط دو بعدی تخمین می زند و لایه ی projection[14] نقاط سه

[8] heatmap
[9] Vanishing gradient
[10] Exploding gradient
[11] unpadded
[12] Ground truth
[13] Encoder(pose estimator)
[14] decoder

بعدی را به نقاط دو بعدی بر می گرداند.در زمان آموزش اگر نقاط دو بعدی از ورودی اصلی فاصله ی زیادی داشته باشند،جریمه انجام می شود.

برای داده ی برچسب دار از داده های حالت های سه بعدی حقیقی برای آموزش یک تابع خطای نظارت شده[15] استفاده می شود.داده های بدون برچسب در پیاده سازی یک انکودر خطای خودکار در زمان برگشت حالات سه بعدی به حالات دو بعدی و بررسی سازگاری با ورودی کاربرد دارد.یک محدودیت نرم برای تطابق میانگین طول استخوان ها داده های برچسب دار و بدون برچسب اعمال می شود.

Back projection یک روش نیمه نظارت شده ی یادگیری برای ارتقای عملکرد در زمانی است که داده های برچسب دار کمیاب باشند.این روش فقط به پارامترهای داخلی دوربین نیاز دارد که آن را برای سناریوهایی که motion capture چالش برانگیز است،عملی می کند.

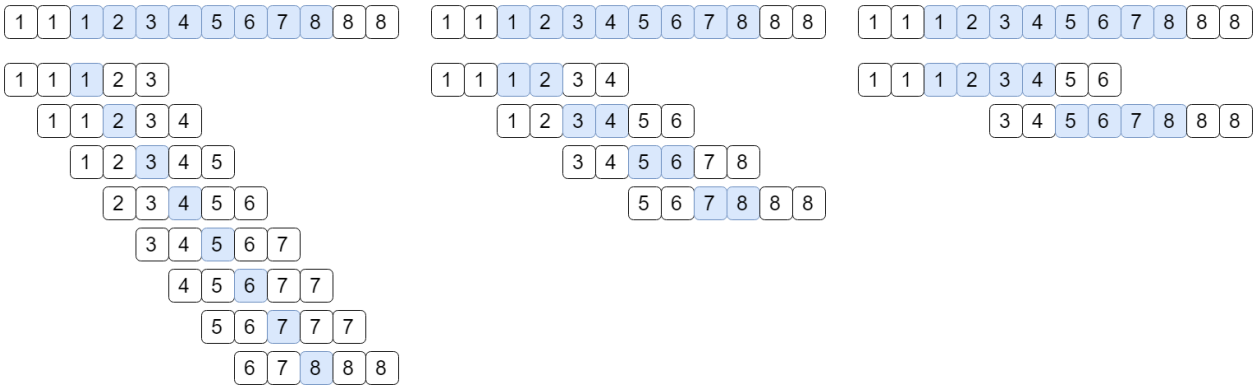


تصویر ۱۲ یادگیری نیمه نظارت شده با حالات دو بعدی بعنوان ورودی(WMPJPE: weigthed MPJPE)

۴.۵ پردازش دسته ها در هنگام آموزش شبکه

تولید دسته ها در هنگام training به صورت تصویر بهتر بیان می شود:

[15] supervised



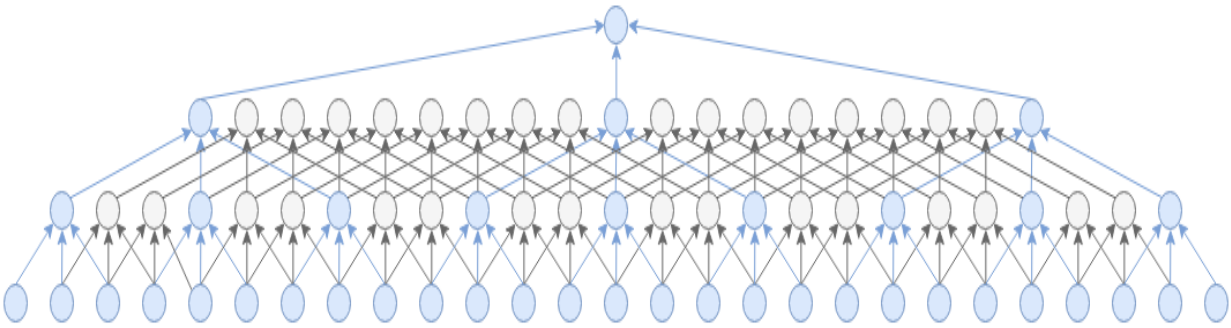
تصویر ۱۳ batch generation during training

نحوه ی دسته بندی بستگی به مقدار stride-- دارد.مقادیر stride[16] از چپ به راست به ترتیب 1و2و4 می باشد.این مثال یک توالی از حالات[17] دو بعدی را نشان می دهد که تعداد فریم ها در آن N=8 است.حالات سه بعدی(مربع های آبی) با یک مدل با فیلد پذیرنده ی[18] F=5 استنباط شده اند.به خاطر لایه گذاری معتبر طول توالی خروجی N-F+1

خواهد بود.

وقتی stride=1 باشد،برای هر فریم یک مثال در training ایجاد می شود.با این کار مطمئن می شویم که دسته ها حد اکثر ناهمبستگی را دارند.اگر stride زیاد شود،آموزش شبکه سریع تر می شود زیرا مدل می تواند محاسبات میانی را دوباره به کار برد.این امر به قیمت بایاس شدن دسته ها است.در هنگام آموزش شبکهاگر سایز دسته بندی 1 باشد (stride=1) یک پیاده سازی بهینه تعبیه شده که dilated convolution را با strided convolution جایگزین می کند.

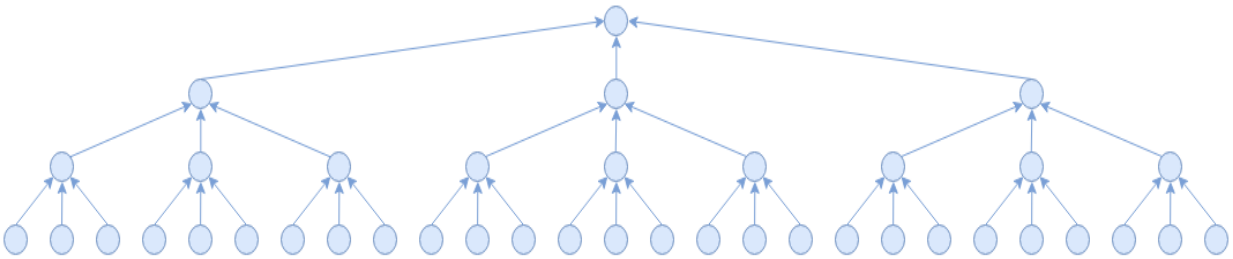
اگر سایز دسته بندی بیشتر باشد سرعت بیشتر می شود ولی برخی ویژگی ها در آن دسته بندی نادیده گرفته می شود.



تصویر۱۴ *dilated convolution*

[16] گام
[17] poses
[18] Receptive field

تصویر بالا یک مدل با فیلد پذیرنده ی 27 فریم را نشان می دهد.از این 27 فریم ورودی یک فریم خروجی ایجاد می شود.این پیاده سازی باعث می شود برخی نتایج میانی در زمانی که تعداد کمی فریم پیش بینی می شود، نادیده گرفته شود.اگرچه برای یک توالی طولانی از فریم ها این رویکرد کارایی بالایی دارد زیرا نتایج میانی بین فریم های متوالی به اشتراک گذاشته می شود.

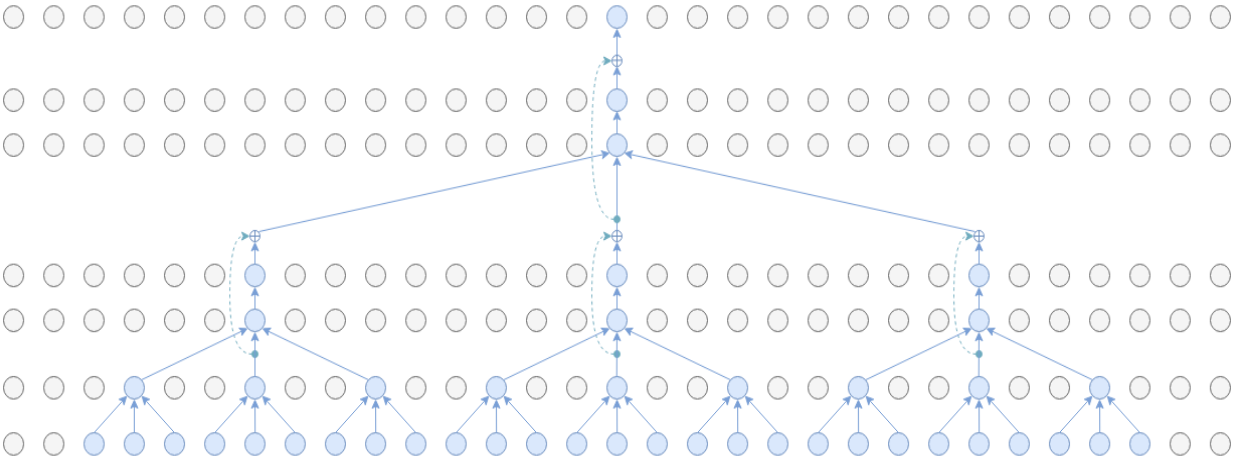


تصویر۱۵ strided convolution

بنابر این،**فقط** برای آموزش شبکه از پیاده سازی بالا استفاده می شود که از استراد کانوولوشن به جای دایلیت کانوولوشن استفاده می کند.این پیاده سازی نتایج یکسانی را تولید می کند اما از محاسبه ی غیر ضروری نتایج میانی اجتناب می کند.

۴.۶ کانوولوشن متقارن[19] در برابر کانوولوشن علیت[20]

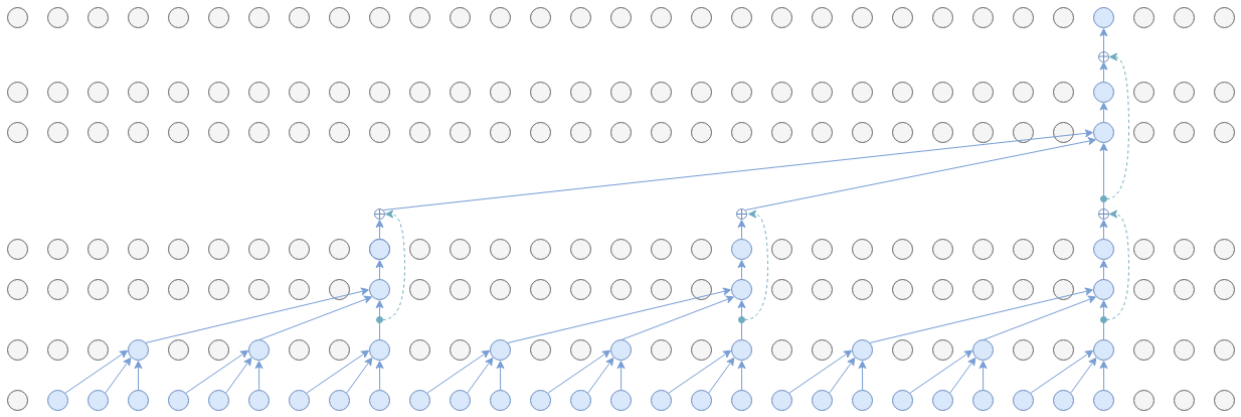
تصویر زیر جریان اطلاعات را از ورودی در پایین به خروجی در بالا نمایش می دهد.یک مدل با فیلد پذیرنده ی 27 استفاده شده است.



تصویر۱۶ کانوولوشن متقارن

[19] Symmetric convolution
[20] Causal convolution

تصویر بالا یک کانوولوشن به صورت متقارن را نمایش می دهد.ازفریم های قبلی و بعدی استفاده می شود تا نتیجه ی بهتری در بازسازی[21] تصویر به دست آید.



تصویر۱۷ causal convolution

شکل بالا کانوولوشن با استفاده از فریم های قبلی را نشان می دهد.این رویکرد برای نرم افزارهای بلادرنگ مناسب است زیرا اطلاعات آینده قابل بهره برداری نیست. هزینه این رویکرد خطای کمی بالاتراست.

۴.۷ اجرای کد ارزیابی در پروژه

پس از دانلود کردن مدل حاصل از آموزش روی دیتاست human 3.6m از این کد برای ارزیابی با استفاده از توابع خطا استفاده کنید:

```
python run.py -k cpn_ft_h36m_dbb -arc 3,3,3,3,3 -c checkpoint --evaluate
pretrained_h36m_cpn.bin
```

۴.۸ مقایسه مفاصل پروژه سه بعدی و کینکت ورژن ۱

جدول۲ مفاصل کینکت متناظر با پروژه سه بعدی

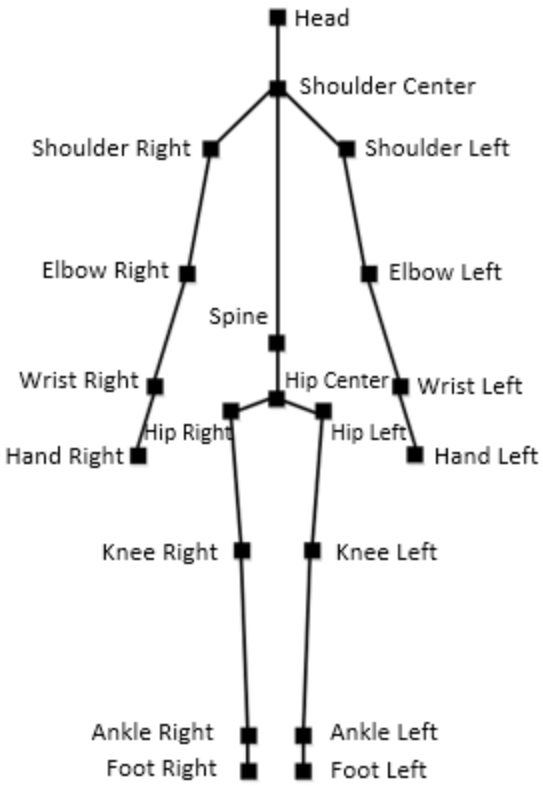
شماره در کینکت	نام در کینکت	شماره در پروژه دیپ
1	Head	**10**
2	ShoulderCenter	**8**
3	ShoulderLeft	**11**
4	ShoulderRight	**14**
5	Spine	**7**

[21] reconstruction

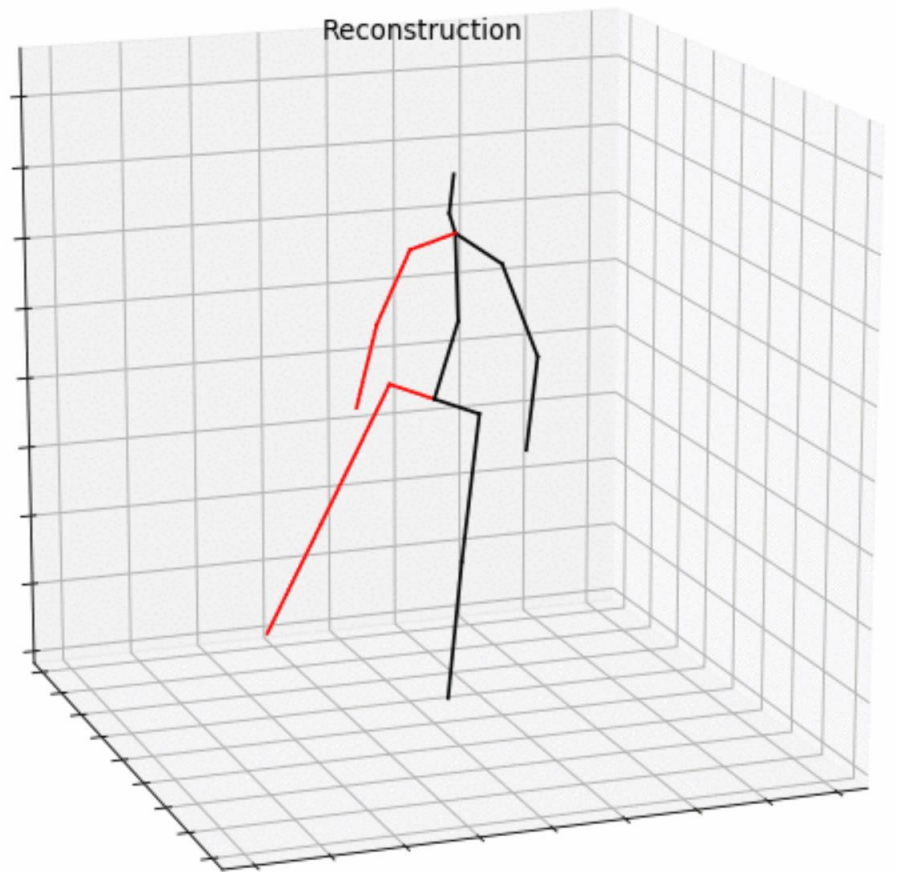
0	HipCenter(تخمین)	6
4	HipLeft	7
1	HipRight	8
12	ElbowLeft	9
13	WristLeft	10
15	ElbowRight	12
16	WristRight	13
5	KneeLeft	15
6	AnkleLeft	16
2	KneeRight	18
3	AnkleRight	19
9	تخمین(jaw)	_

همان طور که در جدول بالا مشاهده می شود هر یک از مفاصل نشان داده شده در پروژه ی سه بعدی با مفصل متناظر آن ها در دوربین کینکت ورژن 1 با مقایسه ی تصاویر و استنباط از آن ها به دست آمده است.

مفصل HipCenter در پروژه ی سه بعدی و دوربین کینکت به طور متفاوتی تخمین زده شده است.



تصویر ۱۸ نمایش اسکلت بدن در کینکت



تصویر ۱۹ نمایش اسکلت بدن در پروژه سه بعدی

هم چنین مفصل فک در پروژه ی videopose3d تخمین زده می شود ولی با کینکت تخمین زده نمی شود.

۴.۹ بررسی کد پروژه

۴.۹.۱ argument.py

در این فایل آرگومان های ورودی مانند d– و k– بررسی می شوند.این آرگومان ها در 5 دسته ی آرگومان های عمومی،مدل،visualization و Experimental visualization دسته بندی می شوند.تنظیمات نامعتبر نیز چک می شود.

۴.۹.۲ camera.py

تابع normalize_screen_coordinates مختصات را به نحوی نگاشت می کند که [0,w] به [-1,1] نگاشت می شود.

تابع image_coordinates بر عکس normalization کار می کند.

world_to_camera تابعی است که چرخش را معکوس می کند.

project_to_2d نقاط سه بعدی را به دو بعدی تبدیل می کند.یک پیاده سازی مجدد همراه با دسته بندی از متلب است.

project_to_2d_linear نقاط سه بعدی را پارامتر های خطی(focal length, principal point) به دو بعدی تبدیل می کند.

۴.۹.۳ custom_dataset.py

دیتاست را بارگذاری می کند و آن را از 32 مفصل به 17 مفصل کاهش می دهد.هم چنین شانه ها را به والد مناسب دوباره متصل می کند.

۴.۹.۴ generators.py

ChunkedGenerator با داده های دسته بندی شده کار می کند و برای آموزش استفاده می شود.توالی به قسمت هایی با طول مساوی تقسیم می شود و در صورت ضرورت لایه گذاری انجام می شود.در حین آموزش نقاط دو بعدی و سه بعدی معکوس می شوند.آرگومان های ورودی آن عبارتند از سایز دسته بندی،لیست دوربین ها برای آموزش نیمه نظارت شده،لیست نقاط واقعی حالات سه بعدی،لیست نقاط دو بعدی،تعداد فریم های خروجی برای پیش بینی(معمولا 1)،لایه گذاری دوبعدی، آفست نامتقارن برای کانوولوشن علیت(0 یا pad)،augment که حالات را به طور افقی معکوس می کند.kps-left و kps-right نقاط دو بعدی در هر طرف از بدن را نشان می دهد.joints_left و joints_right نقاط سه بعدی در دو طرف بدن را نمایش می دهد.خط 167-99 مربوط به آموزش شبکه است.

UnchunkedGenerator داده ها به صورت دسته بندی نیستند و برای تست کردن استفاده می شود.توالی یکی یکی برگردانده می شود(مانند اینکه سایز دسته بندی 1 باشد).اگر data augmentation فعال باشد،دسته بندی دو توالی را در بر دارد(سایز دسته بندی 2)،دومی بر عکس اولی است.

۴.۹.۵ h36m_dataset.py

خط 14-18 اطلاعات مفاصل چپ و راست در دیتاست human 3.6m نشان می دهد.خط 19-61 اطلاعات پارامترهای داخلی دوربین را نشان می دهد.خط 62-208 اطلاعات پارامترهای خارجی دوربین را برای هر موضوع[۲۲] نشان می دهد.خط 221-225 فریم های دوربین را نرمال می کند.خط 227-231 پارامترهای داخلی دوربین را اضافه می کند.خط 233-243 به بارگذاری نقاط دیتاست می پردازد.خط 245-248 تعدادی از نقاط را حذف می کند تا به 17 نقطه ای برسیم که در پروژه استفاده می شود.خط 249-251 نقاط شانه را به والد مناسب مجدد متصل می کند.

[22] Subjects in human 3.6m

loss.py ۴.۹.۶

خط 11-18 میانگین خطای موقعیت مفاصل[۲۳](میانگین فاصله اقلیدسی) را محاسبه می کند.خط 19-26 MPJPE وزن دار[۲۴] را محاسبه می کند.خط 27-67 خطای حالات[۲۵] MPJPE را بعد از مقیاس، چرخش و انتقال محاسبه می کند.MPJPE نرمال شده[۲۶] را فقط با تغییر مقیاس محاسبه می کند.خط 80-89 میانگین خطای شتاب[۲۷] در مفاصل را محاسبه می کند.

model.py ۴.۹.۷

خط 10-78 TemporalModelBase را نشان می دهد که ورودی ها را اعتبارسنجی می کند(فقط فیلترهای عرضی فرد پشتیبانی می شود).خط 41-48 حوزه ی پذیرای مدل بنابر تعداد فریم را بر می گرداند.خط 50-62 آفست نامتقارن برای لایه گذاری توالی را برمی گرداند.اگر کانوولوشن علیت غیرفعال باشد این مقدار 0 است.در غیر اینصورت نیمی از اندازه ی فیلد پذیرنده است.خط 79-139 مدل سه بعدی با کانوولوشن را نمایش می دهد.این پیاده سازی می تواند برای همه ی موارد کاربرد استفاده شود.خط 85-120 مدل را ایجاد می کند.آرگومان های آن عبارتند از تعداد نقاط ورودی(17 برای human 3.6m)،تعداد ویژگی های هر نقطه ی ورودی(2 برای نقاط دو بعدی)،احتمال dropout ،تعداد نقاط خروجی،filter-width که فیلد پذیرنده را مشخص می کند و اینکه کانوولوشن علیت به جای کانوولوشن متقارن برای کاربرد های واقعی استفاده شود. خط 140-197 یک مدل بهینه سازی برای سایز دسته بندی یک فریم در نظر گرفته شده و هر دسته به اندازه ی فیلد پذیرنده است و اندازه ی خروجی 1 است.این مورد فقط در آموزش وقتی stride=1 باشد رخ می دهد.این مدل دایلیت کانوولوشن را با استراید کانوولوشن جایگزین می کند تا از تولید نتایج میانی بدون استفاده پرهیز کند.

quaternion.py ۴.۹.۸

خط 10-25 چرخش را اعمال می کند.خط 27-35 چرخش را معکوس می کند.

skeleton.py ۴.۹.۹

خط 31-70 نقاط مشخص شده را حذف می کند.

utils.py ۴.۹.۱۰

خط 12-43 یک تابع تورچ را بسته بندی می کند تا بتواند با آرایه های numpy کار کند.نوع ورودی و خروجی به طور یکپارچه تبدیل می شوند.خط 44-47 برای انکود است.

visualization.py ۴.۹.۱۱

خط 17-23 شفافیت را بدست می دهد.خط 25-32 فریم بر ثانیه را به دست می دهد.خط 33-42 فیلم را به عنوان ورودی می خواند.خط 62-206 انیمیشن خروجی را ارائه می دهد.می تواند mp4 باشد که نیاز به ffmpeg دارد و می تواند gif

[23] Mean per-joint position error(MPJPE)
[24] WMPJPE
[25] PMPJPE
[26] NMPJPE
[27] mean per joint velocity error(MPJVE)

باشد که نیاز به imagemagick دارد.فیلم را دیکود می کند و فیلم را از ffmpeg بارگذاری می کند و اسکلت بدن را در صورتی که والد را داشته باشیم(نقاط همخوانی داشته باشند)،ترسیم می نماید.

Run.py ۴.۹.۱۲

خط 30-36 اگر پوشه checkpoint وجود نداشت آن را ایجاد می کند.خط 37-49 دیتاست را بارگذاری می کند.خط 51-62 داده ها را آماده می کند.خط 63-70 نقاط تشخیص داده شده ی دو بعدی را بارگذاری می کند.نقاط دو بعدی را در kps-left و kps-right و نقاط سه بعدی را در joints-left و joints-right می گذارد.خط 91-98 فریم های دوربین را نرمال می کند.خط 99-109 موضوعات برای آموزش و یادگیری نیمه نظارت شده را مشخص می کند.خط 110-163 موضوع و فعالیت مربوط را از دیتاست واکشی می کند و در صورت نیاز downsample انجام می دهد.خط 165-168 فعالیت مربوط را مشخص می کند.خط 171-181 مشخص می کند که آیا از مدل بهینه شده استفاده شود یا خیر.هم چنین در صورت شناسایی تنظیمات غیرسازگار(سایز دسته بندی بیشتر از یک باشد یا بهینه سازی غیر فعال باشد) به مدل معمول بر می گردد.خط 182-185 مدل معمول را فراخوانی می کند.خط 186-193 فیلد پذیرنده،لایه گذاری و کانوولوشن علیت را مشخص می کند.خط 200-203 در صورت داشتن کودا روی سیستم از آن استفاده می کند.خط 212-214 UnchunkedGenerator را برای تست فراخوانی می کند.خط 217-641 برای آموزش است.نرخ یادگیری و اینکه کدام مدل(مدل معمول یا مدل بهینه برای سایز دسته بندی 1) استفاده شود مشخص می شود.هم چنین بهینه ساز Adam استفاده می شود که محبوب‌ترین روش برای بهینه سازی است.از خط 299 دوره های آموزش شروع می شود.خط 641-707 مربوط به ارزیابی است.خط 656 مدل موقعیت ها را نمایش می دهد.خط 659-664 با داده های معکوس نشده میانگین را حساب می کند.خط 701 خطای شتاب را محاسبه می کند.خط 693-702 خطا ها را محاسبه می کند.خط 707-762 برای نمایش انیمیشن به همراه اسکلت بدن است.خط 735-749 انتقال دوربین را معکوس می کند و در صورت نبود نقاط واقعی[۲۸] پارامترهای خارجی دوربین را از یک موضوع تصادفی برای نمایش انیمیشن استخراج می کند.خط 764-842 به ارزیابی می پردازد.خط 781-803 تابع واکشی است و در صورت بزرگتر بودن سایز دسته بندی از 1، downsample انجام می شود.خط 805-835 ارزیابی را اجرا می کند و MPJPE و WMPJPE و NMPJPE و MPJVE را چاپ می کند.

[28] Ground truth

۵ فصل پنجم

معرفی پروژه دو بعدی OpenPose

۵.۱ مقدمه

پروژه openpose یک پروژه یادگیری عمیق برای استنباط مختصات دو بعدی یک ویدیو است و در CVPR 2017 ارائه شده است.از دیتاست body_25 استفاده می شود.

این پروژه هم چنین برای استنباط مختصات سه بعدی نیز استفاده می شود.ما به دلیل دقیق تر بودن پروژه ی videopose3d از آن پروژه استفاده کردیم.هم چنین می توان برای شناسایی و تخمین نقاط صورت و دست ها و کالیبریشن(تخمین ساده ی پارامتر های داخلی و خارجی[۲۹] دوربین و اعوجاج[۳۰]) از آن استفاده کرد.

در این پروژه 25 نقطه تشخیص داده می شود که در این میان نقاط بیشتری را نسبت به پروژه videopose3d در صورت و دستان فرد تشخیص می دهد.

این پروژه به طور ضمنی وابستگی های محدوده طولانی بین متغیرهای در وظایف پیش بینی ساختار یافته مانند برآورد حالات انسانی را مدل می کند و با طراحی یک معماری متوالی متشکل از شبکه های convolutional که به طور مستقیم بر روی نقشه های باور از مراحل قبلی عمل می کنند، بدون نیاز به استنتاج سبک گرافیکی صریح، به این مساله دست می یابد. رویکرد ما به دشواری مشخصه کاهش شیب در طول آموزش از طریق فراهم نمودن یک تابع هدف یادگیری طبیعی که نظارت میانجی را ایجاد می کند، می پردازد، در نتیجه گرادیان منتشر شده را مجدد تولید می کند و رویه یادگیری را هموار می کند.

[29] Camera intrinsic and extrinsic parametres
[30] distortion

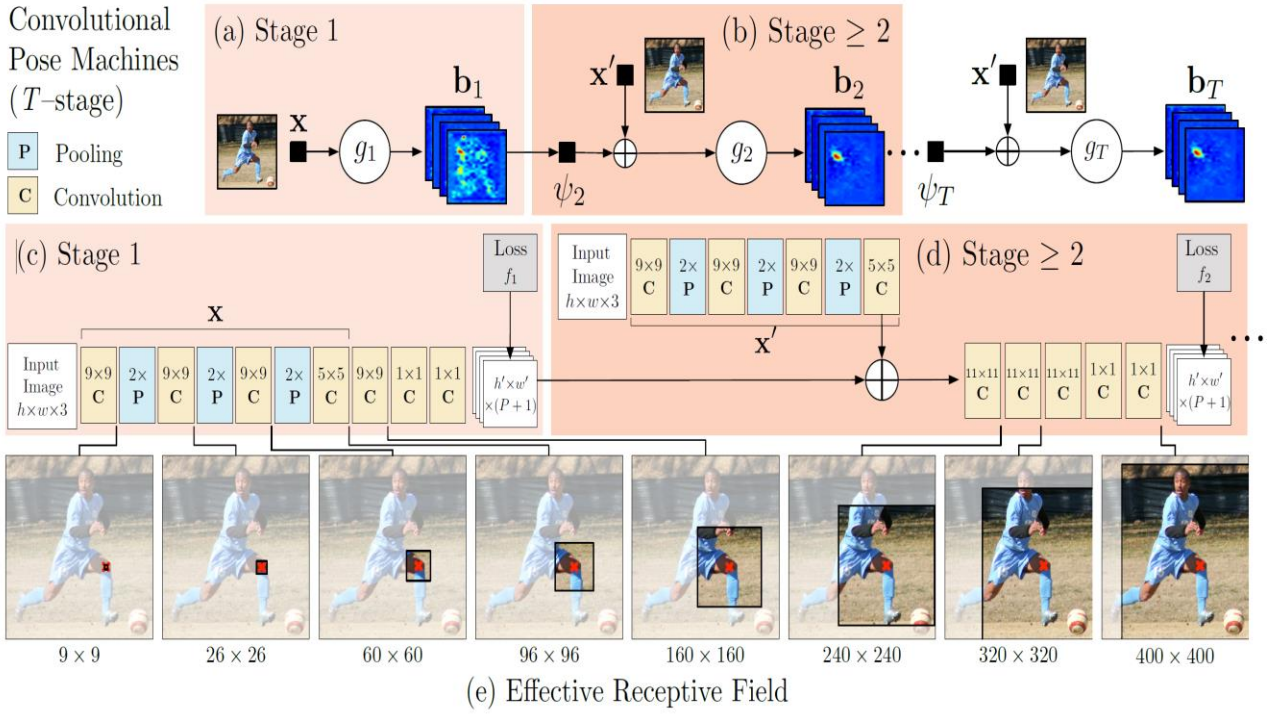
مزیت های معماری کانوولوشن عبارتند از:

- توانایی یادگیری نمایش های ویژگی برای تصویر و بافت فضایی به طور مستقیم از داده ها.
- معماری که امکان آموزش مفاصل با پس انتشار[۳۱] را فراهم می آورد.
- توانایی انجام کارآمد مجموعه داده های آموزشی بزرگ.

دنباله شبکه های کانوولوشن که به طور مکرر نقشه های باور 2D برای مکان هر قسمت تولید می کند. ما شبکه های کانوولوشن را یاد می گیریم که به طور مستقیم با نقشه های باور میانی کار می کنند و مدل های مکانی مستقل از تصویر ضمنی از روابط بین قسمت ها یاد می گیرند.

در هر مرحله نقشه های باور با تخمین به طور فزاینده تصفیه شده برای مکان ها برای هر قسمت تولید می شود. به منظور دستیابی به برهم کنش متقابل محدوده طولانی بین اجزا، طراحی شبکه در هر مرحله از چارچوب پیش بینی متوالی ما با هدف دستیابی به یک حوزه پذیرای بزرگ در هر دو تصویر ذهنی و هم در نقشه های باور، برانگیخته شده است.

۵.۱.۱ ماشین حالت کانوولوشن[۳۲]



تصویر ۲۰ معماری کانوولوشن و حوزه پذیرنده

ماشین حالت در a و b نشان داده شده است.شبکه ی کانوولوشن در c و d نشان داده شده است.a و c معماری را نشان می دهد که تنها براساس شواهد تصویری در مرحله اول عمل می کند. معماری مراحل بعد هم بر روی شواهد تصویری و هم نقشه

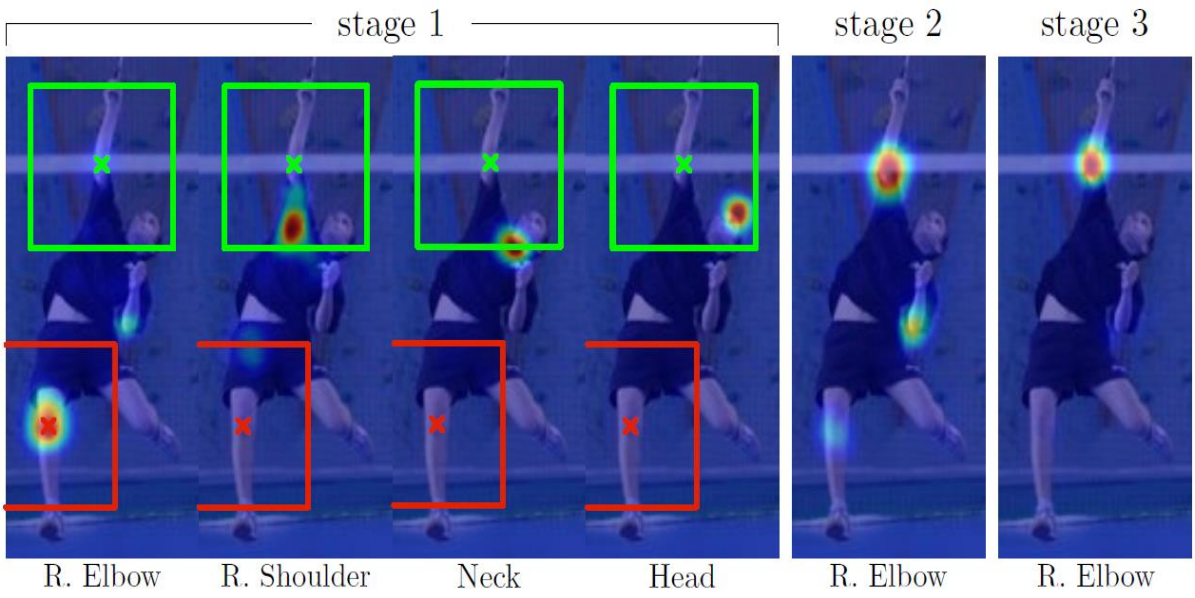
[31] Back propagation
[32] Convolutional Pose machines

های باور از مراحل قبلی فعالیت می کند. شبکه به طور محلی بعد از هر مرحله با استفاده از یک لایه واسطه میانی تحت نظارت قرار می گیرد که مانع از محو شدن شیب در حین آموزش می شود. یک حوزه ی پذیرنده ی بزرگ، این مدل را قادر می سازد تا طیف وسیعی از وابستگی های مکانی محدوده طولانی مانند آن هایی که بین سر و دو زانو هستند را ثبت نماید.

طراحی ما برای یک CPM که مزایای معماری کانوولوشن عمیق را با مدل سازی فضایی ضمنی که توسط چارچوب ماشین حالت به دست می آید،با هم ترکیب می کند.

۵.۱.۲ یافتن محل نقاط با استفاده از شواهد تصویر محلی

اولین مرحله یک ماشین حالت کانوولوشن، نقشه ی باور را تنها از شواهد تصویری محلی پیش بینی می کند.شواهد محلی است به این دلیل که فضای پذیرای اولین مرحله از شبکه به قطعه کوچکی در اطراف پیکسل خروجی محدود می شود. از یک ساختار شبکه متشکل از 5 لایه کانوولوشن و 2 لایه ی 1*1 کانوولوشن تشکیل شده است که یک معماری کاملا کانوولوشن را به دست می دهد.



تصویر ۲۱ محتوای مکان از نقشه های باور

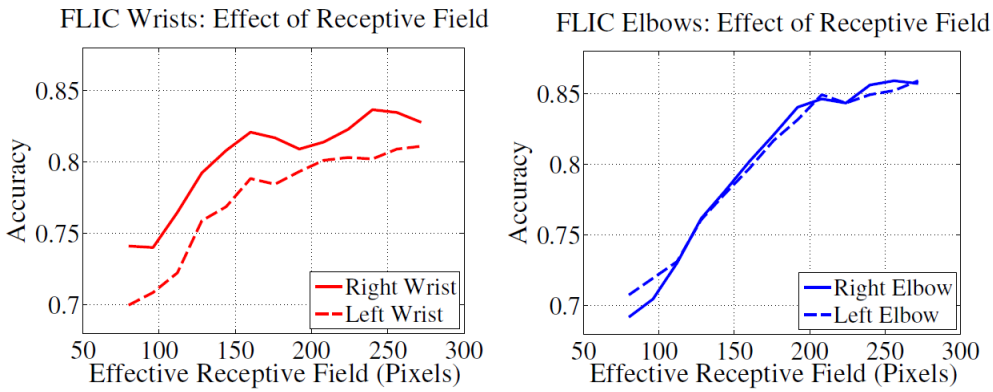
محتوای مکان از نقشه های باور قسمت هایی که تشخیص آن ها راحت تر است، اطلاعات دقیق را برای مکان یابی قسمت هایی که تشخیص آن ها سخت تر است فراهم می کند.محتوای مکان از شانه، گردن و سر می توانند به حذف اشتباه کمک کنند (قرمز) و برآوردی صحیح (سبز)بر روی نقشه ی باور آرنج راست را تقویت می کنند.

هنگام تشخیص قطعات چالش برانگیز مانند آرنج راست،نقشه باور برای شانه راست با قله تیز به عنوان یک نشانه قوی قابل استفاده است.

۵.۱.۳ پیش بینی متوالی با یادگیری مکانی ویژگی های متن

دقت معمولاً برای نشانگرهای پایین زنجیره حرکتی اسکلت انسان به دلیل وجود تفاوت زیاد در پیکربندی و ظاهر بسیار کمتر است.

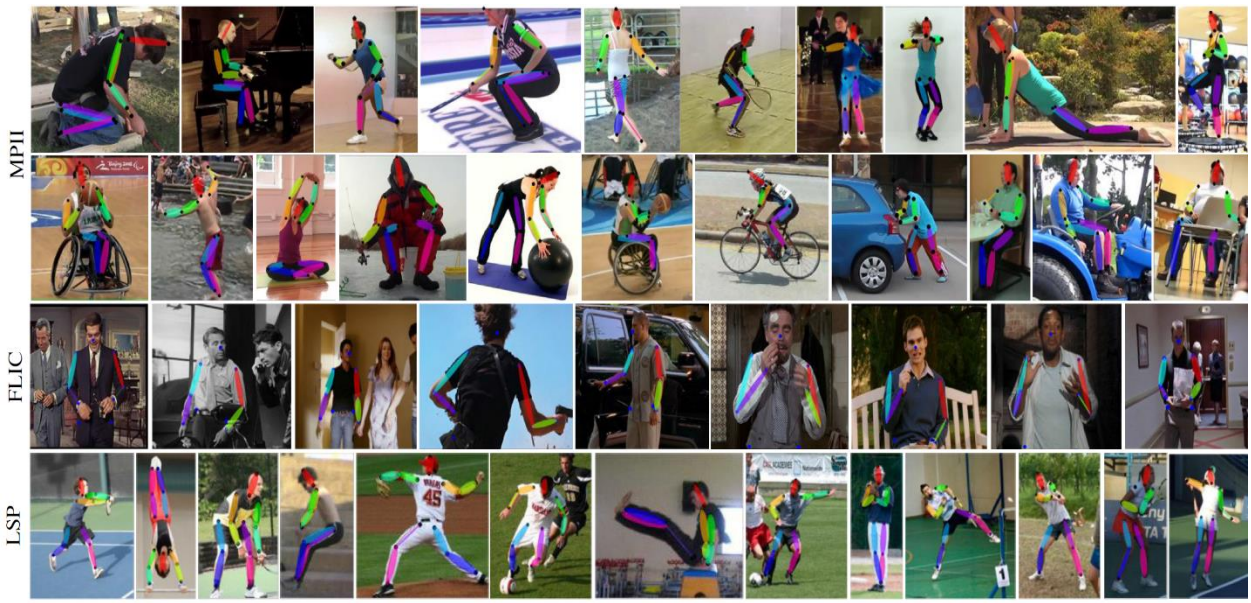
طراحی شبکه با دستیابی به یک گیرنده در لایه خروجی مرحله دوم شبکه هدایت می شود که به اندازه کافی بزرگ است که امکان یادگیری پیچیده و همبستگی های دوربرد بین قطعات را فراهم می آورد.



تصویر ۲۲ اثر اندازه ی گیرنده در دقت در دیتا ست FLIC

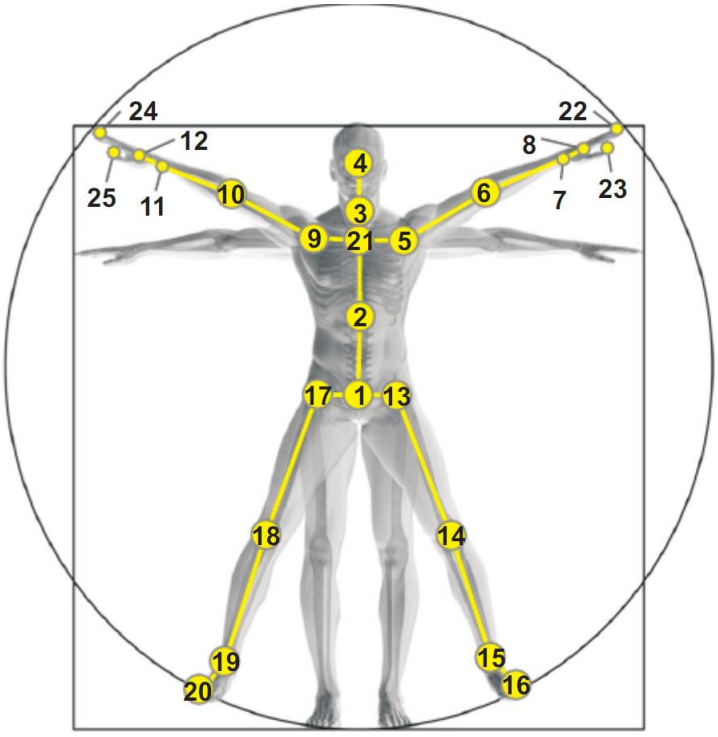
شبکه با گیرنده ی بزرگ در مدل سازی تعامل بین قسمت ها در محدوده ی دوربرد مؤثر است.افزایش اندازه ی گیرنده می تواند با افزایش تعداد لایه ها ی کانوولوشن انجام شود ولی امکان رویارویی با شیب های محو شده در حین آموزش وجود دارد. ما انتخاب می کنیم که برای رسیدن به پذیرا بزرگ به اندازه ی 8 برابر نقشه های حرارتی از لایه های متعدد کانوولوشن استفاده کنیم. ما متوجه شدیم که شبکه stride=8 به اندازه ی stride=4 خوب عمل می کند(حتی در دقت های بالا)،در حالیکه کار ما را برای رسیدن به زمینه های پذیرش بزرگتر آسانتر می کند.

۵.۱.۴ دیتاست ها



تصویر۲۳ نتایج روی دیتاست MPII,LSP,FLIC

این روش قادر به کنترل حالات غیر استاندارد و رفع ابهامات بین قطعات متقارن را برای انواع مختلفی از نمایشهای نسبی دوربین است.



تصویر ۲۴ نقاط دیتاست body25

۵.۲ نقاط تشخیص داده شده توسط پروژه openpose

جدول۳ *openpose keypoints*

نام مفصل	شماره مفصل
بینی	**1**
گردن	**2**
شانه چپ	**3**
آرنج چپ	**4**
مچ چپ	**5**
شانه راست	**6**
آرنج راست	**7**
مچ راست	**8**
Hip center	**9**
Hip left	**10**
زانو چپ	**11**
مچ پای چپ	**12**
Hip right	**13**
زانو راست	**14**
مچ پای راست	**15**
چشم چپ	**16**
چشم راست	**17**
گوش چپ	**18**
گوش راست	**19**
انگشت بزرگ پای راست	**20**
انگشت کوچک پای راست	**21**
پاشنه پای راست	**22**
انگشت بزرگ پای چپ	**23**
انگشت کوچک پای چپ	**24**
پاشنه پای چپ	**25**

۵.۳ بررسی کد پروژه ی openpose

۵.۳.۱ Poseparameters.cpp

خط 6-34 نگاشت شماره ی نقاط به قسمت های مختلف بدن برای دیتاست body-25 را بر عهده دارد.خط 253-256 نقاط دیتاست body_25 را به صورت وکتور در می آورد.خط 416-419 نقاط دیتاست را به صورت دو به دو نشان می دهد.خط

550-539 تابع دریافت نگاشت مربوط به قسمت های مختلف بدن است.خط 565-576 تابع دریافت مدل آموزش داده شده است.خط 578-590 تابع دریافت قسمت های بدن مربوط به شماره ی حالات از مدل است.خط 591-603 تابع دریافت قسمت های بدن به صورت دو به دو است.

poseParametersRender.cpp 5.3.2

خط 59-70 تابع دریافت بزرگنمایی مربوط به مدل است.خط 72-84 تابع دریافت رنگ مربوط به مدل است.خط 85-96 تابع دریافت رندر قسمت های بدن به صورت دو به دو است.خط 98-115 تابع دریافت تعداد عناصر برای رندر است.

poseRenderer.cpp 5.3.3

خط 9-61 نام قسمت ها را از روی ایندکس بر می گرداند.

PoseExtractor.cpp 5.3.4

خط 57-69 کپی نقشه های حرارتی را دریافت می کند.خط 83-95 تابع دریافت مفاصل مربوط به حالات است.خط 96-107 تابع دریافت امتیاز مربوط هر حالت است.

poseExtractorNet.cpp 5.3.5

از کودا استفاده می شود.خط 102-238 یک کپی از نقشه های حرارتی است.در خط 114 اندازه ی نقشه ی حرارتی دریافت می شود.در خط 117-118 حافظه به نقشه ی حرارتی تخصیص داده می شود.خط 126-151 مربوط به نقشه های حرارتی قسمت های مختلف بدن است.خط 138-140 بازه ی بزرگنمایی نقشه ی حرارتی را از [0,1] به [-1,1] تغییر می دهد.خط 141-144 بازه ی بزرگنمایی نقشه ی حرارتی را به [0,255] تغییر می دهد.خط 146-148 از مقدار های خارج از بازه اجتناب می کند.خط 152-189 نقشه های حرارتی مربوط به پس زمینه است.خط 167-181 مربوط به بازه ی نقشه های حرارتی و مشابه خطوط 126-150 است.خط 277-290 تابع دریافت مفاصل است.خط 291-304 تابع دریافت امتیاز مربوط به حالات است.خط 360-371 تابع پاک کردن نقاط و امتیاز ها است.

poseGPURenderer.cpp 5.3.6

در رندر روی پردازنده ی گرافیکی تعداد قسمت های بدن،ارتباط های دو به دوی قسمت های مختلف بدن،حالت کلی بدن و نقشه های حرارتی مورد نیاز است.خط 76-84 تخصیص حافظه برای رندر روی پردازنده ی گرافیکی است.خط 102-204 مربوط به رندر روی پردازنده ی گرافیکی است.خط 115-128 مفاصل را ترسیم می کند.خط 173-184 وابستگی بین نقاط را ترسیم می کند.

poseExtractorCaffe.cpp 5.3.7

خط 259-263 اطلاعات مربوط به نقشه های حرارتی را کپی می کند.خط 684-700 یک اشاره گر به نقشه های حرارتی را بر می گرداند.خط 702-718 اندازه ی نقشه های حرارتی را باز می گرداند.

۶ فصل ششم

مقایسه ی پروژه های سه بعدی و نتایج sterio

۶.۱ نمونه گیری

6.1.1 نمونه گیری با webcam

ضبط ویدیو های وب کم با debut video capture software یا iPi Recorder 4 امکانپذیر است.لازم به ذکر است که شرکت لاجیتک SDK مربوط به وب کم ها را در دسترس عموم قرار نداده است.

جدول۴ مشخصات Logitech c930e

ابعاد	بدون کلیپس و کابل: 24 × 94 × 29 میلی متر - همراه کلیپس : 71 × 94 × 43.3 میلی متر
وزن	162 گرم
طول	1.83 متر
ریموت	ندارد
وضوح تماس تصویری	Full HD
رزولوشن تماس تصویری	1920 × 1080پیکسل
رزولوشن ضبط ویدئو	1920 × 1080پیکسل
سرعت تصویربرداری	30فریم بر ثانیه
فرمت فشرده سازی (کدگذاری ویدیو)	H.264
کیفیت واقعی عکس	3مگاپیکسل
نوع لنز	Glass(شیشه)
محدوده زاویه دید لنز	90درجه و بالاتر
زاویه دید لنز	90درجه
نوع فوکوس	خودکار
قابلیت زوم	دارد - دیجیتال
مقدار زوم	4X
قابلیت چرخش	دارد - دستی
زاویه چرخش	عمودی
تصحیح خودکار نور	Rightlight™ 2(فناوری اصلاح خودکار و دستی نور در محیط های مختلف روشنایی)
میکروفون داخلی	دارد
نوع تولید و تکثیر صدای میکروفون	استریو - با قابلیت حذف نویز

چراغ نشانگر آماده به کار	نشانگرLED
قابلیت نصب بر روی سه پایه	دارد
ویژگی های وب کم	دارای درپوش حریم خصوص - 2 میکروفون با قابلیت دریافت صدای فراگیر - قابلیت نصب آسان بر روی لپ تاپ ، LCDو CRT و سه پایه - قابلیت خودکار تصحیح نور - هماهنگی با برنامه های BlueJeans, Broadsoft, LifeSize Cloud, Vidyo, and Zoom

در طی کار با این وبکم کد متلب مربوط به دریافت هر فریم از یک ویدیو را اجرا کردم. در این کد متلب VideoReader()برای خواندن ویدیو به کار رفته است.

hasFrame(vidObj) اگر فریم دیگری در ویدیو موجود باشد،true بر می گرداند.

readFrame(vidObj) یک فریم می خواند.

image(s(300).cdata) یک تصویر از فریم را نمایش می دهد.

movie(s,1,vidObj.FrameRate) برای نشان دادن یک ویدیو است.

به دلیل اینکه با این وب کم امکان ضبط همزمان دو فیلم نیست از آن در استریو استفاده نشده است.

6.1.2 نمونه گیری با دوربین VR

اولین قدم برای خواندن یک فایل ویدئویی، ایجاد یک شی VideoCapture است.

```
cam1 = cv.VideoCapture(1)
```

باید یک شی VideoWriter ایجاد کنیم. ابتدا باید نام فایل خروجی را با فرمت آن (مثلا: output.avi)مشخص کنیم. سپس، ما باید کد چهارگانه و تعداد فریم در ثانیه (FPS)را مشخص کنیم. در نهایت، اندازه فریم (رزولوشن) باید منتقل شود.برای فرمت mp4 باید از کد چهارگانه ی زیر استفاده شود:

```
out1 = cv.VideoWriter('outpy1.mp4',cv.VideoWriter_fourcc('M', 'P', '4', 'V'), 30,
(frame_width1,frame_height1))
```

با استفاده از کد VR می توانستیم دو ویدیو همزمان از دوربین لپ تاپ و یکه از دوربین های VR دریافت کنیم اما قادر به دریافت همزمان دو فیلم از دوربین VR نبودیم.پس از این دوربین در استریو استفاده نکردیم.

۶.۱.۳ نمونه گیری با کینکت RGB

برای نمونه گیری از دو دوربین کینکت ورژن 1 به صورت هم سطح و موازی استفاده شده است و فاصله[۳۳] ی بین آن دو نیز محاسبه شده است.از نرم افزار iPi Recorder 4 برای گرفتن هم زمان چند ویدیو از دوربین های کینکت ورژن 1و2 و دوربین

[33] T=0.28m

های دیگر مانند دوربین روی لپ تاپ می توان استفاده کرد.در این نرم افزار ابتدا پشت صحنه ارزیابی می شود و سپس فیلم ها ضبط می شود.

نمونه های گرفته شده شامل موارد زیر است:

- حرکت دادن دست ها به صورت افقی
- حرکت دادن دست ها به صورت عمودی
- حرکت دادن پاها به صورت افقی
- حرکت دادن پاها به صورت عمودی
- راه رفتن به صورت افقی
- راه رفتن به صورت عمودی
- نمونه دوم راه رفتن به صورت افقی
- نمونه دوم راه رفتن به صورت عمودی

خروجی نرم افزار iPi Recorder 4 یک فایل با پسوند iPiVideo. است که چند فیلم ضبط شده را در یک فایل نگهداری می کند.سپس نمونه های ویدیوی دوربین کینکت اول و دوربین دوم با نرم افزار iPi Mocap Studio از جدا می شوند.

6.2 اجرای پروژه سه بعدی videopose3d (Inference in the wild)

6.2.1 استنباط مختصات دو بعدی مفاصل با Detectron

برای استنباط تمام ویدیو های موجود در input_directory باید از این کد استفاده کنیم:

```
python tools/infer_video.py \
  --cfg configs/12_2017_baselines/e2e_keypoint_rcnn_R-101-FPN_s1x.yaml \
  --output-dir output_directory \
  --image-ext mp4 \
          --wts
https://dl.fbaipublicfiles.com/detectron/37698009/12_2017_baselines/e2e_keypoint_rcnn_R-101-
FPN_s1x.yaml.08_45_57.YkrJgP6O/output/train/keypoints_coco_2014_train:keypoints_coco_2014_valminusminival/generalized_rcnn/model_final.pkl \
  input_directory
```

خروجی در output_directory به صورت آرشیوهای numpy (.npz) خواهد بود.

۶.۲.۲ ایجاد دیتاست

سپس،در مسیر data یک دیتاست ایجاد می کنیم.

```
python prepare_data_2d_custom.py -i /path/to/detections/output_directory -o myvideos
```

کد بالا یک دیتاست به نام myvideos ایجاد می کند و آن را در فایل data_2d_custom_myvideos.npz ذخیره می کند.

۶.۲.۳ رندر و گرفتن خروجی به صورت آرشیو numpy

از این کد برای رندر کردن یک فیلم دلخواه استفاده می کنیم.خروجی فایل gif است که اسکت بدن را نمایش می دهد.هم چنین مختصات مفاصل در فایل های npy. ذخیره می شود.

```
python run.py -d custom -k myvideos -arc 3,3,3,3,3 -c checkpoint --evaluate
pretrained_h36m_detectron_coco.bin --render --viz-subject foothorizontalright.mp4 --viz-
action custom --viz-camera 0 --viz-video foothorizontalright.mp4 --viz-output
foothorizontalright.gif --viz-size 6
```

۶.۲.۴ تبدیل فایل های آرشیو numpy به فایل متنی

با استفاده از کتابخانه ی numpy و تابع np.load() می توان اطلاعات را از آرشیو numpy خواند و سپس مطابق با فرمت فایل های رکورد های کینکت که در پروژه های قبلی ایجاد شده است،یک فایل متنی ایجاد کرد.محاسبه ی زمان ذخیره شده در این فایل متنی با استفاده از شماره فریم و اینکه دوربین های کینکت با نرخ 30 فریم بر ثانیه فیلم را ضبط می کردند،انجام شده است.این کار امکان مقایسه ی نتیجه ی این پروژه را با کینکت ها فراهم می کند.

۶.۳ اجرای پروژه دوبعدی

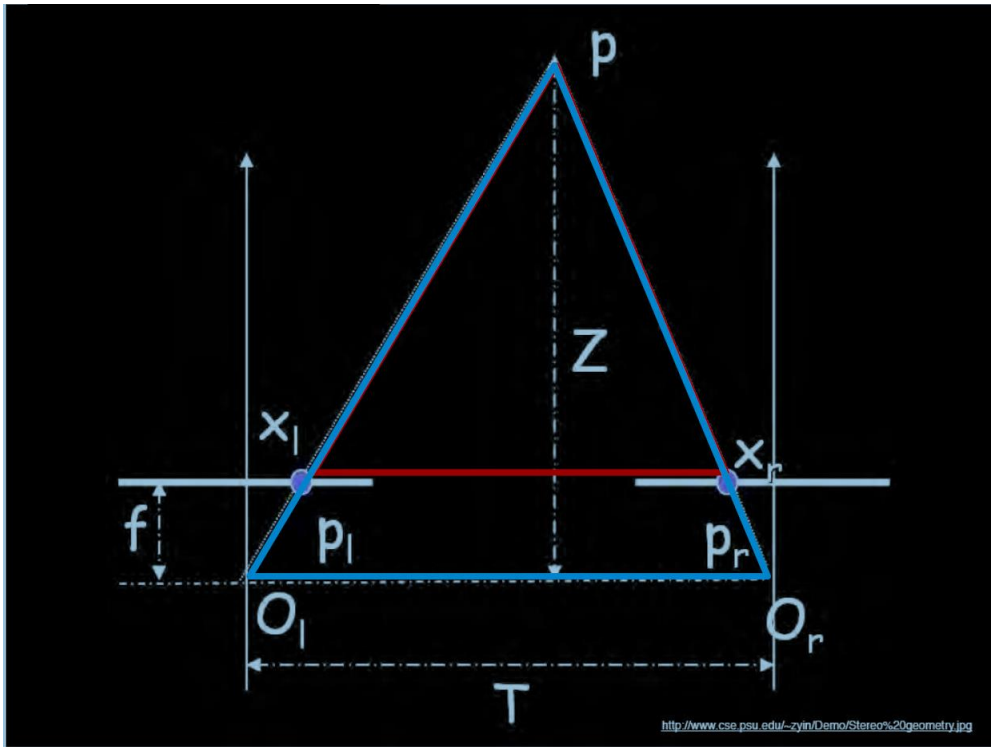
این کد را در powershell یا command prompt اجرا کنید تا فایل های json مربوط به مختصات سه بعدی هر فریم از فیلم در پوشه path ایجاد شود.

```
bin\OpenPoseDemo.exe --video finalvideo.mp4 --write_video foothorizontalright.avi --
write_images images --write_json path/
```

۶.۴ استریو

۶.۴.۱ شرح روش استریو

برای تخمین عمق از دو فیلم دو بعدی از روش مثلث سازی استفاده کردیم.در این روش دو دوربین کینکت به صورت موازی فیلم تهیه می کنند.با این روش می توانیم رابطه ای بین فاصله ی نقاط از یکدیگر و عمق نقطه ی مورد نظر در دنیای واقعی به دست آورد.



تصویر ۲۵ تخمین عمق نقطه ی p از روی دو تصویر به روش مثلث سازی

با این رابطه می توانیم disparity را برای هر نقطه در هر فریم از دو فیلم محاسبه کنیم.

$$disparity = x_r - x_l \qquad (1\text{-}5)$$

مثلث های متشابه در تصویر بالا (P,p_1,p_r) و (P,O_1,O_r) هستند.

$$\frac{T - disparity}{Z - f} = \frac{T}{Z} \qquad (2\text{-}5)$$

طبق این رابطه عمق نقاط به دست می آید:

$$Z = \frac{fT}{disparity} \qquad (3\text{-}5)$$

با استفاده از کتابخانه ی json و توابع jsf.read() و json.loads() مختصات دو بعدی نقاط را از فایل ها می خوانیم.سپس با استفاده از ضریب zarib= 0.0002645833 اعداد مختصات x,y,z را از پیکسل به متر تبدیل می کنیم و همین طور فاصله کانونی دوربین کینکت RGB را از 525 پیکسل به 0.13890625 متر تبدیل می کنیم.

۶.۴.۲ رسم مختصات دو بعدی نقاط با cv۲

با استفاده از کتابخانه ی cv2 و تابع line() می توان نقاط دوبعدی را رسم کرد.ابتدا باید مسیر یک تصویر را برای اینکه اسکت بدن در هر فریم روی آن رسم شود مشخص کنیم.رنگ و ضخامت خطوط نیز باید مشخص شود.باید نقاطی که به هم وصل

می شوند را در آرایه ی SkeletonConnectionMap ذخیره کنیم.با استفاده از cv.imread() تصویر مورد نظر را می خواند.برای هر فریم می توان اسکلت بدن را با این کد رسم کرد:

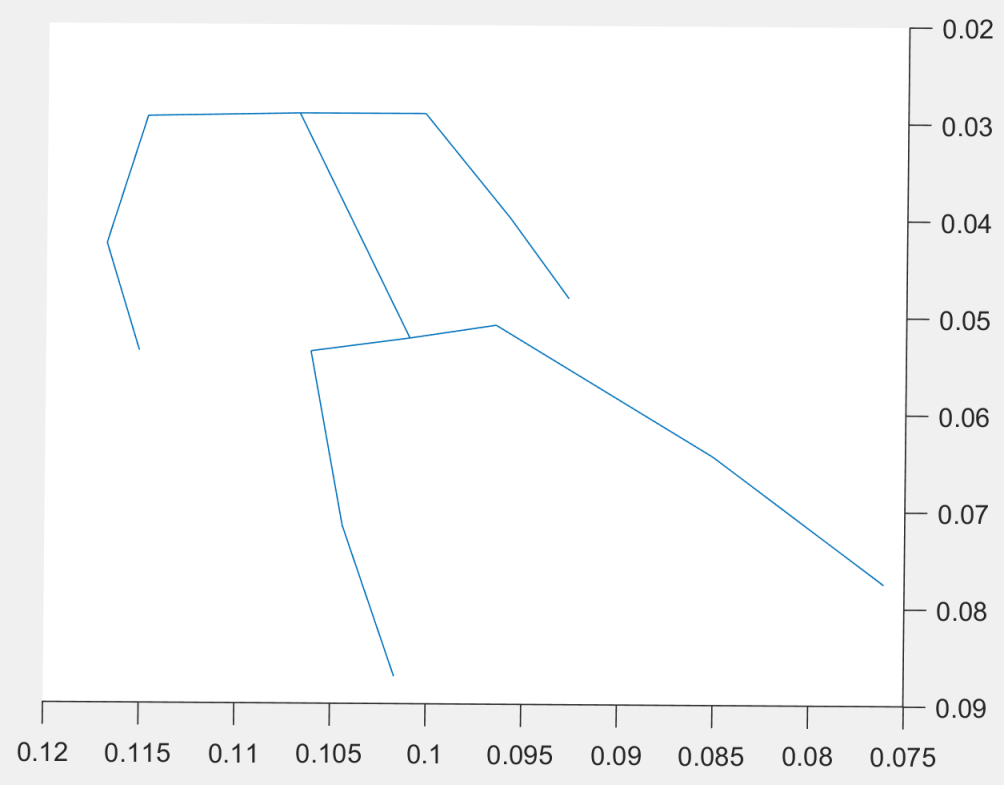
```
image = cv.line(image, start_point, end_point, color, thickness)
```

با استفاده از توابع cv.imshow() و cv.waitkey() می توانیم تصاویر را نشان دهیم.تابع cv.line() نقاط شروع و پایان را تنها به صورت عدد صحیح دریافت می کند.بنابراین باید نقاط را به عدد صحیح تبدیل کنیم.هم چنین باید بزرگنمایی شود(یک عدد ثابت را در مختصات نقاط ضرب می کنیم) تا تصاویر کوچک نباشند.

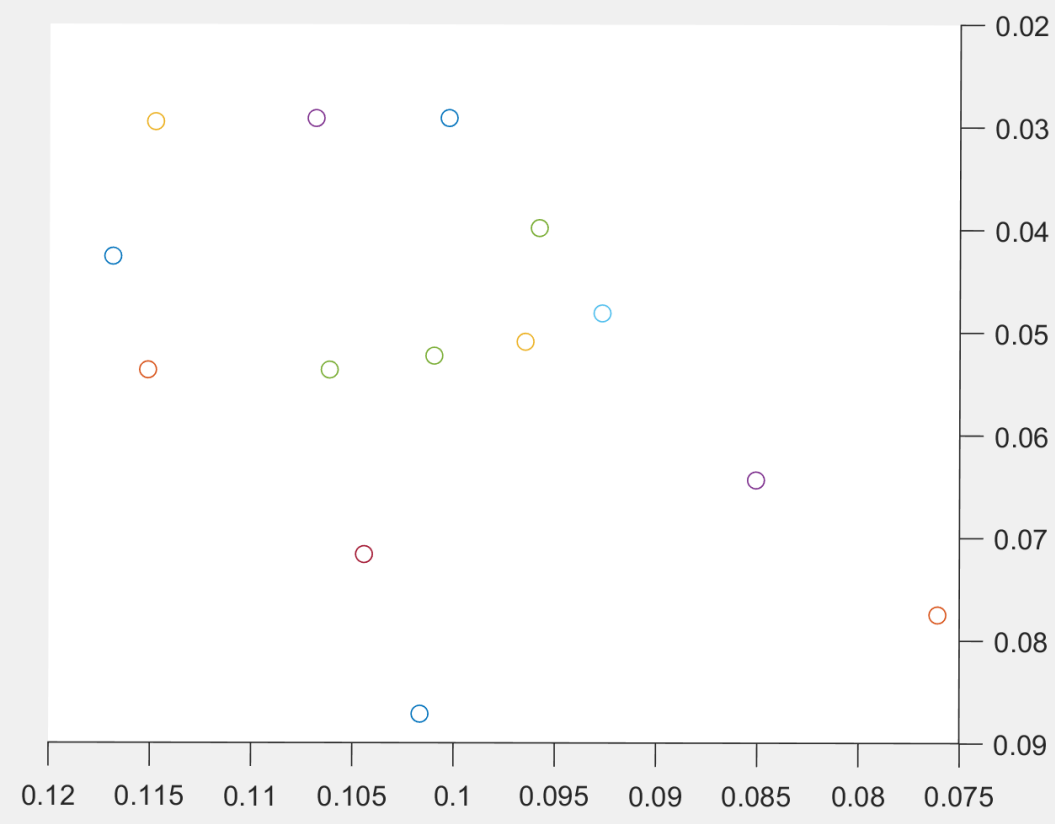
۶.۴.۳ رسم مختصات سه بعدی نقاط با متلب

با استفاده از گزینه ی import data در Home در متلب می توانیم یک فایل متنی را به صورت Numeric Matrix در متلب وارد کنیم.این بار نیز SkeletonConnectionMap را ایجاد می کنیم و با استفاده از تابع line(x,y,z) خطوط را رسم می کنیم.هم چنین hold on هم باید فعال باشد تا همه ی خطوط را با هم نمایش دهد.

عبارت drawnow limitrate را باید در پایان رسم هر فریم بنویسیم تا هر فریم به صورت جداگانه رسم شود یا اینکه شکل کلی رسم شده متشکل از همه ی فریم ها را مشاهده کنیم.عبارت clf برای اینکه پس از رسم هر فریم صفحه پاک شود،مورد نیاز است.همچنین باید مدت زمان کوتاهی پس از رسم هر فریم صبر شود و سپس فریم بعدی رسم شود.



تصویر ۲۶ فریم ۲۵ از foothorizontal



تصویر ۲۷ نقاط فریم ۲۵ از foothorizontal

۶.۵ نتایج مقایسه پروژه سه بعدی با نقاط حاصل از استریو

نقاط پروژه سه بعدی videopose3d دقت بسیار بالایی دارند.در خطوط رسم شده در متلب در برخی فریم ها خطاهایی وجود دارد ولی شکل کلی حرکت توسط استریو نیز تشخیص داده است.در این بین برخی خطوط دقیق رسم نشده اند.

۶.۶ پیشنهاد برای آزمایش‌های بعدی

در صورتی که تمایل داشته باشید پروژه های یادگیری عمیق ادامه دهید،پیشنهادهای مختلفی را می‌توان مطرح کرد که از جمله‌ی آن‌ها می توان به این موارد اشاره کرد:

1. اضافه کردن کلاس‌های دیگر مثل نشستن، ایستادن،حرکت های فرد در حال انتظارو حرکت های نامتقارن مانند حرکت با یک دست در جیب و ....
2. انجام پروژه های یادگیری عمیق برای کاربرد های دیگر مانند مواردی که در بخش 1.3.1 ذکر شد.
3. نتایج پروژه ی videopose3d با نتایج کینکت ورژن 1 و2 مقایسه شود.
4. از پروژه detectron2 که به دلیل منسوخ شدن caffe برای جایگزینی detectron ایجاد شده است برای تخمین دو بعدی مفاصل استفاده شود.
5. از روش های دیگر استریو استفاده شود.

۷ منابع

[1] 3*D human pose estimation in video with temporal convolutions and semi-supervised training*
Dario Pavllo ETH Z¨urich
Christoph Feichtenhofer Facebook AI Research
David Grangier Google Brain
Michael Auli Facebook AI Research

[2] *https://github.com/facebookresearch/VideoPose3D*

[3] *https://github.com/CMU-Perceptual-Computing-Lab/openpose*

[4] *https://github.com/facebookresearch/Detectron*

[5] *https://github.com/facebookresearch/detectron2*

[6] *Convolutional Pose Machines*
Shih-En Wei
shihenw@cmu.edu
Varun Ramakrishna
vramakri@cs.cmu.edu
Takeo Kanade
Takeo.Kanade@cs.cmu.edu
Yaser Sheikh
yaser@cs.cmu.edu
The Robotics Institute
Carnegie Mellon University