



گزارش علمی پروژه پایانی درس مباحث ویژه ۲

مهدیه دست نبوی

دانشگاه شهید رجایی

Mahdie.dn76@gmail.com

زهرا محمدی

دانشگاه شهید رجایی

Smohammadi864@yahoo.com

عنوان : تشخیص و شناخت حرکات دست با آموزش مدل شبکه عصبی کانولوشنالی

چکیده

تشخیص حرکت یک مشکل بسیار جالب در جامعه کامپیوتری برای مدت زمان طولانی است ، به این دلیل که جدا کردن شیء پیش زمینه از یک پس زمینه کلیدی برای یک مسئله چالش برانگیز در زمان حاضر است. علت واضح تر این است که شکاف معنانشناسی در زمانی که یک انسان به یک تصویر نگاه می کند و یک کامپیوتر به یک تصویر مشابه نگاه می کند وجود دارد ، برای یک کامپیوتر، تصاویر ماتریس های ۳ بعدی هستند. به همین دلیل، مشکلات بینایی کامپیوتری و تفاوت آن با دید انسان یک چالش است .

مقدمه

ما قصد داریم حرکات دست را از یک ویدئو تشخیص دهیم. برای شناختن و تشخیص این حرکات از یک توالی ویدئویی زنده، ابتدا باید منطقه دست را خارج کنیم تا تمام قسمتهای ناخواسته را در دنباله ویدئو حذف کنیم. بعد از جدا کردن و شناسایی منطقه دست، سپس انگشتان نشان داده شده در ترتیب ویدئویی را بر اساس تعداد انگشتان حساب می کنیم. اولین گام در شناختن gesture دست، بدیهی است که منطقه دست را با حذف تمام قسمت های ناخواسته در توالی ویدئو پیدا کنید.

این ممکن است در ابتدا سخت به نظر برسد اما با استفاده از پایتون و OpenCV بسیار ساده تر خواهد بود. در اینجا ما به بحث در مورد چند مورد تشخیص و شناخت حرکات دست میپردازیم و آن ها را با پایتون پیاده سازی کرده ایم و در آخر یک محیط گرافیکی برای این موارد با کتابخانه tkinter در پایتون ایجاد کرده ایم که خروجی ها را ساده تر ببینیم .

تشخیص حرکات دست

۱- تغییر جایگاه ماوس کامپیوتر

در این جا برای انجام این عمل از یک `hand.xml` که برخی ویژگی های مربوط به دست را داراست استفاده شده است که با `CascadeClassifier` در `opencv` صدا زده شده است . این مورد تنها تشخیص دست را به همراه دارد و با کشیدن یک مستطیل با طول و عرض مشخص در اطراف دست تشخیص داده شده کار میکند . همچنین در این جا برای هر حرکت در راستای `X` و `Y` خطی قرمز روی صفحه کشیده میشود که به مرور زمان قبلی ها کم رنگ میشوند و با توجه به آن گفته میشود که این حرکت در راستای بالا ، پایین ، چپ ، راست ، بالا چپ ، بالا راست ، پایین چپ یا پایین راست است و بالای صفحه باز شده برای وبکم نوشته میشود .

۲- تغییر میزان صدای کامپیوتر

این کار با استفاده از `subprocess.call` انجام شده است که از آن برای راه اندازی زیر فرایندها استفاده میشود. این کد فقط برای سیستم های لینوکسی قابل اجراست و برای اجرای آن در سیستم های ویندوزی باید از اجرای زیر فرآیندها در ویندوز استفاده کرد . در اینجا در ابتدا فریم را میگیرد و روی آن تغییراتی انجام میشود که کار با آن راحت تر شود . مثلاً برای `threshold` از روش بهینه ی `Otsu's Binarization` استفاده شده است . در مرحله بعدی بزرگترین کانتور پیدا شده است و یک مستطیل دور آن کشیده شده است . در ادامه یکسری کار ها برای پیدا کردن زوایای مختلف دست با تکنیک `cosine` ، تشخیص انگشتان دست و کشیدن کانتور ها انجام شده است . در آخر با توجه به تعداد انگشتانی که در فریم دیده میشود صدا کم و زیاد میشود .

شناخت حرکات دست و آموزش مدل کانولوشنالی

۱- شناخت ۳ حرکت دست

دیتاست این ۳ حرکت دست بدین صورت است :



در اینجا امکانی برای تولید دیتاست بیشتر به همین شکل دیتاست فعلی وجود دارد. یعنی از وبکم حرکت دست شما را در فریم های مختلف استخراج میکند و روی آن پردازشی انجام میشود تا به شکل قابل آموزشی در شبکه مانند دیگر دیتا شبیه شود.

دیتاست فعلی شامل ۱۰۰۰ نمونه در هر کلاس برای آموزش و ۱۰۰ نمونه برای داده تست در نظر گرفته شده است. پس با دیتاست فعلی ۳۰۰۰ نمونه برای داده آموزش و ۳۰۰ نمونه برای داده تست داریم.

برای تفکیک کلاس ها در داده آموزش و داده تست به این گونه عمل شده است که به ازای هر کدام از ۱۰۰۰ نمونه اول که مربوط به یک کلاس هستند در یک بردار آرایه $[0, 1]$ اضافه کردیم و به ازای هر ۱۰۰۰ نمونه دوم آرایه $[0, 1]$ اضافه کردیم و به همین ترتیب برای کلاس سوم و هرچند تا کلاس دیگر.

همین روند را برای داده های تست نیز تکرار کرده ایم که در واقع نوعی نشانه گذاری برای تفکیک کلاس ها است.

معماری شبکه یک مدل کانولوشنالی ساده است که از ۷ لایه کانولوشن ۲ بعدی و به ازای هر کدام یک لایه max pooling و در انتها ۲ لایه fully connected تشکیل شده است که چون ۳ کلاس در اینجا داشته ایم لایه آخر softmax ۳ تایی قرار داده ایم.

در آخر آموزش را با ۵۰ اپیاک شروع کرده ایم که به accuracy نسبتاً خوبی برابر با 0.9567 رسیدیم.

حالا کدی برای تست این مدل بصورت زنده داریم که با استفاده از آن مشاهدات خوبی دیدیم و این نشان میدهد که مدل خوب آموزش دیده است.

۲- شناخت اعداد با حرکات دست

دیتاست اعداد با حرکات دست بدین صورت است:

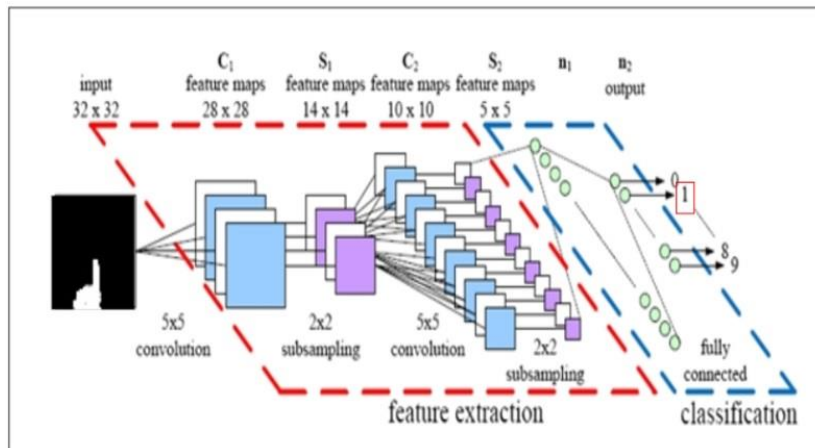


در کل ۱۰ کلاس داریم از ۱ تا ۹ اعداد و یک کلاس که اگر هیچ دستی در تصویر نباشد آن را برگرداند. در اینجا نیز مانند حالت قبل برای هر کلاس ۱۰۰۰ نمونه آموزش و در کل ۲۰ درصد از هر کلاس را برای تست در نظر گرفته ایم. در اینجا نیز امکان جمع آوری دیتا بصورت مطابق با مدل و با فرمت دیتاست اصلی مثل حالت قبلی وجود دارد.

خیلی سعی کردیم این دیتاست را با دیتاست مدل قبلی تلفیق کنیم و روی همه آنها مدل یکسانی آموزش دهیم ولی متأسفانه مدل قبلی روی این دیتاست نتیجه ی مطلوبی نداشت و مجبور به آموزش مدلی دیگر روی این دیتا شدیم. مدل آموزش داده شده روی این دیتا کمی عمیق تر از مدل قبلی است زیرا تعداد کلاس ها افزایش پیدا کرده و مدل قبلی روی این دیتا خوب عمل نمیکرد.

در اینجا مدل باز هم به خوبی مدل قبلی تشخیص نمیدهد ولی روی بعضی اعداد نسبتاً خوب عمل میکند.

شبکه نمایی به صورت زیر دارد :



۳- آموزش مدل کانولوشنالی برای تشخیص یک جمله

تا اینجا فهمیدیم که میتوانیم داده های دلخواه با حرکات دست مختلف و دلخواه تولید کنیم پس به راحتی میتوانیم مدل های کانولوشنالی ساده یا عمیق را روی داده خودمان آموزش دهیم و با تشخیص حرکات دست جملاتی ایجاد کنیم . کاری که به تازگی روی حرکات دستی که مخصوص صحبت ناشنویان است انجام شده است که میتوان با تشخیص این حرکات افراد عادی هم صحبت های ناشنویان را بفهمند .

در اینجا میخواهیم جمله ساده I love you را با تشخیص حرکات دست تولید کرد .

دیتاست به صورت زیر است که کاملاً در قالب دیتاست های قبلی است و میتوان گفت یکی از بهترین پیش پردازش هایی که میتوان روی فریم ها زد همین قالب فریم های بدین گونه است :



باز هم دز ابتدا خواستیم با ۲ مدل قبلی آموزش را انجام دهیم ولی نتیجه خوبی نمود و باید مدلی دیگر آموزش داده میشد . در اینجا تقریباً همان مدل قبلی را با تعداد لایه های کمتر آموزش دادیم و به نتیجه مطلوب رسیدیم . بدین ترتیب میتوان هر داده ای را با مفاهیم دلخواه آموزش داد و حرکات دست را تشخیص داد .

محیط گرافیکی برای دسترسی ساده تر به تشخیص و شناخت حرکات دست گفته شده

در ادامه نیاز به محیطی بود تا بتوان به سادگی موارد گفته شده را تست کرد که برای اینکار از کتابخانه tkinter در پایتون که قابلیت های زیادی را در اختیار ما قرار میدهد استفاده کردیم .

ما در اینجا به طوری عمل کردیم که با زدن هر دکمه زیر روال مربوط به آن اجرا شود و بعد از اتمام کار به محیط اولیه برگردانده میشویم و میتوانیم زیر روال دیگری را اجرا کنیم .

نمایی از این محیط گرافیکی بدین صورت است :

