

دانشگاه صنعتي امیرکبیر  
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

گزارش کارآموزی

موضوع کارآموزی:

**Video Swin Transformer**

|  |  |
| --- | --- |
| نام استاد کارآموزی: | دکتر صاحب الزمانی |
| نام دانشجو: | محمدرضا قادری عمودیزج |
| شماره دانشجویی: | 9627057 |

**تابستان 1399**

**تقدیر و تشکر**

بدینوسیله مراتب قدردانی و امتنان خود را خدمت،

جناب آقای مهندس حسین یعقوب زاده، سرپرست تیم توسعه دهنده پژوهشگاه توسعه فناوری‌های پیشرفته، بابت آموزش‎ها و زحمات بی‌دریغشان، تمامی کارکنان و همراهان بنده در پژوهشگاه توسعه فناوری‌های پیشرفته ،

ابراز و از تمامی زحمات آنان تشکر می‌نمایم.

محمدرضا قادری عمودیزج

شهریور 1400

چكيده

امروزه با توجه به نیازهای جامعه از نظر امنیتی و توسعه هوش مصنوعی و همینطور شناسایی رفتارهای افراد در جامعه از نظر بازدید ویدیوها و تصاویر در شبکه‌های اجتماعی و حتی استفاده از شناسایی این رفتارها در ربات‌های انسان نما برای پاسخ به حرکات پیش آمده نیاز به یادگیری و تشخیص رفتارهای تصاویر و ویدیو شده است. در اواخر قرن بیستم با توجه به رونق هوش مصنوعی از این تکنولوژی برای بررسی و تجزیه وتحلیل خودکار مبنی بر تصویربرداری استفاده شد که بینایی ماشین[[1]](#footnote-1) نامیده شد، که روز به روز در حال گسترش هست.

در این گزارش، پس از تحلیل این روش برای بررسی ویدیوها به صورتی که بتوان با بینایی ماشین به تشخیص نوع رفتاری آنها بیانجامد استفاده می‌کنیم، و نوع رفتاری و انجام تحلیل ویدیوهای، نحوه‌ی استفاده از دیتاست و ابزارها و کتابخانه‌های لازم رو بررسی می‌کنیم.

واژه‌های کلیدی:

هوش مصنوعی، پردازش تصویر[[2]](#footnote-2)، بینایی ماشین، تشخیص رفتار ویدیو، یادگیری عمیق.

فهرست مطالب

[فصل 1: مقدمه](#_Toc69291746) 0

[1-1- مقدمه 0](#_Toc69291747)

[1-2- کارهای مرتبط 0](#_Toc69291748)

[1-3- معماری شبکه 0](#_Toc69291749)

[1-4- جمع‏بندی 0](#_Toc69291750)

[فصل 2: مجموعه دادگان 1](#_Toc69291746)

[2-1- مقدمه 2](#_Toc69291747)

[2-2- نحوه دسترسی 2](#_Toc69291748)

[2-3- جزئیات مجموعه دادگان 2](#_Toc69291749)

[2-4- جمع‏بندی 2](#_Toc69291750)

[فصل 3: راه‏اندازی کد به صورت برخط (Online) 3](#_Toc69291751)

[3-1- مقدمه 4](#_Toc69291752)

[3-2- نصب پکیج‏ها و پیش‏نیازها 4](#_Toc69291753)

[3-3- نحوه دسترسی 4](#_Toc69291754)

[3-4- راه‏اندازی سریع 4](#_Toc69291755)

[3-5- بررسی جزئیات 5](#_Toc69291756)

[3-6- جمع‏بندی 6](#_Toc69291757)

[فصل 4: راه‏اندازی کد به صورت برون خط (Offline) 7](#_Toc69291758)

[4-1- مقدمه 8](#_Toc69291759)

[4-2- نحوه دسترسی 8](#_Toc69291760)

[4-3- نصب پیش‏نیازها به صورت Offline 8](#_Toc69291761)

[4-4- راه‏اندازی سریع 8](#_Toc69291762)

[4-5- بررسی جزئیات کد 8](#_Toc69291763)

[4-6- جمع‏بندی 9](#_Toc69291764)

[فصل 5: جمع‏بندی و پیشنهادات 12](#_Toc69291765)

[5-1- مقدمه 13](#_Toc69291766)

[5-2- جمع‏بندی کلی و پیشنهادات 13](#_Toc69291767)

[5-3- کارهای آینده 13](#_Toc69291768)

[5-4- جمع‏بندی 13](#_Toc69291769)

[مراجع 14](#_Toc69291770)

فهرست اشکال

[شکل (1-1) تصویر نمونه 2](#_Toc34822082)

**فهرست جداول**

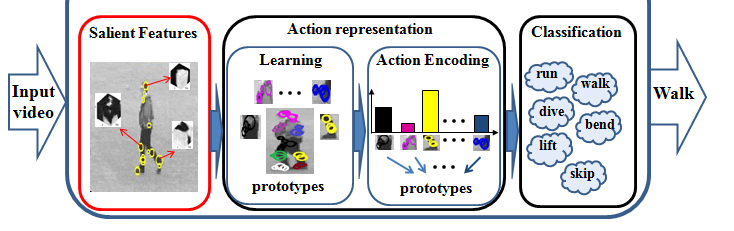
[جدول (1-1) جدول نمونه 2](#_Toc34821840)

1. مقدمه
   1. مقدمه

با توجه به پیشرفت روز افزون هوش مصنوعی و باور محققان و دانشمندان به اینکه در آینده بسیار نزدیک در اکثر قسمت‌های زندگی انسان‌ در ارتباط خواهد بود. یکی از مهمترین کارهای که هوش مصنوعی بایستی در آینده قادر به انجام آن باشد پیش بینی دقیق و انجام عکس العمل یا حرکتی متناسب با آن است، با توجه به اینکه هوش مصنوعی یک نمونه از مغز انسان می‌باشد و از جهتی که انسان برای این که عکس العملی انجام دهد به صورت معمول نیاز دارد در ابتدا آن حرکتی که اتفاق افتاده را ببیند و آنرا متوجه شود (یعنی بداند این عملی که انجام شده چه بوده و چه حرکتی بایستی در قبال آن انجام داد)، پس ما در هوش مصنوعی نیز بایستی بتوانیم متوجه عمل‌هایی که اتفاق می‌افتد باشم و آن‌ها را از هم متمایز قرار دهیم و برای آنها عکس العملی تدارک ببینیم، پس برای همین موضوع مثل انسان‌ها باید ببینیم و روی تصاویر پردازش‌‌*هایی انجام بدهیم.*

*به رسمیت شناختن فعالیت های ویدئویی ، اگرچه یک کار نوظهور است ، اما به دلیل اهمیت کاربردهای روزمره آن ، موضوع تلاش های مهم تحقیقاتی بوده است. نظارت با دوربین های ویدئویی می تواند پیشرفت های زیادی در این زمینه داشته باشد. در زمینه روباتیک ، وظایف ناوبری مستقل یا تعامل اجتماعی نیز می تواند از دانش استخراج شده از ضبط ویدئوی زنده استفاده کند. یکی از این موارد تشخیص فعالیت برای توانبخشی ، تشخیص عملی است که بیماران انجام می دهند و توانایی تشخیص درست یا نادرست بودن آن را دارند. یکی از تکنیک های اصلی مورد استفاده برای تشخیص فعالیت ، بینایی رایانه ، یعنی تشخیص فعالیت مبتنی بر ویدئو است. ویژگی های تصویری بصری اطلاعات اولیه را برای رویدادها یا اقدامات ویدیویی ارائه می دهد.*

*برای مثال در مغز انسان اطلاعات به صورت زیر وارد می‌شود ونتیجه حاصل می‌شود:*

**

***شکل 1-1: روند کلی تشخیص حرکات***

*که با توجه به این پروسه برای تشخیص حرکات توسط هوش مصنوعی نیز از این حالت کلی استفاده می‌شود.*

* 1. بینایی ماشین و پردازش تصویر

پردازش تصویر ، تجزیه و تحلیل و بینایی ماشین بخشی هیجان انگیز و زیبایی از علوم رایانه مدرن است. به دنبال وقوع تحول عظیم در دهه هفتاد ، دهه هشتاد با رشد چشمگیر برنامه‌های کاربردی فعال مشخص شد. سنجش از دور ، تشخیص فنی ، راهنمای خودروهای خودران و تصویربرداری پزشکی سریع‌ترین نقاط در حال توسعه هستند. این پیشرفت را می توان در افزایش تعداد محصولات نرم افزاری و سخت افزاری موجود در بازار و همچنین در تعدادی از دوره های پردازش تصویر دیجیتال و بینایی ماشین ارائه شده در دانشگاه های جهان مشاهده کرد.

* 1. ترانسفورماتور چرخش تصویری[[3]](#footnote-3)

ترانسفورماتور چرخش تصویری که از یک جانبداری استقرایی در ترانسفورماتورهای ویدئویی حمایت می کند ، که منجر به یک مبادله سرعت و دقت بهتر در مقایسه با رویکردهای قبلی می شود که توجه خود را در سطح جهانی حتی با عوامل مکانی و زمانی محاسبه می کند. موقعیت معماری ویدئویی پیشنهادی با اقتباس از ترانسفورماتور چرخشی که برای حوزه تصویر طراحی شده است ، در حالی که همچنان از قدرت مدل های تصویری از پیش آموزش دیده استفاده می کند ، درک می شود.

* 1. نتیجه گیری

حال که با توجه به مفاهیم اولیه و نوع کلی تجزیه و تحلیل منابع برای بدست آوردن طرحی کلی و تشخیص آن منبع هست می‌توانیم با استفاده از کتابخانه‌ها و ابزارهای موجود در این زمینه‌ها آنها را به همان گونه که مغز انسان به دنبال تجزیه و تحلیل آن است دنبال کرده و به جوابی برای آن برسیم.

1. معرفی محل کارآموزی
   1. تاریخچه

میزان توسعه یافتگی کشورها در جهان امروز بر مبنای علم و دانش استوار است. پژوهش و تولید علم و انواع فناوری، از مهم ترین عناصر رشد و توسعه ی اقتصادی، اجتماعی، فرهنگی و سیاسی هر کشوری به حساب می آید ایران اسلامی حداقل از دو دهه ی اخیر، گام های استواری را در مسیر توسعه و عزت برداشته و با توجه به همین بالندگی سرزمین پر افتخارمان در مسیر توسعه و عزت، مؤسسان پژوهشگاه توسعه فناوری های پیشرفته خواجه نصیرالدین طوسی در سال 1371 با تأسیس یک مرکز تحقیقاتی، از پیشگامان حرکت در این مسیر سرنوشت ساز بوده اند. در سال 1381 با تلاش شبانه روزی و با شعار "ما می توانیم" و جذب استادان و نخبگان از دانشگاه‌های تراز اول کشور، این مرکز تحقیقاتی با اخذ مجوز از وزارت علوم، تحقیقات و فناوری، تبدیل به پژوهشکده شد تا بتواند سهم بیشتری در تولید علم در کشور داشته باشد. با گسترش فعالیت های روزافزون این پژوهشکده و رشد تقریباً بیست برابری در جذب محققان و پژوهشگران، در سال 1391 با تبدیل پژوهشکده به پژوهشگاه برگ زرین دیگری بر افتخارات این پژوهشگاه افزوده شد. پژوهشگاه توسعه فناوری های پیشرفته خواجه نصیر الدین طوسی در قالب پنج پژوهشکده و یک مؤسسة فرهنگی هنری و با به کارگیری حدود 200 نفر از دانش آموختگان و فارغ التحصیلان برتر دانشگاه های کشور مشغول به فعالیت است.



**شکل 1-2: روند توسعه پژوهشکده**

* 1. معرفی کلی شرکت

راه و هدف ملت ایران، الگو‌شدن در جهات گوناگون در دنیا و تمدّن‌سازی نوین اسلامی است که یکی از پایه‌های اصلی این تمدّن، علم و فناوری بوده و ایران اسلامی تاکنون گام‌های استواری در مسیر اقتدار و عزّت امت اسلامی با توانمندی در علم و فناوری برداشته است. مؤسسان پژوهشگاه توسعه فناوری‌های پیشرفته خواجه‌ نصیرالدین طوسی در سال 1371 با تأسیس یک مرکز تحقیقاتی از پیشگامان حرکت در این مسیر سرنوشت‌ساز بوده‌اند و پس از یک دهه فعالیت علمی با جذب فرهیختگان و پژوهشگران متعهد و متخصص کشور توانستند با اخذ مجوز پژوهشکده از وزارت علوم، تحقیقات و فناوری، نقش مؤثری در بالندگی سرزمین پر افتخارمان داشته باشند. در سال 1391، با گسترش فعالیت‌های علمی و فناوری و رشد بیست برابری جذب محققان و پژوهشگران، وزارت علوم، تحقیقات و فناوری با تبدیل وضعیت پژوهشکده به پژوهشگاه موافقت کرد و برگ زرین دیگری بر افتخارات این مرکز علمی افزوده شد. این پژوهشگاه در جهت انجام طرح های پژوهشی و پروژه‌های فناوری مورد نیاز سازمان‌ها و شرکت‌ها ی ایرانی با استفاده از قابلیت‌ها و ظرفیت‌های در دسترس کشور در قالب پنچ پژوهشکده زیر مشغول به فعالیت می باشد :

* پژوهشکده پردازش داده
* پژوهشکده افتا (امنیت فضای تبادل اطلاعات )
* پژوهشکده فاوا (فناوری ارتباطات و اطلاعات )
* پژوهشکده کیمیازی
* پژوهشکده هوافضا
  + 1. پژوهشکده پردازش داده

با توجه به موضوع کارآموزی ما به صورت مستقیم با پژوهشکده پردازش داده ارتباط داریم کمی بیشتر در مورد این پژوهشکده بدانیم:

موضوع ثبت ، پردازش و تحلیل علائم زیستی (مانند صوت ، تصویر و علائم مرتبط با فعالیت های فیزیولوژیکی انسان) یکی از زمینه های بسیار گسترده و مطرح در بین شاخه های مختلف پردازش علائم است . تشکیل این پژوهشکده ، گام اول در جهت ایجاد بستر مناسب جهت فعالیت در این زمینه و تعریف و اجرای پروژه های کاربردی مناسب در این راستا بوده است . با توجه به جوانب مختلف تحقیقات مطرح در این زمینه ، هدف نهایی از ایجاد این پژوهشکده ، تشکیل گروه های کارآمد و هماهنگ از متخصصان گرایش های مختلف مرتبط با این مساله بوده است که توانایی لازم برای توسعه ، گسترش و بومی سازی علوم و فناوری پردازش علائم زیستی در داخل کشور را داشته باشد.

**توانمندی های پژوهشکده**

* برگزاری دوره های آموزشی و تخصصی
* برآورد نيازمندي‌ها، تعريف، امكان‌سنجي و اجراي پروژه‌ها در زمينه‌هاي مرتبط با پردازش صوت و گفتار، پردازش تصویر و پردازش علائم حیاتی
* ارائه مشاوره‌هاي علمي، پژوهشي و تخصصي در زمينه‌هاي مرتبط با پردازش صوت و گفتار، پردازش تصویر و پردازش علائم حیاتی
* تولید نرم افزارهای کاملاً عملیاتی در زمینه پردازش صوت و گفتار، پردازش تصویر و پردازش علائم حیاتی

خود این پژوهشکده به سه زیر گروه صوت، تصویرو علائم زیستی تقسیم می‌شود که پژوهش زیر تحت گروه تصویر به انجام رسید.

* + 1. گروه تصویر

، گسترش و بومی‌سازی علوم و فناوری­ های پردازش تصویر در داخل کشور از اهداف کلان این گروه تحقیقاتی می‌باشد. تلاش ما در فراهم نمودن زمینه­ ی مناسب برای پیش برد فعالیت‌های پژوهشی در زمینه پردازش تصویر و ویدئو به منظور ارتقای سطح علمی و کمک به تلفیق علم و عمل در این حوزه است. در این راستا، برآورد نیازمندی‌ها، ارائه طرح‌ها، تعریف و اجرای پروژه‌ها در زمینه‌های مختلف علوم و فناوری‌های مذکور، جهت نیل به اهداف کلان، عمده‌ترین فعالیت‌ها و اهداف جزئی این گروه تحقیقی را تشکیل می دهند. این گروه تلاش دارد تا پاسخگوی حداقل نیازهای حال و آینده‌ی کشور در زمینه‌های علوم و فناوری های پردازش تصویر باشد.

* 1. افتخارات

پژوهشگاه توسعه فناوری‌های پیشرفته موفق به کسب افتخارات زیر در طی سالیان گذشته شده است:

* رتبه سوم تحقيقات کاربردي يازدهمين جشنواره بين المللي خوارزمي در سال 1376
* انتخاب به عنوان پژوهشکده ی برتر از سوي وزارت علوم، تحقيقات و فناوري در سال 1384
* اخذ لوح و تنديس زرين از سومين دوره ی جشنواره و نمايشگاه بين‌المللي رسانه ­هاي ديجيتال(وزارت فرهنگ و ارشاد اسلامي) در سال 1388 به خاطر نرم افزار شنوا
* اخذ لوح و تنديس زرين از چهارمین دوره ی جشنواره و نمايشگاه بين‌المللي رسانه ­هاي ديجيتال (وزارت فرهنگ و ارشاد اسلامي) در سال 1388 به خاطر نرم افزار شیوا «گویا»
* اخذ لوح و تنديس از ششمين كنفرانس بين المللي انجمن رمز ايران در سال 1388
* شایسته تقدیر كتاب ابزارهای مدیریت دانش در فصل پاییز 1389 از سوی خانه کتاب، وزارت فرهنگ و ارشاد اسلامي
* كسب رتبه ی اوّل به عنوان پژوهشكد ی برتر از وزارت علوم تحقيقات و فناوري در سال 1391
  1. اهداف

با توجه به نیازهای که در ادامه به تغییر و تکمیل نرم افزارهای هوش مصنوعی و تکمیل نوع کاربردی آنها و رسیدن به درآمد زایی، پژوهشگاه توسعه فناوری‌های پیشرفته اهدافی به مانند زیر را در دستور کار خود قرار داده است:

* توسعه و گسترش پژوهش در زمینه فناوری‌های پیشرفته
* تجاری سازی دستاوردهای پژوهشی
* عمومی سازی کاربرد فناوری‌های پیشرفته
* زمینه‌سازی مناسب برای ارتقای فعالیت‌های پژوهشی مرتبط در چهارچوب‌های مبانی ارزش انقلاب اسلامی
  1. جمع بندی

ما در این پژوهشکده در دوره کارآموزی وظیفه تحیلیل و بررسی پروژه تشخیص ویدیو را به عهده داشتیم. دوره کارآموزی بنده به دلیل بیماری کرونا، به صورت دورکاری برگزار شد و محیط کار اینجانب درون منزل بود.

1. مجموعه دادگان
   1. مقدمه

در این پروژه با توجه به این که ما دنبال این هستیم که حرکات انسان رو بشناسیم و یاد بگیریم نوع حرکات انسان در ویدیو ها، از 3 نوع دیتاست استفاده می‌کنیم که از نظر نوع داده‌های آموزشی و تستی متفاوت هستند. (مقدار و نوع پخش شدن داده‌ها)

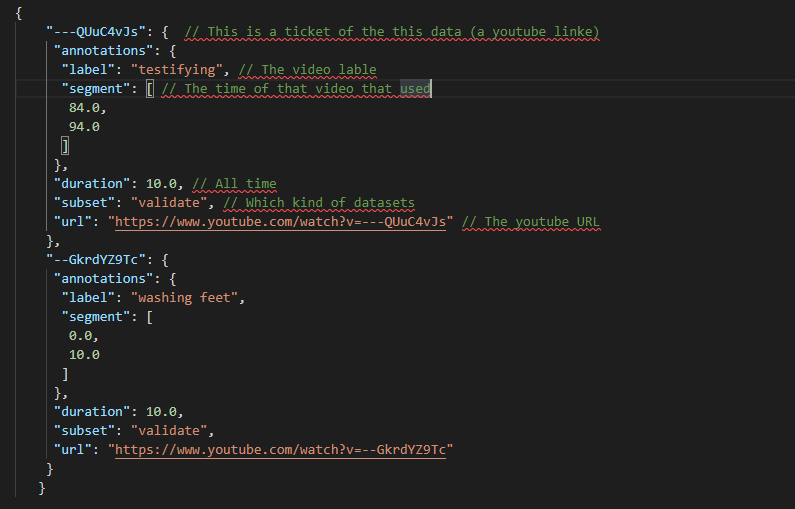
اولین دیتاستی که در مورد آن خواهیم گفت Kinetic-400 ، سپس Kinetic-600 و در نهایت  
 Something-Something v2 رو داریم.

به طور کلی Kineticدارای یک مجموعه از بیش از 650000 ویدیو با کیفیت و اندازه مناسب است که کلاس بندی شده اند با توجه به ورژن آن دیتاست هست و این ویدئوها شامل فعل و انفعالات بین انسان و شیء مانند نواختن سازها ، و همچنین تعاملات انسان با انسان مانند دست دادن و در آغوش گرفتن است. که هر کدام از کلاس ها حداقل به اندازه تعداد کلاس ها ویدیو دارد، که هر ویدیو فقط یک حرکت و حدودا 10 ثانیه زمان دارد. اکثر این ویدیو ها از یوتیوب جمع آوری شده است.

Something-something v2 هم ورژنی آپدیت شده است که مجموعه بزرگی از کلیپ های ویدیویی دارای برچسب که نشان دهنده اقدامات اولیه پیش فرض انسان با اشیا روزمره را داراست که با استفاده از جمعیت زیادی از کارگران جمع آوری شده است. این به مدلهای یادگیری ماشین اجازه می دهد تا درک دقیقی از اقدامات اساسی که در دنیای فیزیکی رخ می دهد توسعه دهند.

* 1. نحوه دسترسی

ویدیوی‎های دیتاست‌ها در دوحالت قابل دسترس خواهند بود اولی به صورت آنلاین و دیگری به صورت آفلاین(دانلود شده). ما با حالت اول کار داریم در این حالت ما یک فایل json یا csv داریم که دارای مشخصاتی به مانند label,youtube\_id,time\_start,time\_end,split که Split به معنی نوع استفاده‌ای برای ویدیو است(یادگیری/تست/اعتبارسنجی). این ویدیوها در فایلهای مانند csv به صورت زیر برای مثال ذخیره می‌شود.



**شکل1-3: نمونه از csv دیتاست‌ها**

با داشتن فایل بالا می‌توان با خواند اطلاعات مورد نیاز از این فایل و دسته بندی قسمت های مختلف برای کلیپ ها (به عنوان مثال به صورت یک دیکشنری) در می‌آوریم و شروع به کار می‌کنیم.

* 1. جزئیات مجموعه دادگان

[Kinetic-400:](https://arxiv.org/pdf/1705.06950.pdf)  در حال حاضر در این ورژن از دیتاست 306245 ویدیو موجود هست که هر کلاس 400 - 1150 تا ویدیو دارد که این ویدیو ها به سه قسمت تقسیم می‌شوند که یک قسمت برای یادگیری[[4]](#footnote-4) که برای هر کلاس بین 250 - 1000 ویدیو دارد ، 50 تا برای اعتبارسنجی[[5]](#footnote-5) و در نهایت 100 ویدیو برای تست[[6]](#footnote-6) برای هر کلاس موجود است.



**شکل 2-3: تعداد کلیپ ها برای هر کلاس در پارتیشن های یادگیری/اعتبارسنجی/تست**

در مقاله ارجاع داده شده درباره این دیتاست درباره موارد مختلفی مانند گیج کننده ترین کلاس ها و سخترین کلاس ها برای تصمیم گیری صحبت شده.(در مورد 400 کلاس که چه موضوعی هست گفته شده)

[Kinetic-600:](https://arxiv.org/pdf/1808.01340v1.pdf) در حال حاضر در این ورژن از دیتاست 495547 ویدیو موجود هست که هر کلاس 600 - 1200 تا ویدیو دارد که این ویدیو ها به سه قسمت تقسیم می‌شوند که یک قسمت برای یادگیری که برای هر کلاس بین 450 - 1000 ویدیو دارد ، 50 تا برای اعتبارسنجی و در نهایت 100 ویدیو برای تست و یک مجموعه تست اضافی حدودا 50تایی برای فعالیت های خاص در این ورژن در نظر میگیریم برای هر کلاس موجود است. نسبت به kinetic-400 از 368 تا از کلاس‌های آن استفاده شده است، برای بقیه 32 کلاس به دلیل اینکه تقاطع زیادی با کلاس های دیگر داشت و شاید برای تشخیص دچار مشکل میشدیم تغییر کرد.



**شکل 3-3: تعداد کلیپ ها برای هر کلاس در پارتیشن های یادگیری/اعتبارسنجی/تست (مقایسه با 400)**

در مقاله ارجاع داده شده در بالا درباره این دیتاست درباره موارد مختلفی مانند گیج کننده ترین کلاس ها و سخترین کلاس ها برای تصمیم گیری صحبت شده.(کلاس های اضافه شده نسبت به 400 در انتهای مقاله ارجاع داده شده صحبت شده)



**شکل 4-3: نمونه‌ای از ویدیو‌های Kinetic**

[Something-Something v2:](https://20bn.com/datasets/something-something) در حال حاضر در این دیتاست 220847 ویدیو موجود هست که شامل 168913 تا ویدیو برای یادگیری و 24777 ویدیو برای اعتبارسنجی و 27175 ویدیو نیز برای تست در نظر گرفته شده با 174 برچسب متفاوت. در این دیتاست بر مبنای اسم برخورد می‌شود برای مثال قراردادن چیزی بر روی چیز دیگر. در ورژن جدید تر این دیتاست TwentyBN 30408 تا شی متمایز از یکدیگر داریم.

* 1. جمع بندی

با توجه به دیتاست‌هایی که در موردشون صحبت کردیم همگی دارای یه ساختار کلی یکسانی هستنند و همگی دارای برچسب‌های مشخص و منحصر به فرد هستند و با تغییر نام دیتاست‌ها تنها نوع کلاس بندی در آن‌ها و تعداد این کلاس‌های مشخصی که هر کدام دارند و تعداد داده‌های برای آموزش و صحت سنجی که مورد استفاده قرار می‌گیرند متفاوت است در حالی که با این داده‌ها کار برای یادگیری و تست برنامه خیلی ساده‌تر از آنچه به نظر می‌آمد شده است.

1. راه اندازی کد
   1. مقدمه

برای هر پروژه‌ای در ابتدا از تعدادی از نرم افزارها و کتابخانه های زبان‌های نوشته شده کدها برای اجرا و راه اندازی استفاده می کنیم. در این پروژه با توجه به اینکه از زبان پایتون برای ساخت ساختار کلی استفاده شده است بایستی یکسری از کتابخانه‌های مربوط به این زبان و یکسری ابزار مربوط به این زبان در راستای کار کردن با ویدیو که قبل از این توسط افراد دیگری تهیه شده است را بر روی سیستم خود نصب کنیم تا به کمک آنها بتوانیم تجزیه و تحلیل ‌های روی ویدیوها برای رسیدن به هدف و تشخیص نوع حرکت در آنها برسیم.

یکی از مهمترین موارد برای شروع کار مشخص کردن سیستم عامل مورد استفاده است، با توجه به تجربه و شناسایی مواردی که مورد انجام خواهند گرفت استفاده از لینوکس توصیه می‌شود.

* 1. نصب پکیج‏ها و پیش‏نیازها

با توجه به اینکه زبان مورد استفاده در این پروژه پایتون هست بهتر است در ابتدا از نصب شدن زبان مورد نظر اطمینان کافی داشته باشیم و حدالمقدور از آخرین نسخه بودن آن مطمئن باشیم.



شکل 1-4: نصب و نسخه پایتون

Pytorch یک کتابخانه‌ای از زبان پایتون هست، که منبع باز برای یادگیری ماشین است که بر مبنای کتابخانه Torch هست که در کاربردهای بینایی ماشین و پردازش زبان استفاده می‌شود که توسط گروه هوش مصنوعی فیسبوک طراحی شده است. برای نصب آخرین ورژن این کتابخانه می توانیم از دستور Pip install torch استفاده کنیم.



شکل 2-4: نصب pytorch

برای چک کردن ورژن نسخه Pytorch میتوانیم از کامپایل فایل pytorch-V.py موجود در پروژه استفاده کرد.

دیگر کتابخانه‌ای که نیاز است numpy هست، NumPy یک کتابخانه برای زبان برنامه نویسی پایتون است که پشتیبانی از آرایه ها و ماتریس های بزرگ و چند بعدی را به همراه مجموعه بزرگی از توابع ریاضی سطح بالا برای کار بر روی این آرایه ها اضافه می کند. برای استفاده از این کتابخانه در ابتدا بایستی آن را به زبان پایتون اضافه کنیم و با استفاده از pip install numpy از طریق ترمینال این کتابخانه را به پایتون اضافه می‌کنیم.

CUDA یک نرم افزار (پلتفرم) محاسباتی موازی و مدل برنامه نویسی است که توسط Nvidia برای محاسبات عمومی در GPU ها توسعه یافته است. CUDA توسعه دهندگان را قادر می سازد تا با استفاده از قدرت پردازنده های گرافیکی در بخش موازی سازی محاسبات ، برنامه های کاربردی با سرعت بالا را افزایش دهند.

برای نصب آن بایستی فایل نصبی آن از آدرس [cuda-download](https://developer.nvidia.com/cuda-downloads) دریافت کرده و نصب کنیم (البته نیاز به نصب visual studio نیز هست).



شکل 3-4: نسخه cuda

برای اینکه سازگاری بهتری بین Cuda و Pytorch اتفاق بیافتد بهتر است از ورژن‌های سازگار با یک دیگر استفاده شود، برای این کار می‌توانیم از سایت [pytorch](https://pytorch.org/get-started/locally/) برای یافتن ورژن سازگار با هم استفاده کنیم در غیر این صورت خطای 1 که درانتهای قسمت دربخش خطاها گفته شده به وجود می‌آید و در این صورت بایستی ورژن torch و یا cuda را تغیر دهیم.

مجموعه کامپایلر GNU یک کامپایلر بهینه ساز است که توسط پروژه GNU تولید شده و از زبانهای برنامه نویسی مختلف ، معماری سخت افزار و سیستم عامل پشتیبانی می کند. بنیاد نرم افزار آزاد GCC را به عنوان نرم افزار رایگان تحت مجوز عمومی GNU توزیع می کند. برای نصب آن از Cygwin استفاده میکنیم تا پکیج های لازم را برای آن بالا بیاورد.



شکل 4-4: نسخه gcc

قبل از نصب mmcv بایستی از نصب numpy و pytorch اطمینان کافی کسب کنیم تا بتوانیم ادامه کار را اجرا کنیم.

Mmcv یک کتابخانه اساسی و کاربردی پایتون برای بینایی ماشین است که از بسیاری از پروژه های تحقیقاتی زیر پشتیبانی می کند:

* MMClassification: جعبه ابزار و معیار طبقه بندی تصویر OpenMMLab.
* MMDetection: جعبه ابزار و معیار تشخیص OpenMMLab.
* MMDetection3D: پلت فرم نسل بعدی OpenMMLab برای تشخیص عمومی اشیاء سه بعدی.
* MMAction2: جعبه ابزار و معیار درک نسل بعدی اقدام OpenMMLab.

این کتابخانه ویژگی های زیر را ارائه می دهد.

* API های جهانی IO
* پردازش تصویر/فیلم
* تجسم تصویر و حاشیه نویسی
* ابزارهای مفید (نوار پیشرفت ، تایمر ،…)
* PyTorch با مکانیسم قلاب
* معماری های مختلف CNN
* اجرای با کیفیت از گزینه های مشترک CUDA

برای نصب این کتابخانه نیاز است که ابتدا جدول زیر را برای سازگار بودن با cuda و pytorch چک کنیم.



که دو حالت برای نصب وجود دارد که یکی cpu محور (که تنها بر انجام تشخیص بوده و در نهایت فقط خروجی دارد) ولی نوع دیگر آن استفاده از Cuda است به صورت گرافیکی روی ویدیوها تغییر ایجاد می‌کند.

Mmcv کلا به دو صورت full که دارای کلیه امکانات این کتابخانه‌ها هست و حالت دیگه به صورت lite است که به صورتی سبک است که فقط قسمت‌های عملیاتی کلی را انجام می‌دهد.

نصب این کتابخانه به صورت زیر است:

pip install mmcv-full -f [https://download.openmmlab.com/mmcv/dist/{cu\_version}/{torch\_version}/index.html](https://download.openmmlab.com/mmcv/dist/%7bcu_version%7d/%7btorch_version%7d/index.html)

FFmpeg یک پروژه نرم افزاری رایگان و منبع باز[[7]](#footnote-7) است که شامل مجموعه ای از کتابخانه ها و برنامه ها برای مدیریت ویدئو ، صدا و سایر فایلها و جریانهای چند رسانه ای است. در مرکز خود ابزار خط فرمان ffmpeg است که برای پردازش فایل های ویدئویی و صوتی طراحی شده است.

دیگر کتابخانه‌ها و ابزار به صورت اختیاری هست که برای انجام بهتر و کامل تر این پروژه استفاده می‌شود که به صورت زیر است:

av

decord >= 0.4.1

imgaug

librosa

lmdb

moviepy

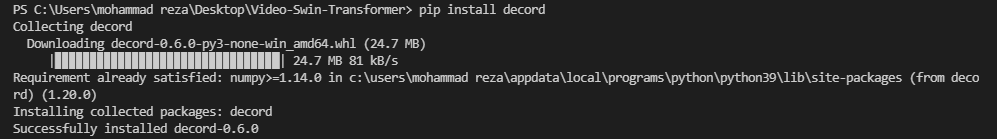
onnx

onnxruntime

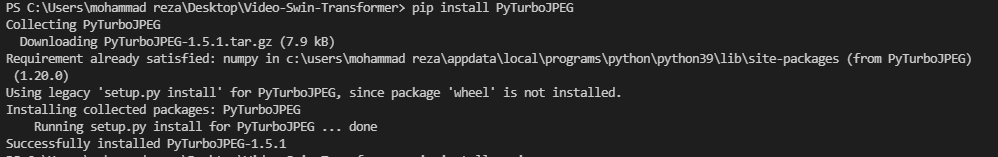
PyTurboJPEG

برای مثال Decord کارآمد و انعطاف پذیر است. این روش های برش ویدئویی مناسب را بر اساس یک بسته بندی در بالای رمزگشاهای ویدئویی با سرعت سخت افزار ارائه می دهد ، به عنوان مثال. کدک های FFMPEG/LibAV و Nvidia. این نرم افزار به گونه ای طراحی شده است که بتواند تجربه دست و پا زدن ویدئویی ناخوشایند را به منظور ارائه تجربیات روان مانند بارگیری تصادفی تصادفی برای یادگیری عمیق ارائه دهد. علاوه بر این ، این برنامه چند پلتفرمی مانند لینوکس ، ویندوز و سیستم عامل مک عمل می کند. با دستگاه خواننده ویدئویی جدید ، دیگر نیازی به رمزگشایی فیلم ها به فریم ندارید ، فقط آموزش مجموعه داده ویدیوی خود را با سرعت آموزش بیشتر شروع کنید.

برای نصب این ابزار از pip install decord استفاده می‌کنیم.



یا برای pyTurboJPEG که یک بسته پایتون هست، از libjpeg-turbo برای رمزگشایی و رمزگذاری تصویر JPEG.



و برای هر کدام دیگر از موارد اختیاری می‌توانیم بنابر لزوم آنها را نصب کنیم.(قسمتی از موارد نصب شده در فایل نصب شده‌ها موجود است)

* 1. نحوه دسترسی

برای اجرا بایستی محیطی در anaconda ایجاد کنیم. Anaconda توزیع زبانهای برنامه نویسی Python و R برای محاسبات علمی است که هدف آن ساده سازی مدیریت و استقرار بسته است. توزیع شامل بسته های علم داده مناسب برای Windows ، Linux و macOS است.

ابتدا ان را بر روی سیستم خود نصب میکنیم و سپس با توجه به اینکه به یک محیط open-mmlab (پروژه های منبع باز برای تحقیقات دانشگاهی و کاربردهای صنعتی. OpenMMLab طیف وسیعی از موضوعات تحقیقاتی بینایی رایانه را پوشش می دهد ، مانند طبقه بندی ، تشخیص ، تقسیم بندی و وضوح فوق العاده.) برای اجرای یک محیط مجازی conda هست را با توجه به نسخه پایتون تولید می‌کنیم.

conda create -n open-mmlab python=3.7 -y

و سپس با فعال کردن این محیط عملیات‌های هم گام سازی بین پکیج ها و کتابخانه‌ها را انجام می‌دهیم.

conda activate open-mmlab

حالا پس از پایان این قسمت از نصب و انجام هماهنگی ها بین کتابخانه‌ها و ابزارها نیاز به نصب mmAction2 برای انجام محاسبات بینایی ماشین لازم است و می توانیم با استفاده از دستور زیر به طور کلی آن را نصب کنیم.

pip install git+https://github.com/open-mmlab/mim.git

mim install mmaction2

یکی از خطاهایی که ممکن است رخ دهد MIM install mmaction2 failed که به دلیل یکسان نبودن یا پیدا نکردن نسخه (No matching distribution found for mmcv-full)

است که کل خطای موجود در فایل mmaction2.txt موجود هست که برای حل این مشکل بایستی با توجه به github این پروژه بایستی نسخه مورد نظر رو به صورت دستی نصب کرد و یکی دیگر از راه حل‌ها تغییر نسخه mmcv است.

بعد از نصب mmaction2 یکسری از فایل‌های در این حالت اضافه می‌شود که در این فایل در قسمت دیتا می‌بایستی همچین درختی از فایل نمایش داده شود

mmaction2

├── mmaction

├── tools

├── configs

├── data

│   ├── kinetics400

│   │   ├── rawframes\_train

│   │   ├── rawframes\_val

│   │   ├── kinetics\_train\_list.txt

│   │   ├── kinetics\_val\_list.txt

│   ├── ucf101

│   │   ├── rawframes\_train

│   │   ├── rawframes\_val

│   │   ├── ucf101\_train\_list.txt

│   │   ├── ucf101\_val\_list.txt

│   ├── ...

و حالا کار نصب پکیج‌ها و ابزارها به حالا کافی است فقط نصب را راستی آزمایی کنیم.

تیکه کد زیر را برای چک کردن این است که آیا نصب کتابخانه‌ها و ابزارها به درستی نصب و هماهمنگ شده اند یا نه.

import torch

from mmaction.apis import init\_recognizer, inference\_recognizer

config\_file = 'configs/recognition/tsn/tsn\_r50\_video\_inference\_1x1x3\_100e\_kinetics400\_rgb.py'

device = 'cuda:0' # or 'cpu'

device = torch.device(device)

model = init\_recognizer(config\_file, device=device)

# inference the demo video

inference\_recognizer(model, 'demo/demo.mp4', 'demo/label\_map\_k400.txt')

با توجه به اینکه از gpu برای کارکرد سیستم استفاده شده یا فقط cpu میتواند نوع پاسخ فرق کند.

* 1. راه‏اندازی سریع

args = [

        '--name', \_\_file\_\_.split('/')[-1].split('.')[0],  # name is filename

        '--print-freq', '1',

        '--data', 'D:/Charades/Charades\_v1\_rgb/Charades\_v1\_rgb',

        '--dataset', 'charadesrgb',

        '--cache-dir','./cache/',

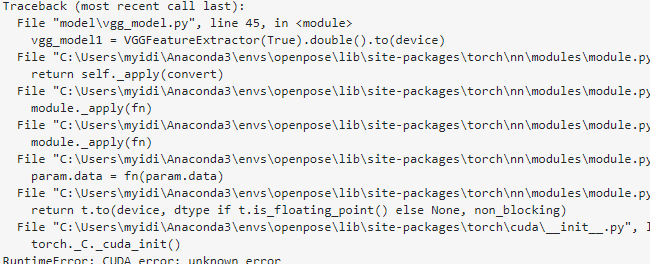
        '--pretrained',

        #'--evaluate',

    ]

**نمونه خطا:**

خطای اولی که ممکن است رخ دهد خطای سازگار نبودن و یا نیافتن cuda توسط pytorch خواهد بود که اروری شبیه به ارور زیر خواهد بود.

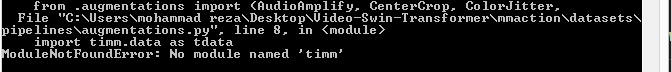


شکل 4-n : خطای اول(مشکل cuda و pytorch)

برای حل کردن این مشکل می‌بایستی با دستور زیر torch مناسب با cuda را نصب کنیم.

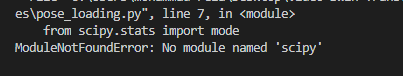
conda install pytorch torchvision cudatoolkit=10.1 -c pytorch

یکی از ارورهای دیگر که با اجرای برنامه راستی آزمایی امکان هست رخ دهد پیدا نکردن کتابخانه [[8]](#footnote-8)Timm هست ) مجموعه ای از مدل های تصویر ، لایه ها ، برنامه های کاربردی ، بهینه سازها ، زمانبندها ، بارگذارنده ها / تقویت کننده ها ، و اسکریپتهای مرجع آموزش / اعتبارسنجی است که هدف آنها جمع آوری طیف گسترده ای از مدلهای SOTA با قابلیت بازتولید نتایج آموزش ImageNet است.(



برای حل این مشکل می‌بایستی pip install timm را در ترمینال اجرا کنیم.

یکی دیگر از خطاها پیدا نکردن scipy هست ) یک نرم افزار منبع باز برای ریاضیات ، علوم و مهندسی است. کتابخانه SciPy به NumPy بستگی دارد ، که دستکاری آرایه N بعدی مناسب و سریع را ارائه می دهد. کتابخانه SciPy برای کار با آرایه های NumPy ساخته شده است و بسیاری از روالهای عددی کاربر پسند و کارآمد مانند روالهای ادغام و بهینه سازی عددی را ارائه می دهد.(



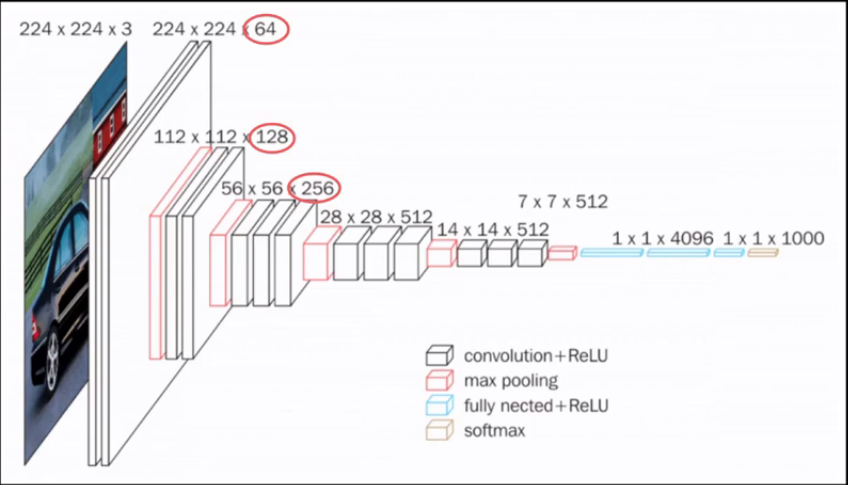
برای حل این مشکل هم در ترمینال از pip install scipy میتوان استفاده کرد.

کتابخانه‌های دیگر به مانند eniops هست که کتابخانه ‌ای بسیار قوی برای کار با تصاویر و تبدیل آنها به شکل‌های متفاوت است که می‌توانیم با استفاده از pip و یا conda آنرا نصب کنیم. در این [ویدیو](https://user-images.githubusercontent.com/6318811/116849688-0ca41c00-aba4-11eb-8ccf-74744f6cbc23.mp4) می‌توانیم کارهای که می‌توان با eniops انجام داد رو دید.

* 1. الگوریتم و روش کلی

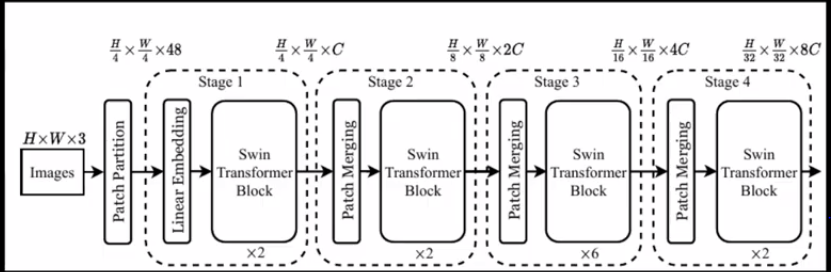
روش‌های شبکه عصبی زیادی به مانند ResNet و یا VGG موجود است برای ترانسفورماتورهای بینایی ولی ترانسفورماتورهای چرخشی به طور کلی این شبکه‌های عصبی را به نحوی تغییر داده اند که به مقصود خود در تحلیل ویدیوها برسند.

بدنه اصلی در بینایی، سگمنت‌ها وتخمین مکانی آنها، تشخیص اشیاء و سنتز تصاویر است، که بایستی به عنوان ویژگی‌ها به شبکه عصبی داده شود تا شبکه عصبی بنابر این اطلاعات تصاویر یا ویدیوها را تشخیص دهد. خوب بدیهی است بدست آوردن این ویژگی‌ها برای طیف عظیمی از تصاویر و ویدیوها کاری سخت و نشدنی است.



با توجه به این تصویر میشه به راحتی فهمید که هر چقدر فریم‌ها رو ریزتر می‌کنیم کار برای تشخیص سخت تر میشه.

اما در ترانسفورماتور چرخشی روال به گونه‌ی دیگری است. در تصویر زیر معماری کلی ترانسفورماتورهای چرخشی رو مشاهده می‌کنیم:



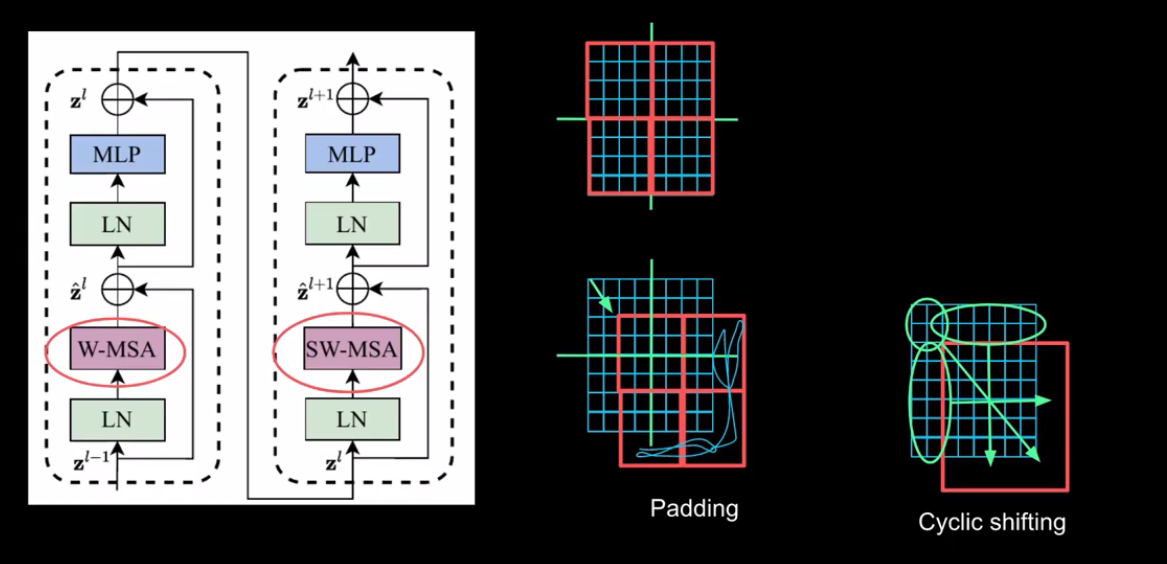
که به طور کلی از چهار قسمت تشکیل شده:

* تقسیم بندی تکه‌ها[[9]](#footnote-9)
* جاسازی خطی[[10]](#footnote-10)
* بلوک ترانسفورماتور چرخشی[[11]](#footnote-11)
* ادغام شدن تکه‌ها[[12]](#footnote-12)

در قسمت اول از این پروسه یعنی تقسیم بندی تکه‌ها، تصویر به تکه‌های کوچکتر شکسته می‌شود به صورتی با یک دیگر هیچ تداخلی نداشته باشند (یعنی به معنای واقعی تصویر را تیکه تیکه می‌کنیم و برای راحتی این تیکه‌ها را هم اندازه فرض می‌کنیم) که برای مثال تصویری با اندازه‌ی H\*W\*3 به چهار تیکه تقسیم میشود و اگر H,W را 8 در نظر گرفته باشیم به تیکه‌های با ابعاد 4\*4\*48 خواهیم رسید.

در قسمت بعدی که جاسازی خطی را داریم به این صورت است که یک شبکه عصبی کوچک برای اینکه تیکه‌ای با ابعاد بگیرد و به تیکه با ابعادی مشخص تبدیل کند( در اینجا 48 بعد را می‌گیرد و تیکه‌ای با ابعاد C را تحویل می‌دهد) در این جا بیشتر مقدار C به اعدادی مثل 96،128،192 مد نظر است.

قبل از اینکه به بلوک ترانسفورماتور چرخشی نگاه کنیم، MSA[[13]](#footnote-13) (توجه چند سر یک ماژول برای مکانیسم های توجه است که چندین بار به طور موازی از طریق یک مکانیسم توجه اجرا می شود. سپس خروجی های توجه مستقل به هم متصل شده و بصورت خطی به بعد مورد انتظار تبدیل می شوند. به طور شهودی ، چند سر توجه اجازه می دهد تا به طور متفاوتی به قسمت های دنباله توجه کنید) که این مورد تنها مشکلی که دارد به پنجره های دیگر توجه نمی‌کند و این خود مشکل ساز هست برای همین ما دو مدل که یکی ساده و دیگری جابه‌جا شده نسبت به حالت فعلی هست در این قسمت استفاده می‌کنیم، پس با این تفسیر خود این بلوک دو لایه (W-MSA , SW-MSA) رو خواهد داشت. در لایه اول به صورت معمولی از MSA استفاده می‌کنیم ولی در لایه دوم تصویر به صورت شیفت خورده(از نظر عمودی و افقی) و برای قسمت‌های خالی هم دو کار میتوان انجام داد که اولی صفر قرار دادن آن پیکسل‌ها و دیگر شیفت دادن چرخه‌ایلسسیش‌تیسیشیسشیسشیش تصویر است.



در قسمت پایانی که ادغام شدن تیکه‌ها هست با توجه به اینکه تعداد تکه نصف می‌شود با ضرب در دو کردن مقدار C این کمبود پوشش داده می‌شود با استفاده از یک شبکه عصبی خطی آنها را باهم ادغام می‌کنیم.

این کار با توجه به نوع داده‌های به کار برده مقدارهای C متفاوتی خواهد داشت و نوع شبکه‌های عصبی آن تغییر خواهد داشت.

در نهایت این کار در 4 استیج پشت سرهم انجام می‌شود تا به سرحد منظور برسد.

* 1. تجارب جدید
     1. زبان YAML

زمانی که ما نیاز به مستندسازی داریم، چالش ما پیدا کردن ابزاری سریع، ساده و قابل استفاده بود. ابزاری که ما برای مستندسازی انتخاب کردیم، زبان برنامه‌نویسی YAML بود.

YAML یک زبان برای طبقه‌بندی داده قابل خواندن برای انسان است که معمولاً برای فایل‌های پیکربندی استفاده می‌شود، اما می‌تواند در کاربردهای دیگری هم مورد استفاده قرارگیرد [5]. YAML بسیاری از برنامه‌های ارتباطات مشابهی را به عنوان XML هدف قرار می‌دهد. این زبان به دو روش نوشته می‌شود:

1- سبک زبان پایتون که با پیش‌آمدی در متن، تودرتویی را پشتیبانی می‌کند.

2- روش فشرده‌تری که مانند JSON برای لیست‌ها از [] و برای نگاشت[[14]](#footnote-14)‌ها از {} استفاده می‌کند که YAML 1.2 را زیر مجموعهی JSON قرار می‌دهد.

انواع داده‌های شخصی‌سازی شده در YAML مجاز هستند، اما YAML به صورت طبیعی اعداد (مانند رشته‌ها، اعداد صحیح و شناور)، لیست‌ها و آرایه‌ها (که رمزنگاری‌شده یا نگاشت شناخته می‌شود) را رمز گذاری می‌کند.

پسوند نام پرونده برای YAML، از سال ۲۰۰۶ تا به اکنون، yaml. است.

* + 1. متدولوژی اسکرام[[15]](#footnote-15)

برای چابک[[16]](#footnote-16) شدن و مدیریت چابک پروژه‌ها روش‌های مختلفی وجود دارد، برخی از روش‌های مدیریت پروژه‌های چابک به نسبت سایر آنها معروف تر هستند و بیشتر مورد توجه و استفاده قرار گرفته‌اند. آشنایی کامل، بی نقص و همزمان با همه روش‌های مدیریت چابک پروژه مانند اسکرام، کانبان [[17]](#footnote-17)و XP امری ضروری نیست اما داشتن آشنایی با آنها در اجرای چابکی به شما این امکان را می‌دهد تا در انتخاب روش‌های متناسب با هر نیاز و شرایط انتخاب‎های متعددی داشته باشید.

اسکرام پر استفاده ترین روش در مدیریت چابک پروژه‌ها است که سه رکن اصلی دارد[7]:

1. شفافیت[[18]](#footnote-18) : ایجاد دید روشن برای افرادی که در به وجود آمدن نتیجه مسئول هستند.

2. بازبینی[[19]](#footnote-19) : بررسی مستمر اینکه در حال انجام چه کاری با توجه به اهداف هستید.

3. تطبیبق[[20]](#footnote-20) : تغییر فرآیند‌های تیم به منظور کاهش خطاها.

تیم‌های اسکرام در بسته‌های کوتاه‌مدت تکراری به نام اسپرینت[[21]](#footnote-21) کار می‌کنند[8]. در عمل، اسکرام از تعدادی از فعالیت‌ها و رسم‌های تکرارشونده در هر اسپرینت پیروی می‌کند. با این کار در واقع اسکرام گاردریل‌[[22]](#footnote-22)های محافظی ایجاد می‌کنند تا افراد به طور موثرتری کار کنند.

* + 1. داکر[[23]](#footnote-23)

به طور کلی داکر،مجموعه ای از پلتفرم به عنوان محصولات خدماتی است که از مجازی سازی در سطح سیستم عامل برای ارائه نرم افزار در بسته هایی به نام ظروف(کانتینر)[[24]](#footnote-24) استفاده می کند.

ظروف از یکدیگر جدا شده و نرم افزارها ، کتابخانه ها و فایلهای پیکربندی خود را به هم پیوند می دهند. آنها می توانند از طریق کانال های مشخص با یکدیگر ارتباط برقرار کنند. با استفاده از این ظروف ایجاد، پیاده سازی و اجرای برنامه‌ها رو بسیار ساده می‌کند. پس به طور کلی با استفاده از داکر سرعت و عملکرد برنامه بهبود قابل ملاحظه‌ای پیدا می‌کند و حجم آن نیز کاهش می‌یابد.

کانتینرها نسبت به ماشین‌های مجازی از منابع کمتری استفاده می‌کنند و همچنین خیلی سریع قابلیت اجرا پیدا می‌کنند که به طور کلی به سه بخش تقسیم می‌شوند:

* سازنده[[25]](#footnote-25) : فناوری مورد استفاده برای ساخت کانتینر
* موتور[[26]](#footnote-26): فناوری مورد استفاده برای راه‌اندازی کانتینر
* تنظیم[[27]](#footnote-27) : فناوری مورد استفاده برای تنظیمات و مدیریت کانتینر

ظرف به یک برنامه نویس کمک می‌کند تا پروژه خود را با تمام بخش‌های آن به صورت یک بسته درآورد. او با وجود این تکنیک، می‌تواند مطمئن باشد که پروژه اش در سیستم‌های دیگر بدون نیاز به ابزارها یا تنظیمات خاص، اجرا می‌شود.

* 1. جمع‏بندی

به طور کلی در این فصل ابتدا به نصب کتابخانه‌ها و ابزارهای مورد نیاز برای اجرا و فعال سازی و تحلیل و تشخیص ویدیو، و به بیان دیگر اجرای الگوریتم مورد نیاز برای اینکه نوع رفتاری رو در ویدیو ها بدست بیاوریم. در ادامه بخش به نحوه دسترسی و استفاده از این کتابخانه‌ها و ابزارها برای پیش برد هدف خودمون را دیدیم و با خطاهای معمولی که در استفاده از این کتابخانه‌ها و ابزارها ممکن است رخ دهد رو حل کردیم و در انتها با موارد جدیدی که در این کارآموزی به تجربه‌هایمان اضافه شد به صورت کلی آشنا شدیم.

1. جمع بندی و پیشنهادات
   1. جمع‏بندی کلی و پیشنهادات

هدف این کارآموزی، اقدام برای یادگیری و پیشبرد و توسعه یکی از ابزارهای هوش مصنوعی که در آینده بسیار نزدیک به دلایل مختلف و زیادی شبیه به استفاده در ربات‌های انسان نما برای برقراری ارتباط با انسان‌ها و دادن پاسخ و عکس العمل مناسب و یا در پیش بینی رفتاری کاربر با توجه به ویدیوهای که کاربران در شبکه‌های اجتماعی با توجه به محبوبیت و استفاده زیاد کاربران از آن می‌تواند مورد کاربرد باشد، نیاز به راه اندازی و بومی سازی این نرم افزار یا به صورت دیگر ابزار برای تشخیص ویدیوها گرفتیم.

پیشنهادی اساسی با توجه به اینکه سیستم عامل ویندوز به صورت کلی محبوب تر از لینوکس هست با تغییراتی رو کدها و داده‌های این ابزار اون را برای سیستم عامل ویندوز نیز راه اندازی و قابل استفاده کنیم.

پیشنهاد دیگر به این گونه که با تغییر الگوریتم یادگیری به صورت خود یادگیری بتوانیم دقت تشخیص رو در این ویدیوها بالا ببریم.

* 1. کارهای آینده

با توجه به پیشنهادات بالا یکی از کارهای موثر در بهبود این ابزار و استفاده با دقت بالاتر می‌تواند تغییر در الگوریتم یادگیری و استفاده از شبکه عصبی در یادگیری استفاده کنیم تا با سرعت بالاتری و با دیتا کمتری به یک دانش کلی از این دیتاها برسیم.

منابع و مراجع

[1] Q. Liu, A. H. Sung, and M. Qiao, "Video steganalysis based on the expanded markov and joint distribution on the transform domains detecting msu stegovideo," in *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*, 2008: IEEE, pp. 671-674.

1. Machine vision [↑](#footnote-ref-1)
2. Image Processing [↑](#footnote-ref-2)
3. Video Swin Transformer [↑](#footnote-ref-3)
4. Training [↑](#footnote-ref-4)
5. Validations [↑](#footnote-ref-5)
6. Test [↑](#footnote-ref-6)
7. Open source [↑](#footnote-ref-7)
8. Py**T**orch **Im**age **M**odels [↑](#footnote-ref-8)
9. Patch Partion [↑](#footnote-ref-9)
10. Linear Embedding [↑](#footnote-ref-10)
11. Swin Transformer Block [↑](#footnote-ref-11)
12. Patch merging [↑](#footnote-ref-12)
13. Multi Head Self attention [↑](#footnote-ref-13)
14. Map [↑](#footnote-ref-14)
15. Scrum [↑](#footnote-ref-15)
16. Agile [↑](#footnote-ref-16)
17. Kanban [↑](#footnote-ref-17)
18. Transparency [↑](#footnote-ref-18)
19. Inspection [↑](#footnote-ref-19)
20. Adaptation [↑](#footnote-ref-20)
21. Sprint [↑](#footnote-ref-21)
22. Guardrails [↑](#footnote-ref-22)
23. Docker [↑](#footnote-ref-23)
24. Container [↑](#footnote-ref-24)
25. Builder [↑](#footnote-ref-25)
26. Engine [↑](#footnote-ref-26)
27. Orchestration [↑](#footnote-ref-27)