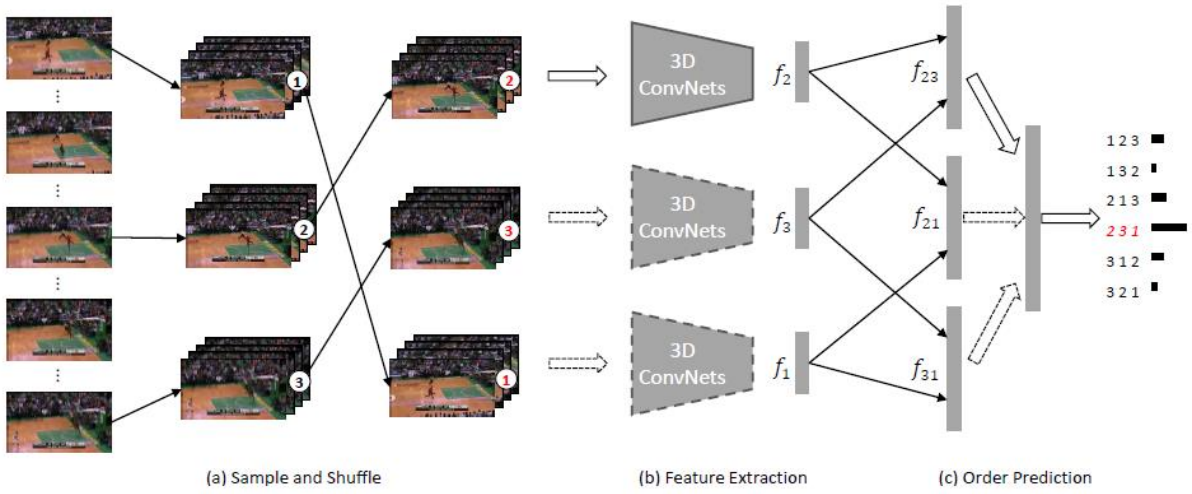
مطابق شکل زیر این روش از سه مرحله تشکیل شده است. در مرحله اول از کلیپ‌ها به طور یکنواخت به طوری که همپوشانی نداشته باشند نمونه برداری می‌کنیم و سپس ترتیب کلیپ‌ها را به هم می‌ریزیم. در مرحله دوم از شبکه‌های کانولوشنی سه بعدی استفاده می‌کنیم تا ویژگی‌های کلیپ‌ها را استخراج کنیم. پارامترهای وزن بین این شبکه‌ها یکسان است. مرحله آخر که مرحله پیشبینی ترتیب می‌باشد را از طریق دسته‌بندی یا classification حل می‌کنیم. ویژگی‌های استخراج شده به صورت جفت بهم متصل می‌شوند و لایه‌های fully connected در بالا قرار میگیرند تا ترتیب واقعی را پیش بینی کنند. پس از آن یک لایه softmax برای محاسبه‌ی خروجی توزیع احتمال روی ترتیب‌های ممکن اعمال می شود. این چارچوب را می توان از انتها به انتها آموزش داد و بعد از آموزش از این شبکه می توان به عنوان یک استخراج کننده ویژگی های ویدئویی یا وزن‌های pre-trained استفاده کرد.



یک کلیپ از فریم‌های متوالی ساخته شده که از از فیلم با اندازه $c \times l \times h \times w$ ساخته شده است ، جایی که c تعداد کانال‌های فریم است ، l تعداد فریم‌ها است ، h و w نشان دهنده ارتفاع و عرض فریم‌ها است. یک کرنل کانولوشنی سه بعدی دارای اندازه $t \times d \times d$ است ، جایی که t عمق زمانی است و هر دو d اندازه مکانی هستند.

یک تاپل مرتب از کلیپها $C = (c_1, c_2, \ldots, c_n)$

ویژگی‌های استخراج شده با $F = (f_1, f_2, \ldots, f_n)$

اندیس ترتیب زمانی را نشان می دهد.

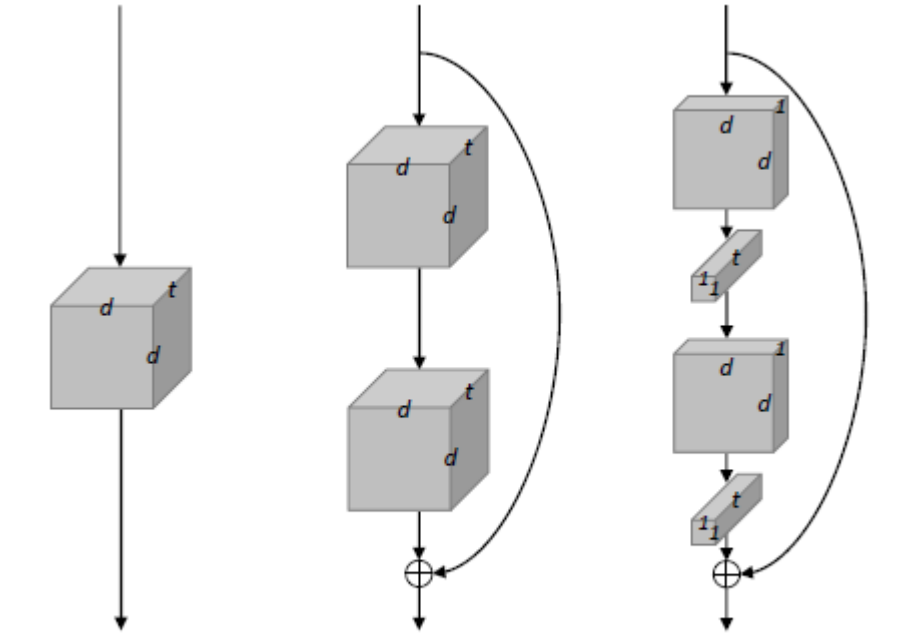
نمونه‌برداری و به هم زدن ترتیب

برای N کلیپ N! ترتیب وجود دارد که کار طبقه‌بندی را مشکل می‌سازد. ما باید تعداد کلیپ‌های انتخابی را محدود کنیم زیرا از آنجا که پیش بینی ترتیب کلیپ فقط یک کار واسط است و ما روی یادگیری CNN های سه بعدی تمرکز داریم ،

این کار باید قابل حل باشد. در غیر این صورت ، اگر حل کل کار خیلی سخت باشد ، تقریباً چیزی نمی‌توان یاد گرفت. تعداد کلیپ ها را بین 2 تا 5 محدود می کنیم که باعث می‌شود تعداد کلاس‌های ترتیب کمتر از 120 باشد. کلیپ‌ها بصورت یکنواخت و با فاصله زمانی m فریم از فیلم نمونه برداری می‌شوند. کلیپ ها باید بدون همپوشانی باشند یا حداکثر در یک فریم همپوشانی داشته باشند. پس از نمونه برداری از کلیپ‌ها ، ترتیب آن‌ها را بهم ریخته تا داده‌های ورودی را تشکیل دهند در حالی که ترتیب واقعی هدف می‌باشد. در طول مرحله آموزش ، تعداد نمونه‌های تولید شده متعلق به هر کلاس ترتیب تقریباً یکسان است.

استخراج ویژگی

پس از تهیه کلیپ‌های بهم ریخته ، از CNN های سه بعدی برای استخراج ویژگی‌های هر کلیپ استفاده می شود. CNNهای سه بعدی مشابه برای همه کلیپ ها در یک تاپل استفاده می شود. ما سه CNN مختلف D3 را به عنوان ویژگی استخراج کننده انتخاب می کنیم که شبکه های C3D، R3D و R(2 + 1)D هستند.



بلوک‌های کانولوشنی C3D: کرنل کانولوشن سه بعدی کلاسیک با اندازه t × d × d که برای تشکیل شبکه C3D روی هم قرار گرفته اند.

بلوک‌های کانولوشنی R3D: کرنل‌های کانولوشن 3 بعدی کلاسیک با اتصال میانبر.

بلوک‌های کانولوشنی R(2 + 1)D: کرنل سه بعدی به یک کرنل فضایی دو بعدی (1 × d × d) و یک کرنل زمانی یک بعدی (t × 1 × 1) تجزیه می شود.

از batch normalization بعد از لایه‌های کانولوشنی نیز استفاده شده‌است.

پیشبینی ترتیب

پیشبینی ترتیب به عنوان یک کار طبقه بندی فرموله شده است. ورودی یک قسمت از ویژگی‌های کلیپ است ، و خروجی یک توزیع احتمال در ترتیب‌های مختلف است. ما از یک پرسپترون چند لایه ساده استفاده می کنیم ، و ویژگی های استخراج شده به صورت جفت به هم متصل شده‌اند ، که هم برای پیش بینی ترتیب و هم برای یادگیری استخراج کننده‌های ویژگی‌های اساسی بهتر است. با توجه به ویژگی‌های استخراج شده داریم:

$$h_k = g_\theta(W_1(f_i \| f_j) + b_1)$$
$$a = W_2 \|_{k=1}^{N} h_k + b_2$$
$$p_i = \frac{\exp(a_i)}{\sum_{j=1}^{C} \exp(a_j))}$$

در جایی که || به معنی هم اتصال بردارها است ، g یک تابع غیرخطی است ، W و b پارامترهای تبدیل خطی هستند ، h_k رابطه بین f_i و f_j را به دست می آورد ، a نشانگر logitها است و p_i احتمال متعلق بودن ترتیب به کلاس i است .

فرض کنید یک تاپل شامل 3 کلیپ است ، پس از شافل کردن یا بهم ریختن ترتیب ، $C = (c_2, c_3, c_1)$ و ویژگی‌های استخراج شده متناظر $F = (f_2, f_3, f_1)$ را بدست می‌آوریم. همانطور که شکل ابتدایی (c) نشان می دهد ، ویژگی ها ابتدا به صورت زوجی به صورت (f23، f21، f31) بهم متصل و سپس به صورت یک تاپل از سه بردار تبدیل می‌شوند تا رابطه بین هر کلیپ را به دست آورد. این بردارها دوباره بهم پیوسته و یک لایه کاملاً متصل به همراه softmax برای تولید پیش‌بینی نهایی اعمال می‌شود. کلاسهای هدف جایگشت‌های (3, 2, 1) هستند که یکی از آنها ترتیب واقعی (1, 3, 2) است.

صحت پیش‌بینی با استفاده از cross-entropy به صورت زیر اندازه‌گیری می‌شود :

$$\mathcal{L} = -\sum_{i=1}^{C} y_i log(p_i)$$

3D CNNs	C3D	R3D	R(2+1)D
Accuracy	68.5	68.4	64.2

جدل فوق دقت کانولشون‌های مختلف روی مسئله ترتیب پیشبینی هست که C3D و R3D بهتر از R(2+1)D عمل کرده‌اند.

برای بازیابی کلیپ‌ها و فریم‌ها روی دو دیتاست معروف UCF101 و HMDB51 به استفاده از الگوریتم k nearest neighbor به شرح زیر است. سطر اول جدول اول مربوط به روش‌های مبتنی بر کانولوشن دو بعدی هستند.

Methods	Top1	Top5	Top10	Top20	Top50
Jigsaw [27]	19.7	28.5	33.5	40.0	49.4
OPN [22]	19.9	28.7	34.0	40.6	51.6
Büchler et al. [1]	*25.7*	*36.2*	*42.2*	*49.2*	*59.5*
C3D (random)	16.0	22.5	26.7	31.4	39.3
C3D	12.5	29.0	39.0	50.6	**66.9**
R3D (random)	10.5	17.2	21.5	27.0	36.7
R3D	14.1	**30.3**	**40.0**	**51.1**	66.5
R(2+1)D (random)	10.2	17.3	21.9	27.7	38.5
R(2+1)D	10.7	25.9	35.4	47.3	63.9

Table 2. **Frame and clip retrieval results on UCF101.** The methods in top row are based on 2D CNNs while 3D CNNs in our framework are presented in bottom row.

Methods	Top1	Top5	Top10	Top20	Top50
C3D (random)	7.7	12.5	17.3	24.1	37.8
C3D	7.4	22.6	**34.4**	48.5	**70.1**
R3D (random)	5.5	11.3	16.5	23.8	37.2
R3D	7.6	**22.9**	**34.4**	**48.8**	68.9
R(2+1)D (random)	4.6	11.1	16.3	23.9	39.3
R(2+1)D	5.7	19.5	30.7	45.8	67.0

Table 3. **Clip retrieval results on HMDB51.** The 3D CNNs used here are self-supervised trained on split 1 of UCF101 merely.

همانطور که در جدول دو می بینیم ، CNN های آموزش دیده 3 بعدی که خود نظارتی دارند ، عملکرد بهتری نسبت به CNN های آموزش دیده 3 بعدی که به صورت تصادفی مقداردهی می‌شوند و CNN های دوبعدی آموزش دیده خود نظارتی دارند ، به ویژه هنگامی که k بزرگتر می شود.

-2

برای قسمت پ این سوال انواع وزن انتخاب شد اگر ضریب وزن کلاسیفایر نسبت به مسئله زاویه بسیار کوچک نباشد مسئله پیشبینی زاویه اصلا train نمی‌شود و دقت برای آن حدود 25 درصد می‌باشد که برابر دقت دسته بندی رندم برای 4 چرخش راویه است. اگر ضریب وزن کلاسیفایر بزرگ باشد در یکی دو ایپاک اورفیت می‌شود و دقت روی داده‌های آموزش 99 درصد و روی داده‌های تست ده درصد می‌شود. ده درصد دقت همان دقت دسته‌بندی رندم برای این مسئله است. اما به هیچ وجه چه با کوچک و چه با بزرگ کردن این وزن دقت روی این مسئله بهتر نشد تا جایی که حتی اگر یک ضریب 0 و دیگری 1 باشد هیچ تاثیری در دقت مدل ایجاد نمی‌شود. روش‌های متفاوتی برای حل این مسئله امتحان شد شامل کوچک کردن لرنینگ ریت متناسب با ایپاک. تغییر ضریب وزن‌ها متناسب با هر ایپاک، زیاد کردن داده‌های برچسب دار. اما هیچ کدام به حل مسئه کمک نکرد. با پرینت پیشبینی برای داده‌های تست به نظر رسید که همه‌ی داده‌ها به یک کلاس نگاشته شده‌اند شاید مدل اولین کلاسی راکه دیده حفظ کرده و دیگر قادر به یادگیری کلاسهای دیگر نبوده‌است.