این مقاله در قالب نهایی آن در مجله
ماشین‌بینایی و پردازش‌تصویر چاپ خواهد شد
تاریخ انتشار برخط: آبان‌ماه ۱۳۹۹

http://jmvip.sinaweb.net
www.ismvip.ir

افزایش دقت بازشناسی صحنه‌های طبیعی پویا با استفاده از همبستگی بین نقشه‌های ویژگی در شبکه‌های عصبی پیچشی

صفورا حیدری[۱]، عباس ابراهیمی‌مقدم[۲]، مرتضی خادمی درح[۲] و هادی هادی‌زاده[۳]

چکیده

بازشناسی صحنه‌های پویا یکی از زمینه‌های تحقیقاتی اساسی در حوزه بینایی ماشین بشمار می‌رود. در این مقاله با استفاده از شبکه‌های عصبی پیچشی (CNN)، روشی مؤثر جهت بازشناسی صحنه‌های پویا ارائه می‌شود. در روش پیشنهادی، همبستگی بین نقشه‌های ویژگی حاصل از لایه‌های مختلف یک شبکه عصبی به عنوان بردارهای ویژگی حاوی اطلاعات ویدئو، مورد استفاده قرار گرفته است. در این روش، ابتدا N فریم از ویدئو انتخاب شده و به کمک یک شبکه عصبی پیچشی، نقشه‌های ویژگی مربوط به فریم‌های منتخب، استخراج شده و برای هر فریم، یک ماتریس گرام محاسبه می‌شود که بیانگر ویژگی‌های مکانی فریم‌های ویدئو است. سپس با قطعه‌بندی زمانی فریم‌های منتخب و میانگین‌گیری بر روی ماتریس‌های گرام این فریم‌ها، اطلاعات زمانی نیز لحاظ می‌شود. با انجام عملیات کدینگ ویژگی‌ها و سپس pooling، برای هر ویدئو یک بردار ویژگی به منظور طبقه‌بندی ویدئو حاصل می‌شود. نتایج شبیه‌سازی‌ها بر روی سه مجموعه داده مطرح در این زمینه نشان می‌دهد که روش پیشنهادی از دقت بازشناسی بهتری در مقایسه با سایر روش‌های مطرح در این زمینه تحقیقاتی برخوردار بوده و دقت بازشناسی را تا ۹% برای مجموعه داده Maryland و ۳% برای مجموعه داده YUP++ بهبود بخشیده است.

کلیدواژه‌ها

بازشناسی صحنه‌های پویا، شبکه عصبی پیچشی، همبستگی نقشه‌های ویژگی

۱ مقدمه

انسان بیشترین اطلاعات را از طریق دیدن و مشاهده کسب می‌کند. برای انسان تفسیر دنیای پیرامونش، امری عادی به شمار می‌رود، امّا دسته‌بندی خودکار گونه‌های مختلف صحنه‌های طبیعی پیچیده توسط کامپیوتر، به یک مسأله تحقیقاتی اساسی در حوزه بینایی ماشین[۱] تبدیل شده است. عملیات بازشناسی صحنه‌های پویا[۲] به فرآیندی اطلاق می‌شود که در آن تعلق صحنه ورودی به یکی از دسته‌های موجود (به عنوان مثال ساحل، ترافیک و ...) مورد بررسی قرار گرفته و دسته مربوط به صحنه ورودی تشخیص داده می‌شود. در تعریف ارائه‌شده، صحنه به مکانی اطلاق می‌شود که در آن اتفاق و یا عملی در حال وقوع باشد [۱].

در سال‌های اخیر الگوریتم‌های زیادی به منظور بازشناسی صحنه‌های پویا ارائه شده است که در تمامی آن‌ها، سعی بر این است که از میان انبوه پیکسل‌های فریم‌های یک ویدئو، ویژگی‌هایی استخراج شوند که بتوانند به خوبی اطلاعات معناداری را در مورد آن ویدئو ارائه دهند. در واقع پیکسل‌ها در ساختار اولیه خود به اندازه کافی قابلیت تمایز نداشته و علاوه بر این حجم بسیار بالایی

این مقاله در تیر ماه سال ۹۸ دریافت، در فروردین ماه سال ۹۹ بازنگری و در مهر ماه سال ۹۹ پذیرفته شد.

[۱] دانشجوی دکتری، گروه مهندسی برق، دانشکده مهندسی، دانشگاه فردوسی مشهد، مشهد
رایانامه: safoora.heidari@mail.um.ac.ir
[۲] گروه مهندسی برق، دانشکده مهندسی، دانشگاه فردوسی مشهد، مشهد
رایانامه: {a.ebrahimi , khademi} @um.ac.ir
[۴] دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی قوچان، قوچان
رایانامه: h.hadizadeh@qiet.ac.ir

نویسنده مسئول: عباس ابراهیمی‌مقدم

[1] Machine Vision
[2] Dynamic Scene Recognition

نیز دارند. ویژگی‌های استخراج‌شده باید حاوی اطلاعات مفید، بدون افزونگی[1] و دارای قدرت تعمیم‌پذیری[2] بالا باشند. چالش اصلی در مسأله بازشناسی صحنه‌های پویا، این است که بتوان اطلاعات مکانی و زمانی قابل اعتمادی را از این صحنه‌ها به نحوی استخراج کرد که این اطلاعات، بهترین دقت بازشناسی را در عملیات طبقه‌بندی داشته باشند [۲].

از نظر ساختاری می‌توان روش‌های بازشناسی صحنه‌های پویا را به دو دسته کلی تقسیم‌بندی نمود. دسته اول روش‌هایی هستند که سعی در مدل‌کردن ساختار مکان-زمانی ویدئوهای ورودی به صورت همزمان داشته و در واقع فریم‌های ویدئو را به صورت مجزا مورد بررسی قرار نمی‌دهند و برای هر حجم مکعبی کوچک از ویدئوی ورودی، یک توصیف‌گر مکان-زمانی پیشنهاد می‌دهند. در این دسته، از روش‌های مختلفی از جمله روش‌های آماری[3] و توصیف‌گرهای محلی[4]، روش‌های مبتنی بر مدل[5]، روش‌های مبتنی بر حرکت[6]، روش‌های مبتنی بر تبدیلات و استفاده از فیلترهای مکان-زمانی و همچنین روش‌های یادگیری عمیق[7] به منظور استخراج ویژگی‌های مکان-زمانی استفاده می‌شود. روش‌های آماری و توصیف‌گرهای محلی، در حقیقت تعمیم روش‌های مکانی استاندارد در حوزه تصویر از قبیل الگوی دودویی محلی[8] (LBP) به حوزه آنالیز مکان-زمانی صحنه‌های پویا هستند [۳] تا [۵]. در روش‌های مبتنی بر مدل، هدف تخمین پارامترهای یک سیستم پویای خطی[9] (LDS) با استفاده از نظریه شناسایی سیستم[10] به منظور استخراج مشخصات مکان-زمانی یک صحنه است. این روش‌ها در اصل به منظور تولید صحنه‌های پویا طراحی شده‌اند، اما می‌توان از پارامترهای تخمین‌زده‌شده در این روش‌ها، به منظور طبقه‌بندی نیز استفاده نمود [۶] تا [۸]. در روش‌های مبتنی بر حرکت از خصوصیات آماری حرکت استخراج‌شده بین فریم‌های متوالی یک صحنه پویا به منظور توصیف توزیع مکانی حرکت موجود در صحنه استفاده می‌شود [۹] تا [۱۱]. روش‌های مبتنی بر تبدیلات و استفاده از فیلترهای مکان-زمانی نیز در واقع تعمیم همین روش‌ها از حوزه تصویر به حوزه مکان-زمان به منظور تجزیه و تحلیل صحنه‌های پویا هستند. از جمله این روش‌ها می‌توان به استفاده از فیلترهای گابور [۱۲]، فیلترهای جهت‌دار مکان-زمانی [۱۳] تا [۱۵] و همچنین تبدیل موجک [۱۶] اشاره نمود. در روش‌های یادگیری عمیق از شبکه‌های عصبی پیچشی[11] (CNN) به منظور استخراج ویژگی استفاده می‌شود [۱۷] تا [۱۹].

در مقابل، دسته دوم روش‌هایی هستند که در آن‌ها ساختارهای مکانی و زمانی ویدئوها در دو مسیر مختلف مدل شده و در نهایت با هم ترکیب می‌شوند و به این صورت یک توصیف‌گر برای هر ویدئو ساخته می‌شود [۲۰] و [۲۱]. روش‌های مدل‌سازی مجزا ابتدا در [۲۲] معرفی شدند. در این مقاله از ویژگی‌های GIST [۲۳] به منظور جمع آوری اطلاعات مکانی از کل فریم‌های ویدئو و از ویژگی‌های آشوب به منظور جمع‌آوری اطلاعات زمانی، استفاده شده و در نهایت این ویژگی‌ها با هم ترکیب می‌شوند. همچنین در [۲۱] از یک شبکه عصبی پیچشی از پیش آموزش‌دیده استفاده شده است. در این مقاله هر فریم ویدئو به منظور جمع-آوری اطلاعات مکانی، به عنوان ورودی به شبکه عصبی داده می-شود و پس از استخراج اطلاعات مکانی از فریم‌های ویدئو، در نهایت بردارهای ویژگی حاوی اطلاعات مکانی به منظور استخراج آماره‌های زمانی با هم ترکیب شده و با پشت سر هم قرار گرفتن آمارگان مرتبه اول و دوم این ویژگی‌ها، یک بردار ویژگی یکتا برای هر ویدئو تولید می‌شود.

به طور کلی، تحقیقات اخیر نشان داده‌است که روش‌هایی که در آن‌ها ویژگی‌های مکانی و زمانی به صورت مجزا استخراج شده و در نهایت با هم ترکیب می‌شوند، نسبت به روش‌هایی که در آن‌ها ویژگی‌های مکان-زمانی به صورت همزمان استخراج می‌شوند، از دقت بازشناسی بالاتری برخوردار هستند [۲]. همچنین ثابت شده است که در قشر بینایی[12] انسان هم فرآیند مشابهی به منظور تشخیص صحنه‌های پویا در جریان است، به این صورت که مشخصات ظاهری یک شیء و حرکت مربوط به آن، به صورت جداگانه مدل‌سازی و تشخیص داده می‌شوند [۲۴].

علاوه بر علاقه‌مندی‌های علمی به بازشناسی صحنه‌های پویا، بسیاری از کاربردهای عملی مفید هم در این زمینه ظهور کرده‌اند، که از این بین می‌توان به کاربرد این سیستم‌ها در سامانه‌های امنیتی و مراقبتی نظیر دوربین‌های نظارت بر رویدادهای مکان-زمانی[13] مانند آتش‌سوزی جنگل و یا بهمن، سیستم‌های دستیار راننده، رباتیک و ... اشاره کرد. با وجود اینکه افزایش روز افزون این‌گونه کاربردها، باعث افزایش تحقیقات در زمینه طبقه‌بندی صحنه‌های پویا شده است، اما بهترین روش‌های ارائه‌شده در این زمینه، هنوز از عملکرد انسان فاصله زیادی دارند.

روش پیشنهادی در این مقاله نیز به نوعی در دسته دوم جای می‌گیرد، چرا که در این روش ابتدا اطلاعات مکانی از هر فریم به صورت مجزا استخراج شده و پس از پس از انجام یک سری پردازش‌هایی بر روی ویژگی‌های محلی اولیه، با میانگین‌گیری بر روی ویژگی‌های مربوط به فریم‌های مختلف، اطلاعات زمانی نیز استخراج شده و در نظر گرفته می‌شود. در حقیقت در روش

[1] Redundancy
[2] Generalizability
[3] Statistical
[4] Local Descriptor
[5] Model-based
[6] Motion-based
[7] Deep Learning
[8] Local Binary Patterns
[9] Linear Dynamical System
[10] System Identification
[11] Convolutional Neural Network
[12] Visual Cortex
[13] Spatio-temporal

پیشنهادی، تغییرات اطلاعات مکانی در طول زمان به عنوان اطلاعات زمانی در نظر گرفته شده است.

در این پژوهش سعی بر این است که شبکه‌های عصبی پیچشی [۲۵] که در سال‌های اخیر در حوزه تصاویر ساکن عملکرد قابل قبولی داشته‌اند [۲۶] تا [۲۹]، را به حوزه صحنه‌های پویا تعمیم دهیم، به این صورت که در ساختاری مبتنی بر شبکه‌های عصبی پیچشی، از نقشه‌های ویژگی[1] استخراج‌شده در لایه‌های مختلف شبکه استفاده شده و توصیف‌گرهایی بر اساس همبستگی[2] بین این نقشه‌ها استخراج می‌شود. توصیف‌گرهای ذکرشده حاوی اطلاعات متمایزکننده‌ای در مورد هر ویدئو هستند که می‌توان از آن‌ها در طبقه‌بندی ویدئوهای مختلف بهره جست. در این روش از تمامی فریم‌های ویدئو استفاده نمی‌شود و در واقع با حذف فریم-هایی با کمترین آنتروپی، از پیچیدگی محاسباتی این روش کاسته شده است. علاوه بر این، در قسمت استخراج ویژگی از همبستگی بین نقشه‌های ویژگی مربوط به لایه‌های مختلف شبکه عصبی پیچشی به عنوان بردارهای ویژگی توصیف‌گر ویدئو استفاده شده است. همچنین با قطعه‌بندی زمانی فریم‌های ویدئو و میانگین‌گیری بر روی ویژگی‌های همبستگی استخراج‌شده از این فریم‌ها، اثرات حرکت دوربین حذف شده است.

در ادامه، در بخش ۲ روش پیشنهادی و چگونگی عملکرد آن مورد بررسی قرار گرفته‌است. در بخش ۳ جزئیات پیاده‌سازی روش پیشنهادی و نتایج شبیه‌سازی‌ها آورده شده است. جمع‌بندی در بخش ۴ ارائه شده است.

۲ روش پیشنهادی

در این مقاله، روشی بر پایه شبکه‌های عصبی پیچشی به منظور استخراج توصیف‌گر ویدئو پیشنهاد می‌شود. این روش دارای شش مرحله اصلی انتخاب فریم‌های کلیدی ویدئو، استخراج ویژگی از این فریم‌ها، کاهش ابعاد ویژگی‌های استخراج‌شده، کدینگ ویژگی-ها، عملیات pooling و در نهایت طبقه‌بندی بردارهای ویژگی محاسبه‌شده برای هر ویدئو است. بلوک دیاگرام کلی روش پیشنهادی در شکل ۱ نمایش داده شده است.

در این روش ابتدا تعدادی از فریم‌های ویدئو به عنوان فریم‌های کلیدی به منظور استخراج ویژگی انتخاب شده و سپس با استفاده از نقشه‌های ویژگی حاصل از لایه‌های مختلف یک شبکه عصبی پیچشی، ویژگی‌های همبستگی بین نقشه‌های ویژگی مختلف [۳۱] محاسبه می‌شوند. پس از استخراج این ویژگی‌ها، از روش تجزیه مقادیر تکین[3] (SVD) به عنوان ابزاری برای کاهش ابعاد بردارهای های ویژگی حاصل از مرحله قبل استفاده می‌شود. پس از این مرحله، کدینگ LLC [4] [۳۰] بر روی ویژگی‌های حاصله اعمال شده و در نهایت با انجام عملیات pooling، بردارهای ویژگی توصیف‌گر هر ویدئو محاسبه شده و به منظور طبقه‌بندی به یک طبقه‌بند ماشین بردار پشتیبان[5] (SVM) داده می‌شوند. به منظور بررسی عملکرد روش پیشنهادی، دقت بازشناسی به صورت درصد تعداد ویدئوهایی که در کلاس مربوط به خود به درستی طبقه‌بندی شده‌اند، به تعداد کل ویدئوهای آن کلاس تعریف می‌شود.

۲-۱ انتخاب فریم‌های کلیدی

به منظور افزایش سرعت و دقت بازشناسی، در این روش تعداد N فریم از ویدئو انتخاب شده و در روش پیشنهادی مورد استفاده قرار می‌گیرد. معیار تغییرات کلی[6]، به منظور انتخاب این N فریم مورد استفاده قرار گرفته‌است. در این معیار ابتدا اپراتور لبه‌یاب سوبل[7] بر روی تمامی فریم‌های ویدئو اعمال شده و سپس اندازه گرادیان فریم‌های فیلترشده محاسبه می‌شود. در نهایت بر روی این مقادیر در هر فریم میانگین‌گیری انجام شده و یک عدد به دست می‌آید که نشان‌دهنده تغییرات کلی آن فریم است. این مقدار در واقع معیاری برای پیچیدگی ساختاری هر فریم است که با استفاده از آن، N فریم با بیشترین پیچیدگی ساختاری، به منظور استخراج ویژگی انتخاب خواهند شد.

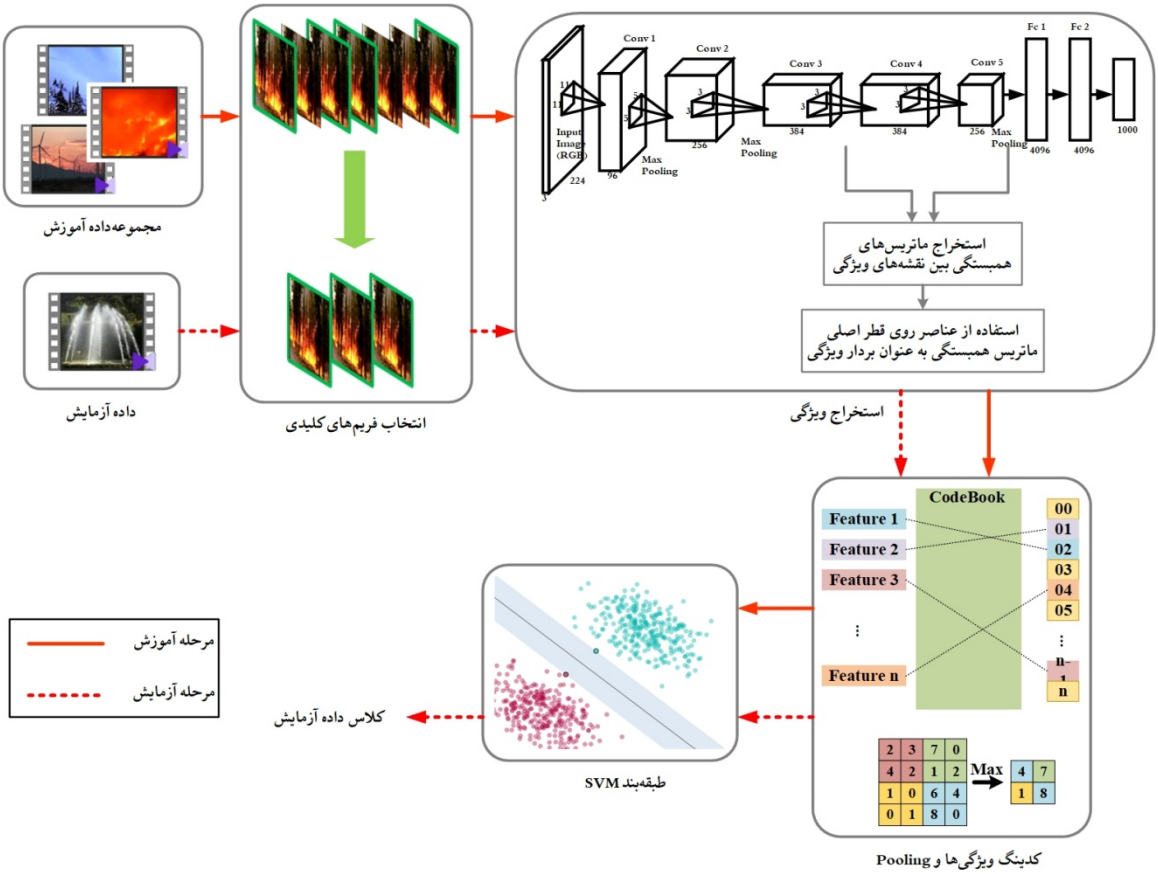
۲-۲ استخراج ویژگی

پس از انتخاب N فریم از ویدئو، نقشه‌های ویژگی مربوط به این فریم‌ها در لایه‌های مختلف شبکه عصبی پیچشی استخراج شده و از این نقشه‌ها به منظور استخراج بردارهای ویژگی توصیف‌گر ویدئو استفاده می‌شود. با کنار هم قراردادن نقشه‌های ویژگی مربوط به فریم tام ویدئو در لایه lام شبکه عصبی پیچشی، ماتریس $\mathbf{A}^{lt} \in \mathbb{R}^{N_l \times M_l}$ ساخته می‌شود که آن را ماتریس ویژگی‌های فریم t-ام و لایه lام می‌نامیم. در این رابطه N_l و M_l به ترتیب نشان‌دهنده تعداد فیلترها در لایه lام و تعداد نقاط مکانی در نقشه‌های ویژگی مربوط به این لایه هستند. در [۳۱] نشان داده شده است که استفاده از همبستگی بین نقشه‌های ویژگی، ابزار قدرتمندی برای توصیف ویدئو در اختیار قرار می‌دهد و بنابراین از آن در تولید بافت‌های پویا استفاده شده است. در نتیجه ماتریس گرام $\mathbf{G}^{lt} \in \mathbb{R}^{N_l \times N_l}$ مربوط به هر یک از ماتریس‌های ویژگی $\mathbf{A}^{lt}$ طبق رابطه ۱ محاسبه شده [۳۱] و سپس طبق رابطه ۲ میانگین‌گیری بر روی ماتریس‌های گرام مربوط به T فریم متوالی از فریم‌های انتخاب شده در مرحله قبل، انجام شده و یک ماتریس گرام حاصل می‌شود.

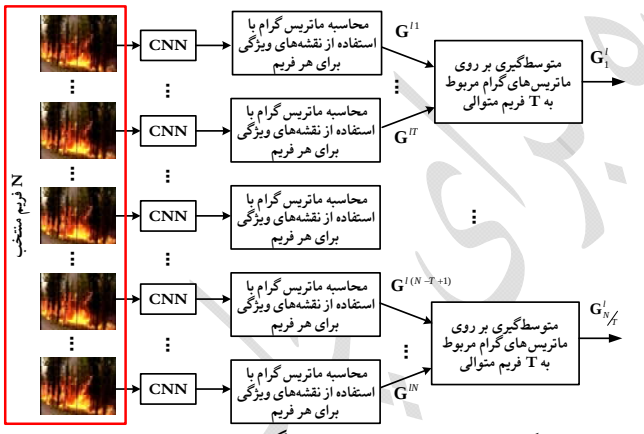
$$\mathbf{G}^{lt}(i,j) = \sum_{k=1}^{M_l} \mathbf{A}^{lt}(i,k) \times \mathbf{A}^{lt}(j,k) \qquad i,j = 1,\dots,N_l \quad (۱)$$

$$\mathbf{G}_r^{l} = \frac{1}{TN_lM_l} \sum_{t=(r-1)T+1}^{rT} \mathbf{G}^{lt} \qquad r = 1,\dots {}^{N}/_{T} \quad (۲)$$

[1] Feature Map
[2] Correlation
[3] Singular Value Decomposition
[4] Locality-constrained Linear Coding
[5] Support Vector Machine
[6] Total Variation
[7] Sobel



شکل ۱- بلوک دیاگرام روش پیشنهادی



شکل ۲- مراحل استخراج ویژگی در روش پیشنهادی

به این ترتیب با دسته‌بندی N فریم منتخب به دسته‌های T تایی و میانگین‌گیری بر روی ماتریس‌های گرام مربوط به فریم‌های هر دسته، تا حدودی می‌توان اثر حرکت دوربین را برطرف کرد.

در نتیجه برای هر ویدئو در هر لایه شبکه عصبی پیچشی، تعداد N/T ماتریس گرام استخراج خواهد شد. ماتریس گرام، ماتریسی متقارن است که عناصر روی قطر اصلی آن، در واقع واریانس نقشه‌های ویژگی را نشان می‌دهند. به منظور کاهش ابعاد ویژگی‌ها و همچنین کاهش پیچیدگی محاسباتی روش پیشنهادی، از عناصر روی قطر اصلی این ماتریس‌ها، به عنوان بردارهای ویژگی توصیف‌گر اطلاعات ویدئو استفاده می‌شود. مراحل استخراج ویژگی در این روش در شکل ۲ نشان داده شده است. پس از استخراج ویژگی‌ها، از روش تجزیه مقادیر تکین به منظور کاهش ابعاد بردارهای ویژگی استفاده می‌شود.

۲-۳ کدینگ ویژگی‌ها

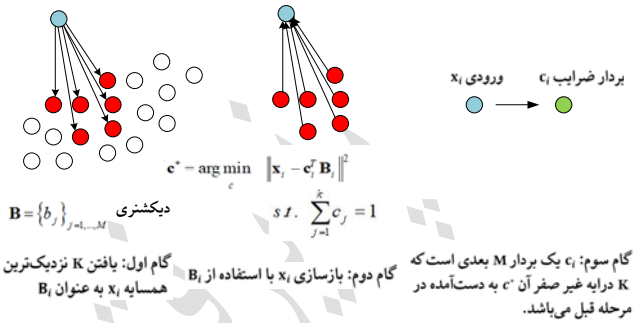
روش‌های کدینگ متفاوتی وجود دارد که می‌توان از آن‌ها به منظور تبدیل ویژگی‌های محلی اولیه به توصیف‌گرهای مؤثرتر بهره جست. انتخاب روش کدینگ مناسب می‌تواند تأثیر زیادی بر روی عملکرد نهایی طبقه‌بندی داشته باشد [۳۲].

در روش کدینگ LLC [۳۰]، ویژگی‌های محلی اولیه با استفاده از نزدیک‌ترین کلمه کدهای موجود در دیکشنری کد می‌شوند.

در این روش، شرط نزدیک‌بودن مکانی کلمات کد جایگزین شرط تُنُک[1] بودن آن‌ها می‌شود. در حقیقت در نظرگرفتن شرط نزدیک‌بودن مکانی کلمات کد از شرط تنک‌بودن مهم‌تر است، چرا که شرط نزدیکی، خود باعث تنک‌شدن کلمات کد خواهد شد. مراحل انجام الگوریتم کدینگ LLC در شکل ۳ نشان داده شده است. همانطور که در این شکل مشخص است، ابتدا با استفاده از الگوریتم K نزدیک‌ترین همسایه[2] (KNN)، K نزدیک‌ترین

[1] Sparse
[2] K Nearest Neighbors

همسایه نمونه $\mathbf{x}_i$ به عنوان عناصر دیکشنری $\mathbf{B}_i$ انتخاب شده و با استفاده از آن‌ها $\mathbf{x}_i$ بازسازی می‌شود. و به این ترتیب بردار ضرایب $\mathbf{c}_i$ ساخته می‌شود. پس از مرحله کدینگ ویژگی‌ها، بردارهای ضرایب به دست آمده برای هر یک از $\mathbf{x}_i$ ها، به عنوان بردارهای ویژگی جدید در نظر گرفته می‌شوند. در این مرحله به منظور دستیابی به یک بردار ویژگی واحد برای هر ویدئو، عملیات pooling بر روی بردارهای ویژگی حاصل از مرحله قبل انجام شده و بردار ویژگی نهایی به منظور طبقه‌بندی وارد طبقه‌بند SVM می‌شود.



شکل ۳- مراحل انجام الگوریتم کدینگ LLC [۳۰]

۳ نتایج شبیه‌سازی‌ها

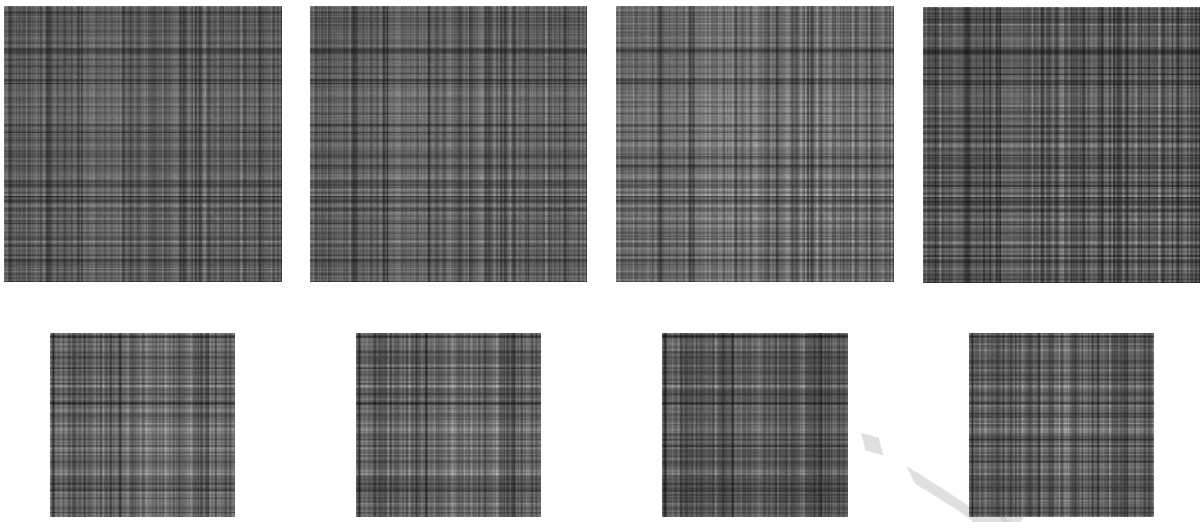
به منظور بررسی دقت بازشناسی در روش پیشنهادی از ۳ مجموعه داده مربوط به بازشناسی صحنه‌های پویا استفاده شده است. مجموعه داده Maryland [۲۲] از ۱۳ کلاس تشکیل شده که در هر کلاس ۱۰ ویدئو از صحنه‌های طبیعی وجود دارد. این ویدئوها از سایت‌های اشتراک ویدئو مانند یوتیوب[1] گرفته شده‌اند، به همین دلیل در این مجموعه داده حرکت دوربین، تغییرات شدید در روشنایی و مقیاس تصاویر و همچنین تغییر در نقطه دید دوربین وجود دارد. متوسط طول ویدئوها در این مجموعه داده، ۶۱۷ فریم و کوتاهترین ویدئو در این مجموعه داده دارای ۳۶ فریم است. مجموعه داده دیگر، YUPenn [۱] است که مشابه مجموعه داده Maryland دارای تغییرات در روشنایی و مقیاس تصاویر و همچنین نقطه دید دوربین است با این تفاوت که در ضبط ویدئو‌های این مجموعه داده، دوربین ساکن بوده است. این مجموعه داده شامل ۱۴ کلاس و ۳۰ ویدئو در هر کلاس است. متوسط طول ویدئوها در این مجموعه داده، ۱۴۵ فریم و کوتاهترین ویدئو در آن دارای ۳۷ فریم است. مجموعه داده سوم، مجموعه داده YUP++ [۱۷] است که با تعمیم کلاس‌های مجموعه داده YUPenn ساخته شده است. این مجموعه داده شامل ۲۰ کلاس و ۶۰ ویدئو در هر کلاس است که نیمی از ویدئوها با دوربین ساکن و نیم دیگر با دوربین متحرک ضبط شده‌اند. متوسط طول ویدئوها در این مجموعه داده، ۱۴۱ فریم و کوتاهترین ویدئو در این مجموعه داده دارای ۳۷ فریم است. لازم به ذکر است که تمامی ویدئوها در این سه مجموعه داده، ویدئوهای کوتاهی هستند که هر کدام از آن‌ها نمایش‌دهنده یک صحنه واحد بوده و تغییر صحنه در آن‌ها اتفاق نمی‌افتد.

در روش پیشنهادی برای پیاده‌سازی مرحله استخراج ویژگی، از شبکه عصبی پیچشی AlexNet [۳۳] از پیش آموزش‌دیده‌شده[2] بر روی مجموعه داده ImageNet، استفاده شده است. این شبکه از ۵ لایه پیچشی[3] و دو لایه تمام‌اتصال[4] تشکیل شده است و یکی از شبکه‌هایی است که در زمینه پردازش تصویر بسیار مورد استفاده قرار می‌گیرد. با توجه به تعداد لایه‌های کمتر این شبکه نسبت به شبکه‌های عمیق‌تری مانند GoogleNet، سرعت پردازش اطلاعات در این شبکه به مراتب بالاتر بوده و همین سرعت بالاتر و عملکرد قابل رقابت با شبکه GoogleNet، دلیلی برای استفاده از این شبکه در روش پیشنهادی بوده است. در لایه‌های پیچشی ابتدایی شبکه، ویژگی‌هایی با ابعاد بیشتر و از نظر مفهومی ساده‌تر استخراج می‌شوند و هر چه در طول این شبکه به جلو حرکت کنیم، ویژگی‌هایی با ابعاد کمتر و پیچیدگی مفهومی بیشتری استخراج می‌شود. به صورت تجربی مشخص شده است که با استفاده از لایه‌های پیچشی سوم و پنجم می‌توان مصالحه‌ای بین دقت بازشناسی و پیچیدگی محاسباتی روش پیشنهادی برقرار کرد. به همین دلیل در شبیه‌سازی‌ها از نقشه‌های ویژگی مربوط به این دو لایه استفاده می‌شود تا علاوه بر پیچیدگی محاسباتی قابل قبول، از ویژگی‌های ساده و پیچیده در کنار هم استفاده شده باشد و به این ترتیب توصیف‌گرهای بهتری برای ویدئو ارائه شود.

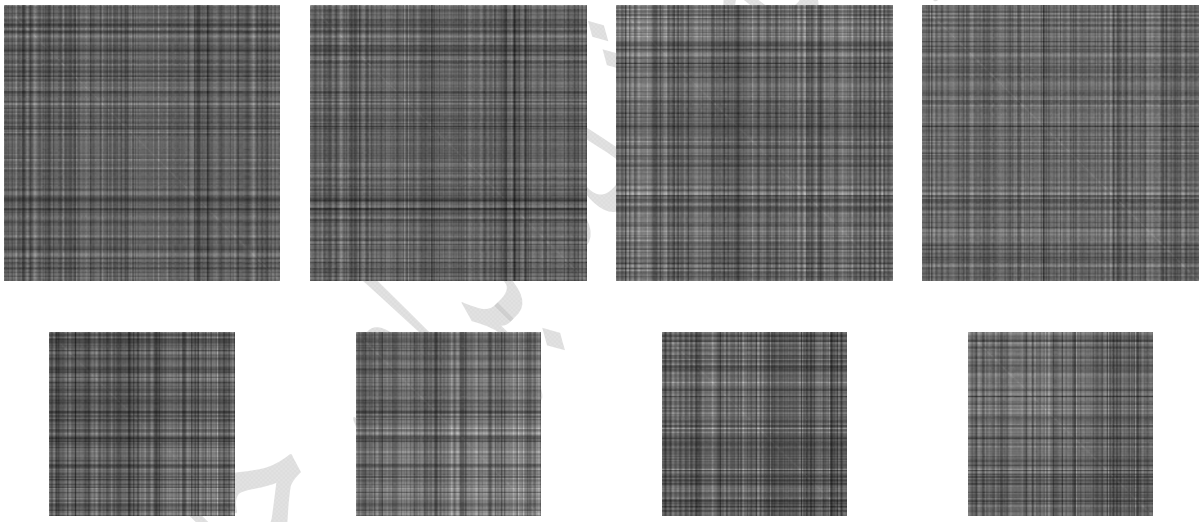
در شبیه‌سازی‌ها $N = 36$ فریم از هر ویدئو بر اساس معیار بیشترین تغییرات کلی انتخاب شده است. به منظور یکسان‌بودن شرایط برای هر سه مجموعه داده، مقدار N با توجه به طول کوتاهترین ویدئوی موجود در این سه مجموعه داده انتخاب شده است. با این انتخاب، با توجه به متوسط تعداد فریم‌ها در هر یک از سه مجموعه داده، تعداد کمی از فریم‌های هر ویدئو مورد استفاده قرار گرفته است، که این امر باعث کاهش پیچیدگی روش پیشنهادی و افزایش سرعت آن می‌شود. این فریم‌ها به شبکه عصبی پیچشی AlexNet داده شده و نقشه‌های ویژگی مربوط به لایه‌های پیچشی سوم و پنجم برای هر فریم استخراج می‌شوند. سپس با استفاده از روابط ۱ و ۲ ماتریس‌های گرام برای هر ویدئو محاسبه شده و از عناصر قطر اصلی این ماتریس‌ها به عنوان بردار-های ویژگی حاوی اطلاعات هر ویدئو استفاده می‌شود. هشت عدد از این ماتریس‌ها صرفاً به عنوان نمونه برای دو ویدئو از دو کلاس متفاوت (بهمن و آتش‌سوزی جنگل) در مجموعه‌داده Maryland به ترتیب در شکل‌های ۴ و ۵ نمایش داده شده است. به منظور سهولت در نمایش، مقادیر این ماتریس‌ها به صورت نرمالیزه‌شده در بازه صفر تا یک به نمایش درآمده است.

[1] YouTube

[2] Pre-trained
[3] Convolutional Layer
[4] Fully-Connected layer



شکل ۴- چند ماتریس گرام نمونه برای یکی از ویدئوهای کلاس بهمن در مجموعه‌داده Maryland، مربوط به لایه پیچشی سوم (سطر اول) و مربوط به لایه پیچشی پنجم (سطر دوم). مقادیر این ماتریس‌ها برای نمایش بهتر در بازه $[۰,۱]$ نرمالیزه شده‌اند.



شکل ۵- چند ماتریس گرام نمونه برای یکی از ویدئوهای کلاس آتش‌سوزی جنگل در مجموعه‌داده Maryland، مربوط به لایه پیچشی سوم (سطر اول) و مربوط به لایه پیچشی پنجم (سطر دوم). مقادیر این ماتریس‌ها برای نمایش بهتر در بازه $[۰,۱]$ نرمالیزه شده‌اند.

در این مرحله با استفاده از الگوریتم SVD، ابعاد ویژگی‌ها کاهش یافته، سپس الگوریتم کدینگ LLC بر روی ویژگی‌های کاهش بعد یافته اعمال می‌شود. پس از این مرحله با انجام عملیات pooling بر روی بردارهای حاصل از فریم‌های ویدئو، یک بردار ویژگی حاوی اطلاعات زمانی برای هر ویدئو استخراج می‌شود. در نهایت با پشت سر هم قرار دادن بردارهای ویژگی محاسبه‌شده از دو لایه پیچشی در شبکه مورد نظر، بردار ویژگی نهایی مربوط به هر ویدئو ساخته شده و با اعمال یک طبقه‌بند SVM با تابع کرنل خطی بر روی این بردارها، عملیات طبقه‌بندی انجام می‌شود. در جدول ۱ دقت بازشناسی روش پیشنهادی با تغییر پارامتر T بررسی شده است. در این جدول نسبت ویدئوهای آموزش و آزمایش به ترتیب ۱۰% و ۹۰% در نظر گرفته شده است، به این معنی که در هر مجموعه داده ۱۰% از ویدئوهای هر کلاس به عنوان ویدئوهای آموزش در نظر گرفته شده است و ۹۰% ویدئوها به منظور آزمایش استفاده شده‌اند.

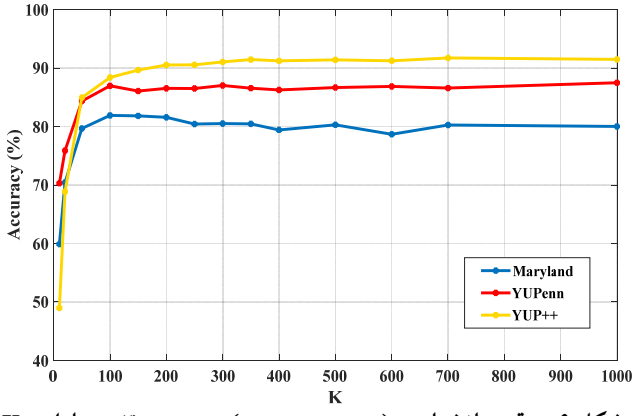
این ویدئوها به صورت تصادفی از میان تمامی ویدئوهای هر کلاس انتخاب می‌شوند. شبیه‌سازی‌ها برای هر مجموعه داده ۱۰۰۰ مرتبه انجام شده و در نهایت بر روی نتایج طبقه‌بندی، میانگین‌گیری انجام شده است. همچنین در این جدول $K = 500$ در نظر گرفته شده است. مشاهده می‌شود که با کاهش T، دقت

بازشناسی در روش پیشنهادی بهبود یافته و در حالتی که $T = 2$ است، این روش بهترین دقت بازشناسی را از خود نشان می‌دهد، به همین دلیل در سایر شبیه‌سازی‌ها $T = 2$ در نظر گرفته می‌شود. در واقع با در نظرگرفتن $T = 2$ یک قطعه‌بندی زمانی بر روی فریم‌های ویدئو انجام شده است که با این قطعه‌بندی زمانی و میانگین‌گیری بر روی مقادیر ویژگی‌های مربوط به هر قطعه، اثر حرکت دوربین تا حدودی برطرف می‌شود. همچنین مشاهده می‌شود که در حالتی که $T = 1$ است، به این معنی که هیچگونه قطعه‌بندی زمانی وجود ندارد، دقت بازشناسی کاهش می‌یابد. علاوه بر این هر چه T افزایش پیدا می‌کند، با توجه به روش انتخاب فریم‌های کلیدی، فاصله بین فریم‌هایی که در میانگین‌گیری شرکت می‌کنند، بیشتر شده و به تَبَع آن فاصله مکانی ویژگی‌های استخراج‌شده از یکدیگر نیز بیشتر شده و همین امر می‌تواند باعث کاهش دقت بازشناسی شود.

جدول ۱- دقت بازشناسی (بر حسب درصد) با تغییر تعداد فریم‌ها در متوسط‌گیری از ماتریس‌های گرام

T	۱	۲	۳	۴	۶
Maryland	۷۹,۹	۸۲	۷۷,۱۳	۶۷,۲۴	۵۴,۷
YUPenn	۸۴	۸۶,۷۱	۷۶,۰۴	۷۱,۲	۵۵,۶۸
YUP++	۸۸,۴	۹۱,۲۱	۷۹,۹	۷۹,۰۱	۶۰,۷۱

در شکل ۶، نمودار دقت بازشناسی روش پیشنهادی بر حسب تغییر پارامتر K در الگوریتم کدینگ LLC نمایش داده شده است. در این شکل نیز مشابه جدول ۱ نسبت ویدئوهای آموزش و آزمایش به ترتیب ۱۰% و ۹۰% در نظر گرفته شده است. همچنین در این شکل $T = 2$ در نظر گرفته شده است.



شکل ۶- دقت بازشناسی (بر حسب درصد) بر حسب تغییر پارامتر K در عملیات کدینگ LLC

مشاهده می‌شود که با افزایش تعداد کلمات کد در دیکشنری، دقت بازشناسی روش پیشنهادی بهبود می‌یابد، امّا در حالتی که $K = 500$ باشد، ضمن دقت بازشناسی مناسب، پیچیدگی محاسباتی نیز چندان زیاد نخواهد بود. به همین دلیل در سایر شبیه‌سازی‌ها از این مقدار برای K استفاده شده است. همچنین مشاهده می‌شود که در این شکل، دقت‌های بازشناسی برای مجموعه داده YUP++ از دو مجموعه دیگر بهتر است که این امر به دلیل بیشتربودن تعداد ویدئوهای هر کلاس در این مجموعه داده است. با توجه به این نکته که نسبت ویدئوهای آموزش و آزمایش در هر سه مجموعه داده یکسان در نظر گرفته شده است، تعداد ویدئوهای آموزش برای مجموعه داده YUP++ نسبت به دو مجموعه داده دیگر بیشتر است و همین امر باعث بهبود در آموزش طبقه‌بند و نتایج طبقه‌بندی و در نتیجه دقت بازشناسی می‌شود.

در جدول ۲ دقت بازشناسی روش پیشنهادی در صورتی که از شکل‌های مختلف عملیات pooling استفاده شود، برای هر سه مجموعه‌داده در حالتی که $T = 2$، $K = 500$ و نسبت ویدئوهای آموزش و آزمایش به ترتیب ۱۰% و ۹۰% است، بررسی شده است و نشان می‌دهد که در صورت استفاده از max pooling، دقت بازشناسی در روش پیشنهادی نسبت به استفاده از min pooling و mean pooling بیشتر بوده و به همین دلیل در سایر شبیه‌سازی‌های این مقاله از max pooling استفاده شده است.

جدول ۲- دقت بازشناسی با تغییر روش pooling

Pooling	max	min	mean
Maryland	۸۲	۵۸,۵۶	۶۸,۰۸
YUPenn	۸۶,۷۱	۶۸,۸۵	۸۵,۴
YUP++	۹۱,۳۶	۷۸,۳۸	۹۰,۷۳

در جدول ۳ دقت بازشناسی روش پیشنهادی با تغییر نسبت ویدئوهای آموزش و آزمایش در حالتی که $T = 2$ و $K = 500$ است، بررسی شده است. بدیهی است که با افزایش درصد ویدئوهای آموزش نسبت به ویدئوهای آزمایش، دقت روش پیشنهادی افزایش می‌یابد. به منظور نمایش دقت بازشناسی روش پیشنهادی در سایر شبیه‌سازی‌های این مقاله از کمترین درصد ویدئوهای آموزش نسبت به ویدئوهای آزمایش یعنی نسبت ۱۰% به ۹۰% استفاده شده است.

در نهایت به منظور مقایسه دقت بازشناسی روش پیشنهادی نسبت به دیگر روش‌های مطرح در زمینه بازشناسی صحنه‌های پویا برای مجموعه داده‌های مختلف، نتایج شبیه‌سازی‌ها در جداول ۴ تا ۶ به صورت میانگین و انحراف معیار نمایش داده شده است.

به منظور یکسان‌بودن شرایط آزمایش برای مقایسه روش پیشنهادی با سایر روش‌های مطرح، در جدول‌های ۴ و ۵ از روش Leave-One-Out (LOO) برای بررسی دقت بازشناسی استفاده شده است. در این روش، یک ویدئو از هر کلاس به عنوان ویدئوی آزمایش در نظر گرفته شده و از مابقی ویدئوها به منظور آموزش طبقه‌بند استفاده می‌شود. این عمل به تعداد ویدئوهای هر کلاس انجام شده و در نهایت بر روی مقادیر محاسبه‌شده، میانگین‌گیری انجام می‌شود. انحراف معیار برای مجموعه‌داده Maryland بسیار

ناچیز و نزدیک به صفر و برای مجموعه‌داده YUPenn دقیقاً برابر با صفر است، به همین دلیل از گزارش آن‌ها صرف نظر شده است. این مقادیر برای مجموعه‌داده ++YUP در جدول ۶ آورده شده است.

جدول ۳- دقت بازشناسی (بر حسب درصد) با تغییر نسبت ویدئوهای آموزش و آزمایش در هر مجموعه داده

Train/Test	۹۰,۱۰	۸۰,۲۰	۷۰,۳۰	۵۰,۵۰	۳۰,۷۰	۲۰,۸۰	۱۰,۹۰
Maryland	۱۰۰	۹۸,۵۴	۹۶,۸۴	۹۳,۶۳	۸۸,۵	۸۵,۲۲	۸۲
YUPenn	۱۰۰	۱۰۰	۹۹,۹۹	۹۹,۶۳	۹۷,۵۲	۹۴,۷۸	۸۶,۷۱
YUP++	۱۰۰	۱۰۰	۱۰۰	۹۹,۸۸	۹۸,۹۷	۹۷,۴۳	۹۱,۲۳

جدول ۴- مقایسه دقت بازشناسی روش پیشنهادی (بر حسب درصد) با روش‌های مطرح بر روی مجموعه داده Maryland

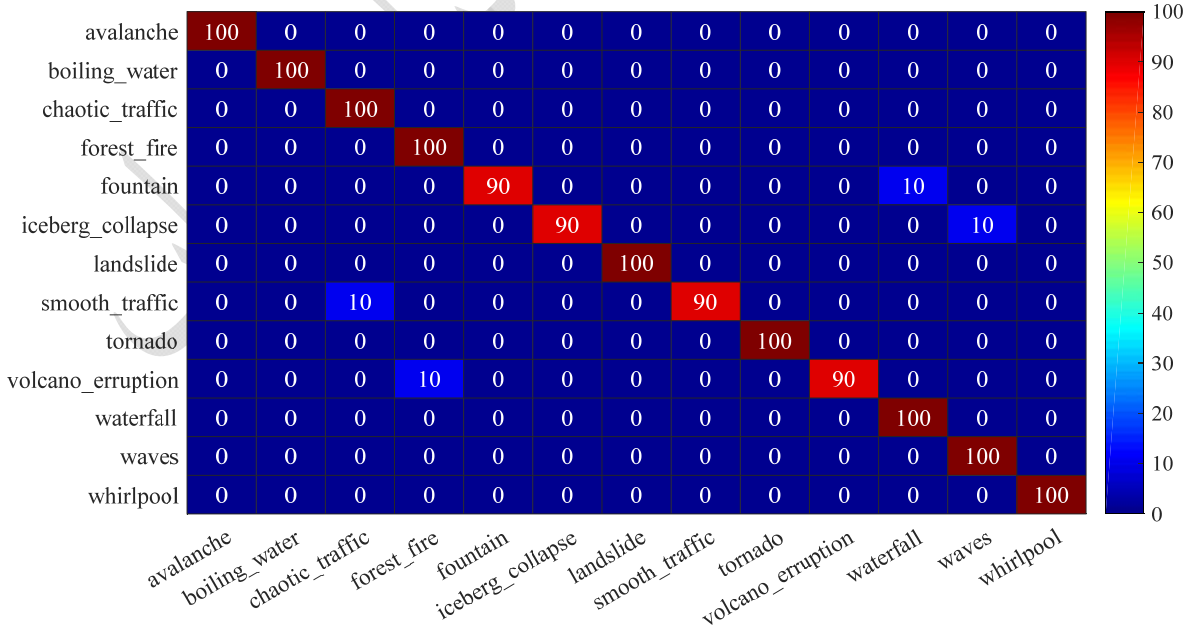
Maryland	*SFA* [15]	*BoSE* [13]	*DPCF* [14]	*C3D* [18]	*st-TCOF* [21]	*RDT* [10]	*DEM* [20]	*Proposed Method*
Avalanche	۹۰	۶۰	۹۰	۹۰	۸۰	۸۰	۹۰	۱۰۰
Boiling Water	۸۰	۷۰	۶۰	۹۰	۹۰	۱۰۰	۹۰	۱۰۰
Chaotic Traffic	۶۰	۹۰	۱۰۰	۹۰	۱۰۰	۱۰۰	۹۰	۱۰۰
Forest Fire	۸۰	۹۰	۹۰	۸۰	۸۰	۱۰۰	۱۰۰	۱۰۰
Fountain	۵۰	۷۰	۸۰	۶۰	۹۰	۷۰	۱۰۰	۹۰
Iceberg Collapse	۷۰	۶۰	۵۰	۶۰	۹۰	۸۰	۸۰	۹۰
Land Slide	۸۰	۶۰	۸۰	۷۰	۱۰۰	۶۰	۸۰	۱۰۰
Smooth Traffic	۷۰	۷۰	۷۰	۸۰	۹۰	۷۰	۹۰	۹۰
Tornado	۸۰	۹۰	۸۰	۸۰	۱۰۰	۹۰	۹۰	۱۰۰
Volcano	۶۰	۸۰	۹۰	۹۰	۱۰۰	۷۰	۱۰۰	۹۰
Water Fall	۷۰	۱۰۰	۷۰	۴۰	۹۰	۱۰۰	۱۰۰	۱۰۰
Waves	۱۰۰	۹۰	۱۰۰	۱۰۰	۷۰	۱۰۰	۱۰۰	۱۰۰
Whirlpool	۸۰	۸۰	۸۰	۸۰	۷۰	۸۰	۹۰	۱۰۰
Average	۷۴,۶۱	۷۷,۶۹	۸۰	۷۷,۶۹	۸۸,۴۶	۸۴,۶۲	۹۲,۳	۹۶,۹۲

جدول ۵- مقایسه دقت بازشناسی روش پیشنهادی (بر حسب درصد) با روش‌های مطرح بر روی مجموعه داده YUPenn

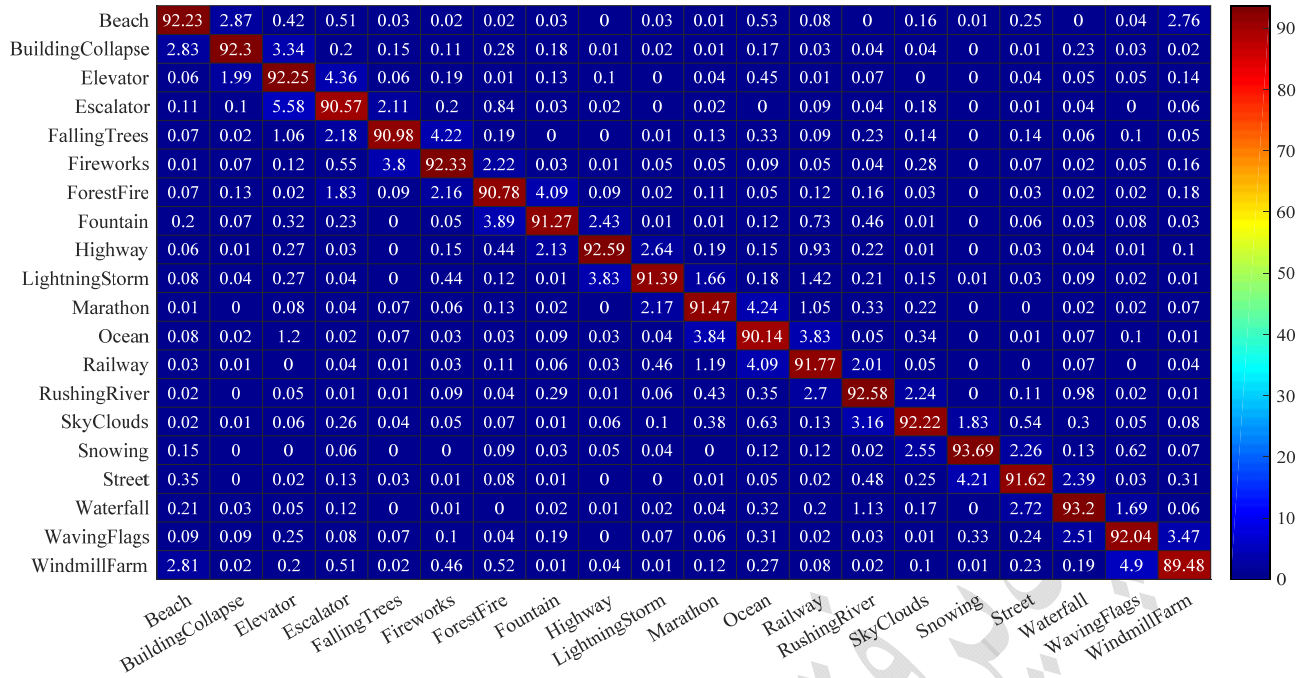
YUPenn	*SFA* [15]	*BoSE* [13]	*DPCF* [14]	*C3D* [18]	*st-TCOF* [21]	*RDT* [10]	*DEM* [20]	*Proposed Method*
Beach	۹۶	۱۰۰	۱۰۰	۹۷	۹۷	۹۶,۶۷	۹۷	۱۰۰
Elevator	۸۶	۹۷	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰
Forest Fire	۹۰	۹۳	۹۷	۱۰۰	۱۰۰	۱۰۰	۹۷	۱۰۰
Fountain	۶۳	۸۷	۹۳	۸۳	۱۰۰	۹۳,۳۳	۹۷	۱۰۰
Highway	۷۰	۱۰۰	۱۰۰	۹۷	۱۰۰	۱۰۰	۱۰۰	۱۰۰
Light. Storm	۸۰	۹۷	۱۰۰	۹۳	۱۰۰	۱۰۰	۹۳	۱۰۰
Ocean	۹۶	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰
Railway	۸۳	۱۰۰	۱۰۰	۹۷	۱۰۰	۹۶,۶۷	۱۰۰	۱۰۰
Rushing River	۸۳	۹۷	۱۰۰	۱۰۰	۹۷	۹۶,۶۷	۱۰۰	۱۰۰
Sky Clouds	۱۰۰	۹۷	۱۰۰	۹۷	۱۰۰	۱۰۰	۹۷	۱۰۰
Snow	۷۳	۹۷	۹۷	۹۳	۹۷	۹۶,۶۷	۱۰۰	۱۰۰
Street	۹۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰
Water Fall	۸۶	۸۳	۹۷	۹۷	۹۷	۹۳,۳۳	۱۰۰	۱۰۰
Windmill Farm	۹۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۱۰۰
Average	۸۴,۷۱	۹۶,۲۸	۹۸,۸۵	۹۶,۷۱	۹۹,۱۴	۹۸,۱۰	۹۸,۶	۱۰۰

جدول ۶- مقایسه دقت بازشناسی روش پیشنهادی (بر حسب درصد) با روش‌های مطرح بر روی مجموعه داده YUP++

YUP++	*SFA*	*BoSE*	*T-ResNet* [17]	*C3D*	*st-TCOF*	*Proposed Method*
Beach	۹۲,۶	۸۳,۳	۷۴,۱	۸۳,۳	۸۵,۵	۰,۱۰±۹۲,۲۳
Build. Collapse	۶۶,۷	۶۶,۷	۹۴,۴	۸۳,۳	۷۸,۹۴	۰,۱۰±۹۲,۳۰
Elevator	۸۵,۲	۹۸,۱	۱۰۰	۹۸,۱	۹۱,۳۳	۰,۱۰±۹۲,۲۵
Escalator	۴۸,۱	۷۴,۱	۹۲,۶	۸۷	۸۴,۳	۰,۰۹±۹۰,۵۷
Falling Trees	۴۲,۶	۷۹,۶	۸۸,۹	۸۸,۹	۷۵,۹۴	۰,۱۲±۹۰,۹۸
Fireworks	۵۱,۹	۸۳,۳	۹۶,۳	۸۱,۵	۵۳,۷۸	۰,۰۹±۹۲,۳۳
Forest Fire	۲۹,۶	۷۷,۸	۱۰۰	۷۹,۶	۷۳,۸۶	۰,۱۱±۹۰,۷۸
Fountain	۱۸,۵	۴۴,۴	۷۵,۹	۳۵,۲	۵۱,۵۷	۰,۱۱±۹۱,۲۷
Highway	۵۵,۶	۵۰	۷۹,۶	۶۴,۸	۶۵,۳۸	۰,۱۰±۹۲,۵۹
Light. Storm	۴۲,۶	۷۹,۶	۹۰,۷	۸۷	۶۸,۳۵	۰,۱۰±۹۱,۳۹
Marathon	۶۶,۷	۸۸,۹	۱۰۰	۱۰۰	۷۲,۰۶	۰,۱۰±۹۱,۴۷
Ocean	۶۴,۸	۷۰,۴	۸۵,۲	۹۶,۳	۷۶,۱	۰,۱۲±۹۰,۱۴
Railway	۲۹,۶	۸۳,۳	۱۰۰	۸۸,۹	۷۲,۹۱	۰,۱۰±۹۱,۷۷
Rushing River	۵۵,۶	۸۱,۵	۸۵,۲	۱۰۰	۷۰,۲۵	۰,۱۱±۹۲,۵۸
Sky Clouds	۸۳,۳	۹۴,۴	۹۶,۳	۹۸,۱	۹۰,۳۷	۰,۱۰±۹۲,۲۲
Snowing	۱۴,۸	۵۷,۴	۵۳,۷	۴۶,۳	۴۹,۷	۰,۱۰±۹۳,۶۹
Street	۷۹,۶	۹۰,۷	۹۸,۱	۹۸,۱	۷۷,۶۴	۰,۰۹±۹۱,۶۲
Water Fall	۷۷,۸	۸۵,۲	۷۵,۹	۹۰,۷	۸۳,۹۸	۰,۰۹±۹۳,۲۰
Waving Flags	۵۳,۷	۸۱,۵	۹۸,۱	۸۸,۹	۶۵,۸۱	۰,۱۰±۹۲,۰۴
Windmill Farm	۷۹,۶	۷۰,۴	۹۴,۴	۸۳,۳	۷۵,۱۳	۰,۱۱±۸۹,۴۸
Average	**۵۶,۹**	**۷۷**	**۸۹**	**۸۴**	**۷۳,۱۵**	**۰,۰۲±۹۱,۷۵**



شکل ۷- ماتریس درهم‌ریختگی متناظر جدول ۴ برای مجموعه‌داده Maryland



شکل ۸- ماتریس درهم‌ریختگی متناظر جدول ۶برای مجموعه‌داده YUP++

در جدول ۶، ۱۰% ویدئوها در هر کلاس به عنوان ویدئوهای آموزش و ۹۰% بقیه به عنوان ویدئوهای آزمایش در نظر گرفته شده و به صورت تصادفی انتخاب می‌شوند. در هر سه این جدول‌ها $T = 2$ و $K = 500$ در نظر گرفته شده است.

ماتریس درهم‌ریختگی[1] برای هر دو مجموعه‌داده Maryland و YUP++ به ترتیب در شکل‌های ۷ و ۸ نمایش داده شده است. چنانچه مشاهده می‌شود در جدول ۵، دقت بازشناسی روش پیشنهادی برای مجموعه‌داده YUPenn در تمامی کلاس‌ها، ۱۰۰% است، به همین دلیل قابل پیش‌بینی است که ماتریس درهم‌ریختگی برای این مجموعه‌داده، یک ماتریس قطری با عناصر روی قطر اصلی برابر با ۱۰۰ خواهد بود. بنابراین از نمایش آن خودداری شده است. همچنین در مورد مجموعه‌داده Maryland مشاهده می‌شود که ویدئوهایی با مشخصات ظاهری مشابه هم، به اشتباه طبقه‌بندی شده‌اند. به عنوان مثال، ویدئوی مربوط به کلاس "آبنما"، به غلط در دسته ویدئوهای مربوط به کلاس "آبشار" طبقه‌بندی شده است. این گونه اشتباه‌ها به دلیل ناچیزبودن تفاوت بین کلاس‌های این مجموعه‌داده رخ می‌دهد.

برتری دقت بازشناسی روش پیشنهادی نسبت به سایر روش‌های مطرح به منظور بازشناسی صحنه‌های پویا برای سه مجموعه‌داده مورد استفاده، در جدول‌های ۴ تا ۶ به وضوح دیده می‌شود. چنانچه مشاهده می‌شود در جدول ۴ علیرغم وجود حرکت دوربین در مجموعه داده Maryland روش پیشنهادی دارای ۹۶/۹۲% دقت بازشناسی است، در حالی که بهترین روش قبلی دقت بازشناسی ۸۸,۴۶% را کسب کرده است. در نتیجه با استفاده از روش پیشنهادی، برای این مجموعه‌داده تا ۹% بهبود در دقت بازشناسی ایجاد شده است.

در خصوص جدول ۵ که دقت بازشناسی روش پیشنهادی را بر روی مجموعه داده YUPenn بررسی کرده است، مشاهده می‌شود که به دلیل سادگی این مجموعه داده نسبت به مجموعه داده Maryland، از حیث عدم وجود حرکت دوربین، تمامی روش‌های مطرح در زمینه بازشناسی صحنه‌های طبیعی پویا از دقت بازشناسی مطلوبی برخوردار هستند. در این حالت نیز روش پیشنهادی نسبت به روش‌های رقیب بهتر عمل کرده است و دقت بازشناسی ۱۰۰% را کسب نموده است.

مجموعه داده YUP++ نسبت به دو مجموعه‌داده دیگر هم از جهت وجود حرکت دوربین و هم از جهت تعداد کلاس‌ها و تعداد ویدئوهای هر کلاس، دارای پیچیدگی بیشتری می‌باشد به همین دلیل مسئله بازشناسی برای این مجموعه داده، چالش‌برانگیزتر از دو مجموعه‌داده قبل است. در جدول ۶ مشاهده می‌شود که با استفاده از روش پیشنهادی و استفاده از ۱۰% ویدئوهای هر کلاس به عنوان ویدئوهای آموزش و ۹۰% آن‌ها به عنوان ویدئوهای آزمایش، دقت بازشناسی بر روی این مجموعه داده به ۹۱/۷۵% رسیده است، در حالی که این مقدار در روش‌های قبل، در شرایط یکسان، در بهترین روش ۸۹% بوده است و به این ترتیب با استفاده از روش پیشنهادی، تا حدود ۳% بهبود در دقت بازشناسی با انحراف معیار ۰,۰۲ کسب شده است.

[1] Confusion Matrix

۴ جمع‌بندی

در این تحقیق روشی مبتنی بر شبکه‌های عصبی پیچشی برای بازشناسی صحنه‌های طبیعی پویا پیشنهاد گردید. در این روش، اطلاعات مکانی و زمانی، با عملکردی مشابه سیستم بینایی انسان، به طور مجزا استخراج شده و در نهایت با یکدیگر ترکیب می‌شوند. در هر ویدئو، پس از انتخاب N فریم با بیشترین پیچیدگی ساختاری، اطلاعات مکانی آن‌ها با استفاده از همبستگی بین نقشه‌های ویژگی در لایه‌های مختلف شبکه عصبی پیچشی استخراج می‌شود. پس از آن، با ترکیب اطلاعات به دست‌آمده از N فریم فوق، اطلاعات زمانی مربوط به آن ویدئو به صورت میانگین‌گیری بر روی T فریم متوالی، استخراج شده که پس از انجام عملیات کدینگ و همچنین pooling، در طبقه‌بندی مورد استفاده قرار می‌گیرد. شبیه‌سازی‌ها بر روی سه مجموعه‌داده مطرح در مسأله بازشناسی صحنه‌های پویا، نشان می‌دهد که روش پیشنهادی با وجود سادگی الگوریتم و پیچیدگی محاسباتی پایین و همچنین استفاده از تعداد صرفاً N فریم که در مقایسه با میانگین تعداد فریم‌ها در هر سه مجموعه‌داده، تعداد کمی از فریم‌های هر ویدئو را شامل می‌شود، از دقت بازشناسی بهتری نسبت به روش‌های دیگر برخوردار است.

همانطور که ملاحظه می‌شود روش پیشنهادی برای مجموعه داده‌های Maryland و YUP++ به ترتیب ۹% و ۳% نسبت به بهترین روش قبلی، بهتر عمل کرده است. در مورد مجموعه داده YUPenn به دلیل ساده‌تربودن این مجموعه داده نسبت به دو مجموعه داده دیگر از حیث عدم وجود حرکت دوربین، روش پیشنهادی نیز مانند اکثر روش‌های موجود از دقت بازشناسی قابل قبولی برخوردار است.

مراجع

[1] K. G. Derpanis, M. Lecce, K. Daniilidis, and R. P. Wildes, *"Dynamic Scene Understanding: The Role of Orientation Features in Space and Time in Scene Classification,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1306-1313, 2012.

[2] S. Hong, J. Ryu, W. Im, H. S. Yang, *"Recognizing Dynamic Scenes with Deep Dual Descriptor Based on Key Frames and Key Segments,"* Neurocomputing, vol. 273, pp. 611-621, 2018.

[3] G. Zhao and M. Pietikainen, *"Dynamic Texture Recognition Using Volume Local Binary Patterns,"* in Dynamical Vision, pp. 165–177, Springer, 2007.

[4] G. Zhao and M. Pietikainen, *"Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions,"* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, 2007.

[5] D. Tiwari and V. Tyagi, *"Improved Weber's Law Based Local Binary Pattern for Dynamic Texture Recognition,"* Multimedia Tools and Applications, pp. 1–18, 2016.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, *"Dynamic Textures,"* International Journal of Computer Vision, vol. 51, no. 2, pp. 91–109, 2003.

[7] V. Venkataraman and P. Turaga, *"Shape Distributions of Nonlinear Dynamical Systems for Video-Based Inference,"* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 38, no. 12, 2016.

[8] J. Miao, X. Xu, X. Xing and D. Tao, *"Manifold Regularized Slow Feature Analysis for Dynamic Texture Recognition,"* arXiv:1706.03015, 2017.

[9] V. Andrearczyk and P. F. Whelan, *"Dynamic Texture Classification Using Combined Co-occurrence Matrices of Optical Flow,"* Irish Machine Vision & Image Processing Conference (IMVI), 2015.

[10] M. R. Khokher, A. Bouzerdoum and S. L. Phung, *"A Super Descriptor Tensor Decomposition for Dynamic Scene Recognition,"* IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 4, pp. 1063-1076, 2019.

[11] M. Marszalek, I. Laptev, and C. Schmid. *"Actions in Context,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[12] W. N. Goncalves, B. B. Machado, and O. M. Bruno, *"Spatiotemporal Gabor Filters: A New Method for Dynamic Texture Recognition,"* arXiv:1201.3612, 2012.

[13] C. Feichtenhofer, A. Pinz, and R. Wildes, *"Bags of Spacetime Energies for Dynamic Scene Recognition,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2681–2688, 2014.

[14] C. Feichtenhofer, A. Pinz, and R. Wildes, *"Dynamic Scene Recognition with Complementary Spatiotemporal Features,"* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 38, no. 12, pp. 2389-2401, 2016.

[15] C. Theriault, N. Thome, and M. Cord, *"Dynamic Scene Classification: Learning Motion Descriptors with Slow Features Analysis,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[16] H. Ji, X. Yang, H. Ling, and Y. Xu, *"Wavelet Domain Multifractal Analysis for Static and Dynamic Texture Classification,"* IEEE Transactions on Image Processing, vol. 22, no. 1, pp. 286–299, 2013.

[17] C. Feichtenhofer, A. Pinz, and R. Wildes, *"Temporal Residual Networks for Dynamic Scene Recognition,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *"Learning Spatiotemporal Features with 3D*

Convolutional Networks," IEEE International Conference on Computer Vision (ICCV), 2015.

[19] Y. Wang and S. Hu, *"Exploiting High Level Feature for Dynamic Texture Recognition,"* Neurocomputing, vol. 154, pp. 217-224, 2015.

[20] J. Zheng, X. Cao, B. Zhang, X. Zhen and X. Su, "Deep Ensemble Machine for Video Classification," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 2, pp. 553-565, 2019.

[21] X. Qi, C. G. Li, G. Zhao, X. Hong and M. Pietikinen, *"Dynamic Texture and Scene Classification by Transferring Deep Image Features,"* Neurocomputing, vol. 171, pp. 1230-1241, 2016.

[22] N. Shroff, P. Turaga, and R. Chellappa, *"Moving Vistas: Exploiting Motion for Describing Scenes,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1911-1918, 2010

[23] A. Oliva and A. Torralba, *"Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope,"* International journal of computer vision, vol. 42, no. 3, pp. 145–175, 2001.

[24] M. A. Goodale and A. D. Milner, *"Separate Visual Pathways for Perception and Action,"* Trends in Neurosciences, pp. 20-25, 1992.

[25] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, *"Backpropagation Applied to Handwritten Zip Code Recognition,"* Neural Computation, pp. 541-551, 1989.

[26] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, *"CNN Features off-the-shelf: an Astounding Baseline for Recognition,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 512–519, 2014.

[27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, *"OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,"* International Conference on Learning Representations (ICLR), 2014.

[28] Y. Sun, Y. Chen, X. Wang and X. Tang, *"Deep Learning Face Representation by Joint Identification-Verification,"* Neural Information Processing Systems (NIPS), pp. 1988–1996, 2014.

[29] L. Herranz, S. Jiang and X. Li, *"Scene Recognition with CNNs: Objects, Scales and Dataset Bias,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 571-579, 2016.

[30] J.Wang, J. Yang, K. Yu, F. Lv, T. Huang and Y. Gong, *"Locality-constrained Linear Coding for Image Classification,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[31] M. Tesfaldet, M. A. Brubaker and K. G. Derpanis, *"Two-Stream Convolutional Networks for Dynamic Texture Synthesis,"* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[32] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, *"The Devil is in the Details: An Evaluation of Recent Feature Encoding Methods,"* British Machine Vision Conference, 2011.

[33] A. Krizhevsky, I. Sutskever, G. E. Hinton, *"ImageNet Classification with Deep Convolutional Neural Networks,"* Neural Information Processing Systems (NIPS), 2012.