



دانشگاه تهران

دانشکده ریاضیات آمار و علوم کامپیوتر

پروژه پایانی کارشناسی علوم کامپیوتر

تشخیص پولیپ در تصاویر کولونوسکوپی بر اساس ترکیبی از تکنیک‌های  
استخراج ویژگی عمیق و یادگیری پیچشی

نگارنده

فاطمه حدادی

استاد

دکتر هدیه ساجدی

مرداد ۱۴۰۰

## چکیده

امروزه سرطان روده بزرگ یکی از خطرناک‌ترین انواع سرطان‌های شناخته شده است. تشخیص به موقع پولیپ‌های داخل روده از اهمیت بالایی برخوردار است. روش‌های فعلی تشخیص تصویری پولیپ، زمان‌بر و غیرمطمئن هستند، چرا که افراد طبق دیدگاه‌های خویش آن‌ها را مورد ارزیابی قرار می‌دهند. روش‌های هوش مصنوعی می‌تواند با کاهش خطای انسانی و قابلیت دسترسی گسترده کمک شایانی برای شناسایی پولیپ‌های سرطانی در تصاویر کولونوسکوپی باشد. در این گزارش، ترکیب جدیدی از چندین تکنیک، از جمله چهارچوب یادگیری باقیمانده به عنوان استخراج‌کننده ویژگی، تحلیل مؤلفه‌های اصلی برای کاهش ویژگی، و یادگیری جمعی برای بهینه‌سازی الگوریتم پیش‌بینی‌کننده، در راستای تشخیص پولیپ‌ها بکار رفته است. از مجموعه داده عمومی با دسترسی آزاد Kvasir برای آموزش و ارزیابی مدل پیشنهادی بکار می‌روند. تصاویر حاوی پولیپ، و بدون پولیپ در این مجموعه داده‌ها گنجانده شده‌اند. راهکار یادگیری ماشین نظارتی مبتنی بر یادگیری عمیق و یادگیری جمعی، تشخیص خودکار پولیپ در تصاویر کولونوسکوپی با نرخ خطای کمتر را ممکن می‌سازد. اما دقت روش پیشنهادی نزدیک به ۹۸,۷٪ است که این نتایج مطابق با جدیدترین روش‌های به کار رفته در تحقیقات است.

## واژه‌های کلیدی:

سرطان روده بزرگ (CRC)، پولیپ، یادگیری انتقالی، یادگیری جمعی، کولونوسکوپی

## پیشگفتار

با توسعه روزافزون در حوزه هوش مصنوعی امروزه سیستم‌ها قادر به تشخیص روابط و ویژگی‌هایی هستند که تا پیش از آن تنها انسان قادر به حل آن بوده است. شبکه‌های عصبی این توانایی را دارند که روابط پیچیده‌ای را شناسایی و بررسی کنند. این قابلیت انگیزه‌ای برای استفاده آن در کاربرد های پزشکی شده است. چرا که با وجود گسترش تکنولوژی کمبود نیروی متخصص و باتجربه همچنان از چالش‌های سیستم درمانی کشورها است.

انگیزه این پروژه، توسعه روشی کاربردی برای تشخیص پولیپ‌های سرطانی روده بزرگ با کمترین میزان دخالت نیروی انسانی است. پولیپ‌های سرطانی در روده بزرگ دارای تنوع بافت و رنگ هستند که تشخیص آن گاهی برای فردی متخصص نیز دشوار خواهد بود. این مدل میتواند در کنار پزشک متخصص برای کمک به تشخیص بیماری استفاده شود. همچنین میتوان از ساختار این طراحی در پیاده سازی مدلی مشابه برای تصاویر پزشکی دیگر مانند سرطان دهانه رحم و تومور های مغزی نیز استفاده کرد. از این رو این پروژه در نهایت به بهبود سیستم های تشخیصی پزشکی کمک شایانی خواهد کرد. بنابراین این پروژه برای تحقیقات بعدی در سایر زمینه های پزشکی نیز مفید میتواند باشد.

برای پیاده سازی مدل از آخرین تحقیقات مرتبط در این زمینه استفاده شده است. این مقالات برخی از فراپارامترهای شبکه معرفی نشده بود. در حین پیاده سازی، خروجی های مدل گاهی در هر اجرا بسیار متفاوت می شد. بدین منظور برای پایداری بیشتری مدل تغییراتی نیز ایجاد شد. از جمله این تغییرات داده افزایی در قسمت بردارهای ویژگی است. داده افزایی معمولاً برای تصاویر مورد استفاده قرار میگیرد اما سبب افزایش محاسبات در فرایند یادگیری می شود. در حالی که با افزایش داده های آموزشی در مرحله یادگیری جمعی به روش AdaBoost در سرعت یادگیری تغییری محسوس ایجاد نخواهد شد.

این گزارش دربردارنده اطلاعاتی از پیش زمینه تحقیق، روش های پیاده سازی شده، کدهای مربوط و نتایج حاصل است. مطالعه این گزارش به تمام علاقه مندان در زمینه یادگیری ماشین و پردازش تصویر توصیه می شود.

## صفحه

## فهرست مطالب

<b>فصل اول: مقدمه.....</b>	<b>۱</b>
۱-۱- تحقیقات قبلی.....	۳
۱-۱-۱- آندوسکوپی.....	۳
۱-۱-۲- تکنیک‌های تشخیص پولیپ مبتنی بر هوش مصنوعی در آندوسکوپی.....	۴
<b>فصل دوم: روش‌ها.....</b>	<b>۶</b>
۲-۱- مجموعه داده.....	۶
۲-۲- پیش‌پردازش تصاویر.....	۷
۲-۲-۱- روش فیلتر میانی.....	۷
۲-۲-۲- روش آستانه‌گیری.....	۸
۲-۲-۳- نرمال‌سازی.....	۸
۲-۳- استخراج ویژگی.....	۹
۲-۳-۱- یادگیری انتقالی با ResNet-50.....	۱۰
۲-۴- داده‌افزایی.....	۱۱
۲-۵- طبقه‌بندی جمعی.....	۱۲
۲-۶- معیارهای سنجش عملکرد و ارزیابی.....	۱۵
<b>فصل سوم: پیاده‌سازی.....</b>	<b>۱۷</b>
۳-۱- فراخوانی کتابخانه‌ها و توابع.....	۱۸
۳-۲- خواندن مجموعه داده و تقسیم آن به داده آموزشی و آزمایشی.....	۱۸
۳-۳- پیش‌پردازش تصاویر.....	۲۰
۳-۴- طراحی مدل.....	۲۱
۳-۵- آموزش مدل.....	۲۲
۳-۶- استخراج بردار ویژگی‌ها.....	۲۳
۳-۷- تحلیل مؤلفه‌های اصلی.....	۲۵
۳-۸- داده‌افزایی بردارهای ویژگی آموزشی.....	۲۶
۳-۹- طبقه‌بندی با یادگیری جمعی.....	۲۷
۳-۱۰- سنجش عملکرد روش مدل.....	۲۸
<b>فصل چهارم: نتایج.....</b>	<b>۳۰</b>

۴-۱- عملکرد سیستم تشخیص رایانه‌ای (CAD) ..... ۳۰

۴-۲- قضاوت غلط در سیستم تشخیص ..... ۳۳

فصل پنجم: بحث و بررسی ..... ۳۴

فصل ششم: جمع‌بندی و نتیجه‌گیری و پیشنهادات ..... ۳۵

منابع و مراجع ..... ۳۶

صفحه	فهرست اشکال
۲	شکل ۱-۱ تفاوت مدل شبکه عصبی پیچشی با مدل پرسپترون چندلایه
۳	شکل ۱-۲ معاینه آندوسکوپی
۷	شکل ۲-۱ نمونه تصاویر کولونوسکوپی حاوی پولیپ
۷	شکل ۲-۲ نمونه تصاویر کولونوسکوپی بدون پولیپ
۹	شکل ۲-۳ مثال‌هایی از تصاویری پس از پیش پردازش و نرمال‌سازی
۱۳	شکل ۲-۴ الگوریتم تقویتی AdaBoost
۱۴	شکل ۲-۵ رأی اکثریت وزنی برای هر پیش بینی
۱۴	شکل ۲-۶ متدولوژی کلی روش یادگیری جمعی
۱۸	شکل ۳-۱ فراخوانی کتابخانه ها و توابع
۱۸	شکل ۳-۲ ذخیره آدرس مجموعه داده
۱۹	شکل ۳-۳ ذخیره آدرس جدید تصاویر مجموعه داده
۱۹	شکل ۳-۴ تقسیم تصاویر حاوی پولیپ
۲۰	شکل ۳-۵ پیش‌پردازش تصاویر
۲۱	شکل ۳-۶ طراحی مدل
۲۲	شکل ۳-۷ آموزش مدل در مرحله اول
۲۳	شکل ۳-۸ آموزش مدل در مرحله دوم
۲۴	شکل ۳-۹ توابع استخراج ویژگی
۲۴	شکل ۳-۱۰ استخراج ویژگی
۲۵	شکل ۳-۱۱ استخراج ویژگی داده تست
۲۵	شکل ۳-۱۲ تحلیل مؤلفه‌های اصلی
۲۶	شکل ۳-۱۳ داده‌افزایی
۲۷	شکل ۳-۱۴ یادگیری جمعی با AdaBoost
۲۸	شکل ۳-۱۵ سنجش عملکرد مدل
۲۹	شکل ۳-۱۶ سنجشگرهای مدل
۲۹	شکل ۳-۱۷ منحنی ROC
۳۲	شکل ۴-۱ مقایسه منحنی ROC برای طبقه بندی پولیپ
۳۳	شکل ۴-۲ تصاویر با دسته بندی غلط

صفحه

فهرست جداول

جدول ۱-۲ اطلاعات مجموعه داده Kvasir.....	۶
جدول ۲-۲ مقایسه شبکه‌های از پیش آموزش دهی شده مختلف.....	۱۱
جدول ۲-۳ مقایسه طبقه‌بندهای مختلف.....	۱۲
جدول ۱-۴ مقایسه عملکرد بین تحقیقات قبلی و روش پیاده سازی شده.....	۳۱

## فصل اول

### مقدمه

سرطان روده بزرگ<sup>۱</sup> از عوامل مهم مرگ و میر در جهان از جمله کشورهای پیشرفته شناخته شده است [۱،۲]. مطابق آمار تهیه شده برای سرطان روده بزرگ در سال ۲۰۲۰، نرخ شیوع در بیماران گروه‌های سنی مسن کاهش یافته است، اما برعکس، میزان ابتلا به این سرطان در جوانان افزایش یافته است [۲]. غربالگری به موقع، جهت تشخیص بیماری، میزان ابتلا و مرگ و میر ناشی از آن را خوشبختانه به شدت کاهش می‌دهد. طی دهه گذشته، سیستم‌های تشخیص مبتنی بر رایانه<sup>۲</sup> به طور گسترده برای غربالگری، تشخیص پزشکی، و درمان سرطان‌های مختلف، از جمله سرطان روده بزرگ بکار گرفته شده‌اند. رادیولوژیست‌ها به کمک فناوری CAD، دامنه‌ی معاینه و تشخیص را محدود می‌کنند. بدین ترتیب به پزشکان کمک شده تا پولیپ‌های روده را در مراحل اولیه بیماری از بدن فرد خارج بکنند. این شرایط، کارایی رویکردهای درمانی را افزایش می‌دهد [۳-۵]. هوش مصنوعی، به مثابه تقلیدی از طرز تفکر انسان تلقی می‌شود. هوش مصنوعی قابلیت‌های هوشمند بودن، یادگیری، و خودسازماندهی را به هر سیستمی اعطا می‌کند [۶]. اخیراً، استفاده از یادگیری عمیق<sup>۳</sup> در حوزه تشخیص رایانه‌ای، اثر قابل ملاحظه‌ای روی تفسیر تصاویر پزشکی بجا گذاشته است. اکنون رادیولوژیست‌ها در شرایط بهتری برای تشخیص بیماری قرار گرفته‌اند. آن‌ها به عنوان مرجع ثانویه، در فرایند تشخیص سرطان ایفای نقش می‌کنند.

در حالت کلی، روش‌های ساخت مدل طبقه بندی کننده تصاویر، شامل سه مرحله هستند: پیش پردازش، استخراج ویژگی و طبقه بندی. مراحل پیش پردازش و نرمالسازی برای تمامی تصاویر ضروری است تا ابعاد تصویر مجدداً تنظیم شده و از کاهش کیفیت جلوگیری می‌شود. از طرفی دیگر، هر تصویر دارای ویژگی‌های منحصر به فردی همچون رنگ، تراکم، شکل، و بافت است [۷]. برای انجام عمل طبقه بندی، ویژگی‌های استخراج شده از مجموعه داده، به کمک الگوریتم‌های یادگیری

---

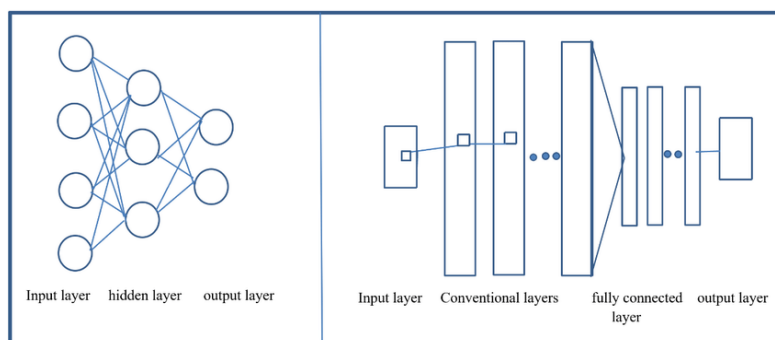
<sup>۱</sup> CRC

<sup>۲</sup> CAD

<sup>۳</sup> Deep Learning



ماشین<sup>۴</sup> از نوع نظارتی، مورد آموزش دهی قرار می‌گیرند. البته استخراج ویژگی، همچنان به پشتیبانی عملیاتی از سوی متخصصان انسانی نیاز دارد [۸]. یادگیری عمیق طی سالیان گذشته، استفاده گسترده‌ای در حوزه‌های پردازش تصویر و ویدئو داشته است. یادگیری عمیق، بر اساس شبکه‌های عصبی پیچشی<sup>۵</sup> نسبت به یادگیری کُل مدل با اعمال فیلترهایی روی قسمت‌های ورودی صورت می‌گیرد. این فرایند شامل دو مرحله‌ی استخراج ویژگی و طبقه‌بندی است. طرز کار این مدل، در تضاد با روش‌های سنتی قدیمی است [۹-۱۰]. در شکل زیر میتوان تفاوت ساختار مدل شبکه‌های عصبی پیچشی با مدل‌های قدیمی تر مانند پرسپترون چندلایه مشاهده کرد.



شکل ۱-۱ تفاوت مدل شبکه عصبی پیچشی با مدل پرسپترون چندلایه

شبکه عصبی پیچشی، یک تکنیک کارآمد و قانع کننده در حوزه یادگیری ماشین، و بخشی از شبکه‌های عصبی عمیق است. علاوه بر این، CNN اخیراً نتایج عالی را از نظر توانایی تشخیص پولپ روده به ارمغان آورده است [۱۰-۲۶]. در سال ۲۰۱۵، در یک آزمایش تشخیص پولپ، CNN عملکرد بهتری از روش‌های غیررایانه‌ای بجا گذاشت [۱۱]. بنابراین استفاده از این روش‌های تشخیص پولپ‌ها بسیار کارآمد خواهد بود. CNN‌ها به دلیل استفاده از فیلترهای گوناگون بر روی بخش‌های تصویر این توانایی را دارند که نمایش‌های مطلوبی از ویژگی‌ها را از انبوهی از تصاویر پراکنده استخراج بکنند. بدین ترتیب وظیفه تشخیص پزشکی بخوبی انجام می‌شود [۱۱].

در این پروژه تلاش شده است که عملکرد سیستم ترکیبی CNN و الگوریتم یادگیری ماشین، جهت تشخیص ناهنجاری‌های جسمانی (وجود پولپ در تصاویر آندوسکوپی) را مورد بررسی قرار بدهیم. در حوزه یادگیری ماشین، یادگیری جمعی در اکثر موارد، برای بهبود کارایی سیستم طبقه‌بند بکار رفته است [۲۷]. بنابراین ترکیب چند الگوریتم یادگیری، دقت دسته‌بندی، قدرت پیش‌بینی روی نمونه‌های جدید را افزایش می‌دهد [۲۸]. برای پروژه فعلی، از مجموعه داده تصاویر آندوسکوپی Kvasir استفاده شده است. کیفیت طبقه‌بندی، بر اساس چندین معیار مورد ارزیابی قرار گرفته است. این شاخص‌ها عبارتند از دقت، نرخ خطا، امتیاز F1، حساسیت و ... .

<sup>۴</sup> Machine Learning

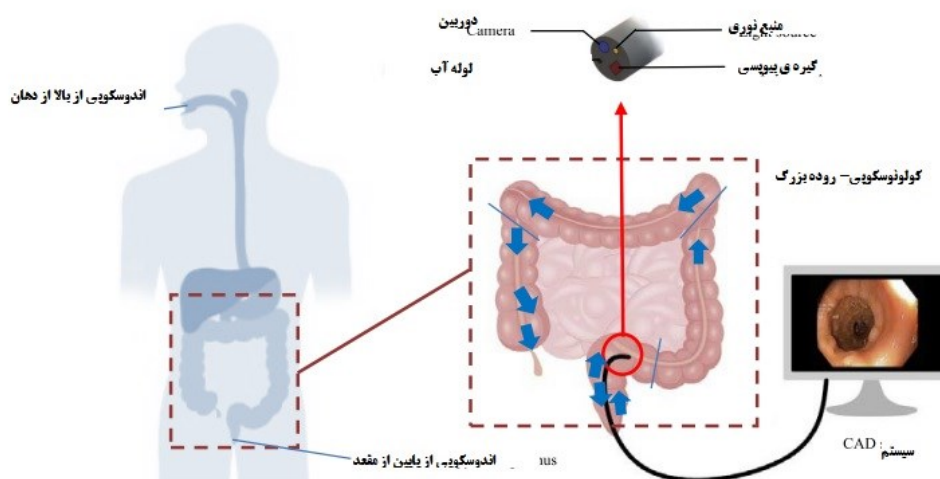
<sup>۵</sup> CNN

برای استخراج ویژگی از ResNet-50 استفاده شده است. در ادامه‌ی کار، تحلیل مؤلفه‌های اصلی (PCA)، افزایش تعداد بردارهای ویژگی با روش KNN و یادگیری جمعی AdaBoost اعمال شده‌اند بخش تحقیقات قبلی، نمای کلی از آندوسکوپی را ارائه می‌دهد. همچنین راجع به نحوه استفاده از هوش مصنوعی برای تحلیل تصاویر آندوسکوپی بحث می‌شود. بخش روش‌ها، جزئیاتی از متدولوژی بکار رفته را ارائه می‌دهد. در بخش‌های نتایج و بحث و بررسی، نتایج ارائه شده معرفی، و راجع به آن‌ها بحث می‌شود. نتیجه‌گیری‌های این پروژه در بخش جمع‌بندی، نتیجه‌گیری و پیشنهادات ارائه می‌شوند. در پایان، در قسمت پیوست کدهای پیاده‌سازی شده توضیح داده می‌شود.

## ۱-۱ تحقیقات قبلی

### ۱-۱-۱ آندوسکوپی

آندوسکوپی، یک عمل پزشکی با ماهیت غیرجراحی است که به پزشک این اجازه را می‌دهد تا اندام یا بافت داخل بدن بیمار را رویت بکند [۳۱]. با توجه به بیماری تشخیصی، هر نوع آندوسکوپی شرایط خاصی را دارا می‌باشد، برای مثال اسفاگوسکوپی، گاستروسکوپی (شکم)، کولونوسکوپی (روده بزرگ)، هیستروسکوپی (رحم)، در آندوسکوپی، یک لوله‌ی باریک و منعطف، به همراه یک سنسور نوری و کپسول متصل به آن، مورد استفاده قرار می‌گیرند. مطابق شکل ۱-۲، در آندوسکوپی از بالا، آندوسکوپ از طریق دهان وارد بدن می‌شود. برای معاینه روده، آندوسکوپ از داخل مقعد وارد می‌شود. علاوه بر این، پزشک توانایی استفاده از گیره‌های بیوپسی (نمونه برداری) را دارد. بدین ترتیب بافت‌هایی با رشد مشکوک به بیماری، برای بررسی‌های تکمیلی نمونه برداری می‌شوند.



شکل ۱-۲ معاینه آندوسکوپی از بالا و پایین و پیشنهاد غربالگری سیستماتیک توسط انجمن آندوسکوپی گوارشی - کولونوسکوپی

اروپا

در حالت عادی، در طول آندوسکوپی، بیش از ۵۵ هزار تصویر از مسیر روده‌ای- گوارشی تهیه می‌شوند. این در حالیست که ناهنجاری‌های مسیر گوارشی، فقط در چند مورد به چشم می‌خورند. در حقیقت، تجزیه و تحلیل تمامی تصاویر ثبت شده توسط سنسور، زمان بر است. کنتراست پایین، پس زمینه‌ی پیچیده، شکل هندسی زخم، رنگبندی هر تصویر، روی نتایج بخش بندی و طبقه بندی تاثیرگذار هستند. این شرایط، فرایند تشخیص بیماری را دشوار نموده‌اند. به منظور پیشگیری از تشخیص غلط، دیدگاه‌های متخصصان مختلف در نظر گرفته می‌شوند. نیاز مبرمی به توسعه‌ی روشی جدید برای تشخیص خودکار ناهنجاری‌های مسیر گوارشی احساس می‌شود. در این وضعیت، حجم کار صورت گرفته برای بررسی و تحلیل داده‌های آندوسکوپی کاهش می‌یابد [۳۵].

## ۲-۱-۱ تکنیک‌های تشخیص پولیپ مبتنی بر هوش مصنوعی در آندوسکوپی

امروزه توجه مضاعفی به روش شبکه عصبی پیچشی در تصاویر پزشکی شده است. در جدیدترین مطالعات، چندین روش مؤثر برای تشخیص زودهنگام و خودکار پولیپ‌های روده بزرگ، معرفی شده‌اند. به عنوان مثال، شین و همکاران [۱۰]، روش شبکه عصبی پیچشی مبتنی بر نواحی تصویر را معرفی نموده‌اند. رویکرد Inception-ResNet به عنوان مکانیزم یادگیری انتقالی<sup>۶</sup>، در سیستم تشخیصی بکار گرفته شده است. روش تشخیصی آن‌ها با تکنیک‌های پس یادگیری ترکیب شده است. در مطالعه‌ی آن‌ها، از چندین مجموعه داده عمومی، و در دسترس از جمله ETIS- CVC-ClinicDB, LaribPolypDB, ClinicVideoDB -CVC، و مجموعه داده ویدئوهای کلونوسکوپی کلینیک مایو- ASO استفاده شده است. گروه دیگری از محققان [۱۸]، از مجموعه داده Kvasir و پیکربندی‌های مختلف، برای انجام آزمایشات استفاده نموده‌اند. حالت‌های مورد استفاده عبارتند از طبقه بندی سراسری ویژگی‌ها، یادگیری عمیق و یادگیری انتقالی. نتایج نشان داده که شبکه عصبی پیچشی از حالت پایه غالب و تصادفی بهتر عمل می‌کند. از طرفی دیگر، عملکرد یادگیری انتقالی از یادگیری عمیق بهتر است زیرا پارامترهای شبکه عصبی پیچشی بهینه سازی نشده‌اند، و تعداد دوره‌های زمانی اجرای آزمایش اندک است. لین و همکاران [۱۹] شبکه عصبی پیچشی مبتنی بر ناحیه (R-CNN) با داده‌افزایی، استخراج ویژگی و تنظیم دقیق، با وزن‌های از قبل آموزش دهی شده از مجموعه داده‌های ImageNet و COCO میکروسافت را، برای تشخیص پولیپ بکار گرفته‌اند. نتایج این تحقیق نشان داده که مدل‌های R-CNN و ResNet-CNN عملکرد مطلوب‌تری را بجا می‌گذارند. علاوه بر این، قدرت محلی سازی آن‌ها روی اکثر تصاویر پولیپ، بهتر از سایر الگوریتم‌های شبکه عصبی پیچشی موجود است.

محمد و همکاران [۲۰]، با ترکیب دو انکودر و ساخت یک شبکه دیکودر، یک معماری Y-NET جدید را معرفی نمودند. VGG-19 از پیش آموزش دهی شده به انکودر اول منتقل می‌شود. تصاویر مجموعه داده پولیپ مایو- ASU به هر دو انکودر ادغام شده اعمال می‌شوند. زوبل و همکاران [۲۱]، معماری R-CNN پوششی و ResNet-101 را برای استخراج

<sup>۶</sup> Transfer Learning

ویژگی‌های تصویر بکار گرفتند. بخاطر محدود بودن تعداد مجموعه داده‌ها، از رویکرد یادگیری انتقالی جهت آموزش یک معماری شبکه پیچیده روی مجموعه داده‌های Bayreuth، CVC-ClinicDB، ETIS-LaribPolypDB استفاده شده است. یک گروه دیگر از پژوهشگران، چهارچوب تشخیص تک مرحله‌ای<sup>۷</sup> با InceptionV3 به عنوان استخراج گر ویژگی را پیشنهاد دادند. در این روش، تشخیص تک مرحله‌ای از شبکه عصبی پیچشی فیدفوروارد<sup>۸</sup>، برای ایجاد کادر مرزی ثابت برای هر شی، روی نگاشت‌های ویژگی مختلف استفاده می‌کند. ارزیابی مدل تشخیص تک مرحله‌ای، با استفاده از مجموعه داده‌های ETIS-LaribPolypDB، CVC-ClinicDB، و CVC-ColonDB صورت گرفته است. وانی و همکاران [۲۳] در تحلیل خود، عملکرد تکنیک‌های مختلف یادگیری عمیق را مقایسه کردند. آن‌ها تصاویر مختلفی از مجموعه داده‌ی CVC-ColonDB، تصاویر بدست آمده از آندوسکوپی کپسول بی سیم<sup>۸</sup> و تصاویر اندوتالس و شلی را مورد استفاده قرار دادند. ندیمی و همکاران [۲۴]، از الگوریتم ZF-Net بهینه سازی شده با گرادیان نزولی تصادفی با گشتاور<sup>۹</sup> استفاده کردند. در این روش، داده افزایی (تکنیک افزایش داده)، پیش پردازش و یادگیری انتقالی در یکدیگر ترکیب می‌شوند. پاتینو بارینتوس و همکاران [۲۵]، مدل یادگیری عمیق مبتنی بر طبقه بندی کودو را پیشنهاد دادند. در روش آن‌ها، VGG-16 به عنوان استخراجگر ویژگی، روی مجموعه داده خصوصی از دانشگاه دوتسو بکار رفته است.

SSD<sup>۷</sup>WCE<sup>۸</sup>SGDM<sup>۹</sup>

## فصل دوم

## روش‌ها

### ۲-۱ مجموعه داده

از مجموعه داده Kvasir در این پروژه استفاده شده است. برای آموزش، اعتبارسنجی و تست مدل تشخیص پولیپ، از محاسبه‌گر ابری گوگل کولب<sup>۱</sup> با رم ۱۲,۶۹ گیگابایت و حافظه دیسک ۶۸,۳۵ گیگابایت استفاده شده است.

در مجموعه داده Kvasir [۱۸]، تصاویر بدست آمده از مسیر گوارشی به ۸ سطح (دسته، کلاس) تقسیم می‌شوند، اما فقط دو دسته (پولیپ - غیرپولیپ) برای اجرای این تحقیق انتخاب شده‌اند. سایر دسته‌ها به ازوفازیت، خط Z (یک نوع سلول پوششی)، پیلوروس، سکوم، رکتوم و نشانه‌های دوگانه مربوط بوده و کاربردی برای این تحقیق ندارند. آن‌ها تشخیص چندین نوع بیماری گوارشی را صورت داده‌اند، که با هدف این مطالعه برابری ندارد.

جدول ۲-۱ اطلاعات مجموعه داده Kvasir

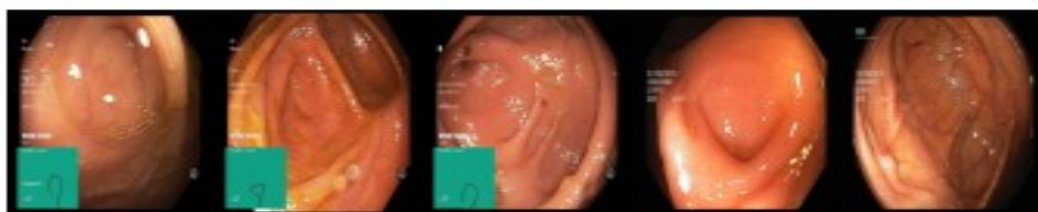
توصیف	Kvasir
نوع داده	تصویر رنگی
نوع فرمت	jpg
ابعاد داده	۵۷۶ x ۷۲۰
تعداد داده	پولیپ ۵۰۰ - سالم ۵۰۰

<sup>۱</sup> Google Colab

در صفحه بعد، شکل ۲-۱ و ۲-۲ به ترتیب نمونه‌هایی از مجموعه تصاویر حاوی پولیپ و بدون پولیپ مجموعه داده Kvasir را نشان می‌دهند.



شکل ۲-۱ نمونه تصاویر کولونوسکوپی حاوی پولیپ در مجموعه داده Kvasir



شکل ۲-۲ نمونه تصاویر کولونوسکوپی بدون پولیپ در مجموعه داده Kvasir

## ۲-۲ پیش پردازش تصاویر

در طول غربالگری اندوسکوپی، ویژگی‌های نوری دوربین تحت تأثیر ویژگی‌های بصری مختلفی همچون آرتیفکت‌ها [۲۶]، ویگنت‌ها [۲۶]، و درخشندگی‌ها [۷] قرار می‌گیرند. بنابراین قبل از استخراج ویژگی، تصاویر باید تحت پیش پردازش قرار بگیرند تا افت کیفیت به حداقل برسد.

### ۲-۲-۱ روش فیلتر میانی

در این شیوه، یک فیلتر میانی [۳۸] برای حذف اطلاعات غیرضروری بکار رفته است. این فیلتر به صورت همزمان تمامی پیکسل‌ها را با ناحیه میانی از پیش تعریف شده ( $3 \times 3$ ) جایگزین می‌کند. معادله ۱، تابع تعمیم یافته متناظر با هر همسایگی میانی را نشان می‌دهد. که در آن،  $k=1$  و مقدار میانی بر روی یک فیلتر  $3 \times 3$  محاسبه می‌شود. نتیجه‌ی کار، کاهش نویز در تصویر  $f(m,n)$  است.

$$f'(m,n) = \text{Med} \mid (-k \leq u,v \leq k)\{F(m+u, n+v)\}, \quad (1)$$

## ۲-۲-۲ روش آستانه گیری

علاوه بر این، فرایند آستانه گیری آروی تصاویر لحاظ شده است. با انجام این کار، اشیای مدنظر از پس زمینه جدا می‌شوند. تصاویر از نظر پیکسلی به دو نوع تقسیم می‌شوند: پیش زمینه و پس زمینه [۳۹]. به عنوان مثال، روش آستانه گیری به صورت خودکار، یک مقدار آستانه (T) را مشخص می‌کند. پیکسل‌هایی با مقدار کمتر از T، پیش زمینه، و تصاویری با مقدار بیشتر از T، پس زمینه محسوب می‌شوند. از سوی دیگر، نورپردازی یا تنظیمات نامناسب در منبع نوری کپسول آندوسکوپی، منجر به تغییر سطوح کنتراست تصاویر می‌شوند. با تغییر محدوده دینامیک تصاویر، کنتراست دستکاری شده و تصاویر یک دست می‌شوند [۴۰]. با افزایش محدوده دینامیک روی طیفی از تصاویر، امکان افزایش مقدار پیکسلی تصاویری با کنتراست کم و زیاد وجود دارد [۴۰]. این کار به کمک معادله‌ی ۲ صورت می‌گیرد.

$$O(x, y) = O_1 + \left( \frac{O_2 - O_1}{I_2 - I_1} \right) [I(x, y) - I_1], \quad (2)$$

در اینجا،  $O_1$  معادل با ۰، و  $O_2$  معادل با تعداد سطوح مطلوب در نظر گرفته می‌شوند.  $I_1, I_2$ ، مقادیر حداقلی و حداکثری سطح خاکستری هستند.

## ۲-۲-۳ نرمال سازی

نرمال سازی، فرایندی است که طی آن، ابعاد تصاویر تغییر یافته و به مقیاسی یکسان تبدیل می‌شوند. در این پروژه ابعاد (۲۲۴\*۲۲۴) برای نرمال سازی در نظر گرفته شده است. نرمال سازی مرحله‌ی مهمی است چرا که تضمین کرده تمامی تصاویر از مقیاس پیکسلی یکسانی برخوردارند. نرمال سازی تصویر اینگونه محاسبه می‌شود:

$$x' = \frac{x - \bar{x}}{\sigma}, \quad (3)$$

که در آن، X بردار ویژگی اصلی،  $X^-$  میانگین بردار ویژگی، و  $\sigma$  انحراف معیار هستند. شکل ۲-۳، مراحل نرمال سازی و پیش پردازش را طی نموده است.

threshold <sup>۲</sup>contrast <sup>۳</sup>



شکل ۲-۳ مثال‌هایی از تصاویری پس از پیش پردازش و نرمال‌سازی

## ۲-۳ استخراج ویژگی

استخراج ویژگی یکی از قدم‌های مهم در تشخیص محتوای تصاویر در حوزه یادگیری ماشین است. تکنیک‌های بسیاری برای این منظور تاکنون ارائه شده است. چند نمونه از این تکنیک‌ها عبارتند از هیستوگرام گرادیان‌های جهت دار [۴۱]، ویژگی‌های مقاوم تسریع شده [۴۲، ۴۱]، تبدیل ویژگی با مقیاس ثابت [۴۲] و الگوهای دودویی محلی [۴۳]. در پروژه حاضر، یادگیری عمیق، به عنوان یک تکنیک خودکار مبتنی بر شبکه عصبی پیچشی، برای استخراج ویژگی بکار رفته است. شبکه‌های عصبی پیچشی، امکان فراگیری ویژگی‌ها بین حجم عظیمی از تصاویر مختلف را دارا هستند [۱۰]. در پایان هر لایه پیچشی می‌توان نتایج حاصل از پردازش تصویر ورودی را به صورت دوبعدی مشاهده کرد. از این نتایج تصویری برای به دست آوردن بردار ویژگی استخراجی هر تصویر می‌توان استفاده نمود [۴۳، ۱۱].

علاوه بر این، عملکرد در حالت موازی، اجرای محاسبات ماتریسی را تسریع می‌کند. خروجی شبکه عصبی اینگونه محاسبه می‌شود:

$$y = \sigma(\omega^L \dots \sigma(\omega^2 \sigma(\omega^1 x + b^1) + b^2) \dots + b^L), \quad (4)$$

HOG<sup>۴</sup>

SURF<sup>۵</sup>

SIFT<sup>۶</sup>

LBP<sup>۷</sup>



که در آن  $x$  ورودی،  $w$  یک دسته از پارامترهای شبکه،  $b$  بایاس و  $y$  نورون خروجی هستند. همچنین تابع فعالسازی، نقش مهمی در یادگیری عمیق ایفا می‌کند. وظیفه‌ی این تابع غیرخطی، اجرای تبدیل ترکیبی است که امکان یادگیری روابط پیچیده تر را فراهم می‌کند. به عنوان مثال، تابع فعالسازی غیرخطی ReLU به صورت زیر تعریف می‌شود:

$$f(x) = \max(0, x), \quad (5)$$

از سوی دیگر، تابع بیشینه هموار روی لایه‌ی کاملاً متصل، به گونه‌ای است که مقدار هر گره را به احتمالات تبدیل می‌شوند. مجموع مقادیر این احتمالات برابر با ۱ است. معادله مربوطه به شرح زیر است:

$$S(y_i) = \frac{e^{y_i}}{\sum_{k=1}^K e^{y_k}}, \quad \text{for } i = 1, \dots, K, \quad (6)$$

که در آن  $y_i$  برابر مقدار ورودی  $i$  ام گره در لایه است.  $K$  تعداد اعداد حقیقی متناظر با توزیع احتمال یا همان تعداد گره های آن لایه است.

### ۱-۳-۲ یادگیری انتقالی با ResNet-50

ساخت یک مدل یادگیری عمیق، تحت تمامی شرایط به انبوهی از داده‌های آموزشی نیاز دارد. یادگیری انتقالی<sup>۸</sup> به عنوان یک راه حل بهینه و کاربردی، برای مدیریت داده‌های موجود بسیار کارآمد است. با بهره‌گیری از آن میتوان میزان دقت مدل را افزایش و زمان آموزش دهی را کاهش داد. یادگیری تقویتی خصوصاً در زمینه پزشکی که دچار محدودیت داده برچسب گذاری مواجه است، کاربرد دارد [۴۴]. در این روش از یک شبکه از پیش آموزش داده شده با دقت بالا در طراحی مدل استفاده میشود. در حال حاضر، شبکه‌های پیچشی آموزش داده شده متعددی همچون LeNet، SqueezeNet، AlexNet، GoogleNet و VGG در دسترس می‌باشند. در این پروژه، ResNet-50 با جایگزینی سه لایه نهایی، با ویژگی‌های جدید متناظر با مجموعه داده تصویری مورد تحلیل، تحت بازسازی قرار می‌گیرد. پس از آموزش مدل ساخته شده به کمک ResNet-50، در نهایت نمایش‌های ویژگی از طریق فعالسازی لایه کاملاً متصل (FC)، از روی تصاویر تست و آموزش بدست می‌آیند. لایه FC، ویژگی‌های ورودی را از تمامی نقاط تصویر جمع آوری کرده و به صورت بردار نمایش میدهد. بدین ترتیب در مجموع با ۲۰۴۸ ویژگی استخراج شده روبرو می‌شویم.

لازم بذکر است که شبکه‌های عصبی مختلف، از نظر دقت و زمان آموزش با یکدیگر تفاوت دارند. این موضوع به تعداد لایه‌های عمیق شبکه، و حجم مجموعه داده آموزش ربط دارد. در جدول ۲-۲ مقایسه‌ای از شبکه‌های معروف طبقه بندی صورت گرفته است. شبکه‌ی Inception-ResNet-V2 با اینکه بهترین دقت را دارد بدلیل محدودیت‌های سخت افزاری

<sup>۸</sup> softmax

<sup>۹</sup> Transfer Learning

پایه سازی شبکه‌های عصبی عمیق، کنار گذاشته شده و از شبکه RESNET-50 استفاده شده است. زمان محاسبات به اندازه‌ی شبکه عصبی و حجم داده‌ها بستگی دارد [۴۶]. یک شبکه عصبی عمیق، هم زمان پردازش و هم زمان آموزش طولانی‌تری خواهد داشت [۵۱].

جدول ۲-۲ مقایسه شبکه‌های از پیش آموزش‌دهی شده مختلف

نام شبکه	عمق شبکه	پارامتر (میلیون)	دقت %	زمان محاسبه تخمینی
AlexNet	۸	۶۱	۹۶,۱	۲۲
GoogLeNet	۲۲	۷	۹۶,۹	۶۷
ResNet-50	۵۰	۲۵,۶	۹۹,۰۲	۲۰۹
ResNet-101	۱۰۱	۴۴,۶	۹۸,۸	۲۸۸
VGG-19	۱۹	۱۴۴	۹۸,۴	۴۹۲
Inception- ResNet-V2	۱۶۴	۵۵,۹	۹۹,۳	۹۵۳

## ۲-۴ داده‌افزایی

یکی از چالش‌های یادگیری نظارتی کمبود داده برچسب دار است. در زمینه تصاویر پزشکی برچسب گذاری داده‌ها نیاز به دقت بسیار بالا و نظر متخصص مربوطه است [۴۴]. برای جبران حجم کوچک داده میتوان از روش‌های داده‌افزایی استفاده کرد. برخی از متداول‌ترین روش‌های داده‌افزایی عبارت‌اند از: داده‌افزایی و وضعیت‌ها، داده‌افزایی رنگ و داده‌افزایی با KNN [۵۹]. تعدادی از این روش‌ها برای داده‌های تصویری و برخی برای داده‌های برداری است. در این پروژه، روش KNN برای افزایش تعداد بردارهای ویژگی به کار گرفته شده است. طبق این روش برای هر بردار ویژگی نزدیک‌ترین همسایه به آن را پیدا کرده و سپس درونیابی و برون‌یابی آن دو را به مجموعه اضافه میکنیم [۵۹]. معادله ۷ و ۸، نحوه محاسبه بردار جدید را به روش درونیابی و درونیابی نشان میدهد.

$$\mathbf{c}' = (\mathbf{c}_k - \mathbf{c}_j)\lambda + \mathbf{c}_j \quad (7)$$

$$\mathbf{c}'_j = (\mathbf{c}_j - \mathbf{c}_k)\lambda + \mathbf{c}_j. \quad (8)$$

interpolation<sup>۱</sup>

extrapolation<sup>۱</sup>

که در آن  $c_j$  و  $c_j$  دو بردار همسایه و  $\lambda$  بین صفر و یک قرار دارد که درجه درون‌یابی و درون‌یابی را کنترل میکند. در این تحقیق آن را برابر با ۰.۵ قرار داده ایم.

## ۵-۲ طبقه بندی جمعی

در یادگیری نظارتی برای طبقه بندی، تابع مدنظر بر اساس داده‌های آموزشی برچسب گذاری شده تحت یادگیری قرار می‌گیرد. این روش در حوزه آمار و یادگیری ماشین بسیار پرکاربرد است. الگوریتم‌های مختلفی برای طبقه بندی وجود دارند. چند نمونه عبارتند از ماشین بردار پشتیبان<sup>۱۲</sup>، طبقه بند بیزی ساده، درخت تصمیم، نزدیک‌ترین همسایگی<sup>۱۳</sup> و AdaBoost. به منظور تعیین بهترین طبقه بند برای مدل پیشنهادی، استخراج کننده ویژگی HOG [۴۱] برای انجام آزمایشات روی تصاویر کولونوسکوپی به کار گرفته شده است که نتایج آن طبق جدول ۳-۲ گزارش شده است.

جدول ۳-۲ مقایسه طبقه‌بندهای مختلف

نام طبقه‌بند	دقت %
درخت تصمیم	۹۵.۷
بیز ساده	۹۷.۵
KNN	۹۷.۹
SVM	۹۸.۵
AdaBoost	۹۸.۷

با توجه به نتایج ارائه شده در جدول، مشاهده می‌شود که AdaBoost بالاترین دقت را به ارمغان می‌آورد، و SVM در ردیف بعدی قرار دارد. لذا برای طبقه بندی از AdaBoost استفاده شده است. هدف تفکیک تصاویر پولیپ از غیرپولیپ است. AdaBoost، یک الگوریتم یادگیری جمعی است که به عنوان جایگزین ابزار طبقه بندی تکی، برای طبقه بندی داده‌های مجهول بکار می‌رود. این روش بر خلاف سایر ابزارها از جمله SVM، دقت یادگیری یکسان را در زمان کوتاه‌تری حاصل می‌کند [۵۲].

با اعمال PCA قبل از آموزش طبقه بند، ضمن حفظ حداکثری تغییرات اساسی (مقادیر ویژه) در مجموعه داده، حجم آن را کاهش می‌دهیم (کاهش بُعدیت). PCA یک فرایند آماری است که به صورت گسترده برای تحلیل داده و کاهش ابعاد

<sup>۱</sup> SVM

<sup>۱</sup> KNN

داده بکار می‌رود. براساس بردارهای ویژه در PCA، داده‌های اصلی  $n$ -بعدی به  $m$ -بعدی تبدیل می‌شوند در داده‌های جدید  $m$ -بعدی، تعداد ویژگی‌ها برابر یا کمتر از ویژگی‌های اصلی است [۵۵]. آن‌ها هرگز بیشتر از ویژگی‌های اصلی نخواهند بود. با کاهش ۲۰۴۸ ویژگی استخراجی با PCA، ۹۰٪ واریانس داده‌ها پوشش داده شده و اکثر ویژگی‌های مهم در محاسبات باقی می‌مانند و تکرار اطلاعات به حداقل می‌رسد. پس از تحلیل PCA، بردار داده‌های ورودی و برجسته‌ها، برای آموزش مدل طبقه‌بند بکار می‌روند. علاوه بر این، الگوریتم جمعی، تصمیمات طبقه‌بندی مختلف را ترکیب کرده و با جبران طبقه‌بندی ضعیف، عملکرد کلی را بهبود می‌دهد [۵۶]. این موضوع در شکل ۲-۶ نشان داده شده است. در هر تکرار آموزش، وزن نمونه‌ها با توجه به نرخ خطا تغییر می‌کند [۵۷]. شبه کد AdaBoost در شکل ۲-۴ مشاهده می‌شود. شکل ۲-۵ ترکیب طبقه‌بندی‌ها برای طبقه‌بند نهایی  $Y(x)$  را نشان می‌دهد.

---

**Algorithm 1** AdaBoost
 

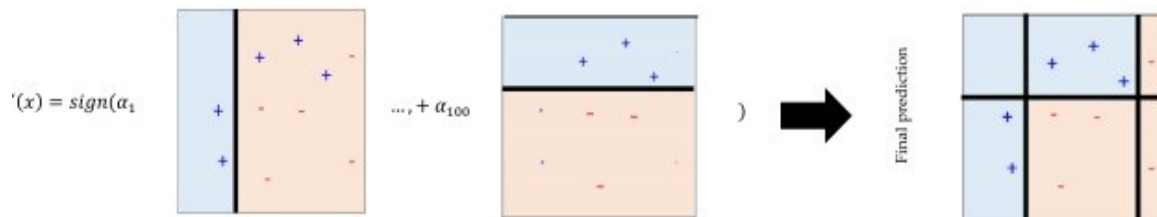
---

- 1: **Input:** a set of data  $\{(x_1, y_1), \dots, (x_{768}, y_{768})\}$
- 2: initialize  $w_1^i = 1/n$  for  $i = 1, 2, \dots, 768$
- 3: there are 100 classifiers
- 4: **for**  $k = 1: 100$
- 5: train weak classifier  $f_k(x)$  using  $w_k^i$
- 6: obtain error rate  $\varepsilon_k$  for 100 classifiers
- 7: calculate  $\alpha_k = 0.5 * \ln((1 - \varepsilon_k)/\varepsilon_k)$
- 8: re-weight, for  $i = 1, 2, \dots, 768$ :
 
$$w_{k+1}^i = \begin{cases} w_k^i * e^{\alpha_k} & \text{if } f_k(x_i) \neq y_i \\ w_k^i * e^{-\alpha_k} & \text{if } f_k(x_i) = y_i \end{cases}$$
- 9: get  $f_1(x), f_2(x), \dots, f_{100}(x)$  classifiers
- 10: weighted-sum 100 classifiers
- 11: **Output:** final classifier,

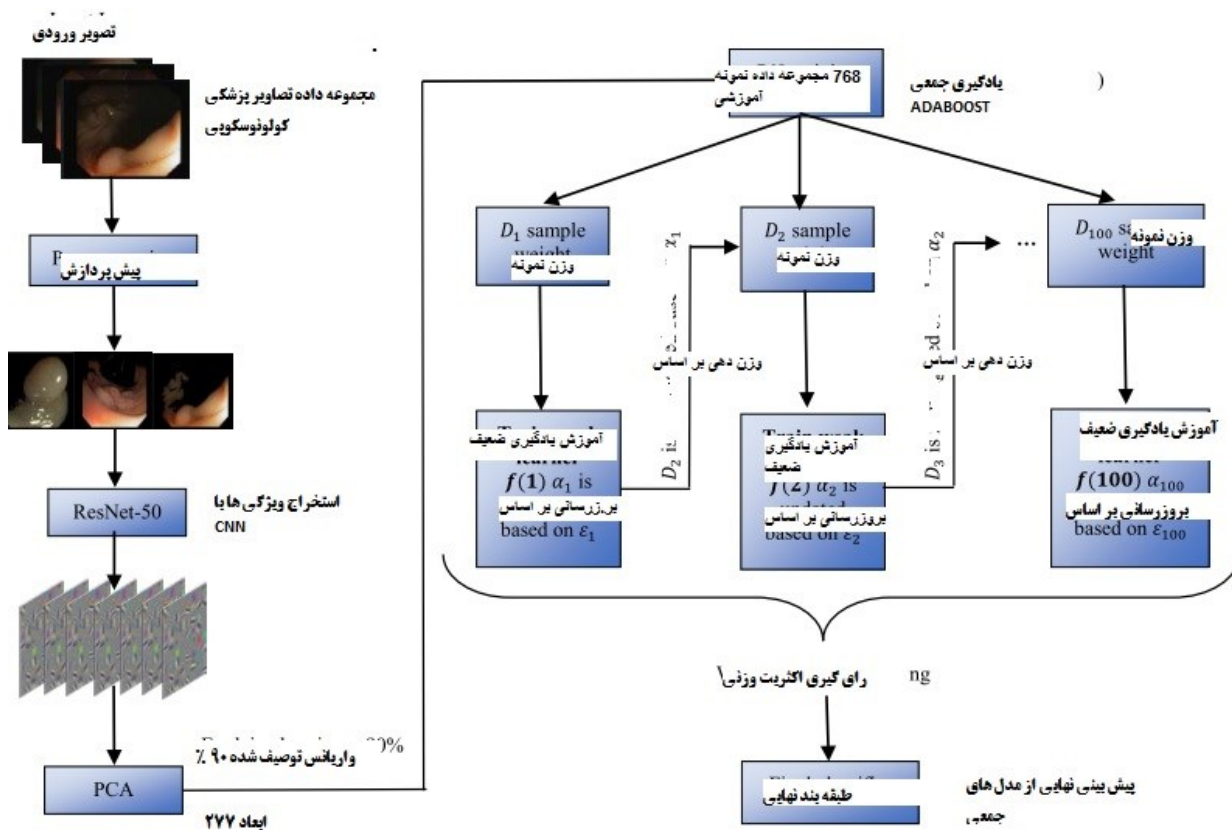
$$Y(x) = \text{sign} \left( \sum_{k=1}^{100} \alpha_k f_k(x) \right)$$


---

شکل ۲-۴ الگوریتم تقویتی - AdaBoost



شکل ۵-۲ رأ اکثریت وزنی برای هر پیش بینی



شکل ۶-۲ متدولوژی کلی روش یادگیری جمعی مبتنی بر تقویت، یادگیرنده‌های ضعیف تحت آموزش مکرر قرار می‌گیرند تا خطاهای مدل‌های قبلی اصلاح بشوند و مدل نهایی دارای کمترین انحراف باشد.

## ۶-۲ معیارهای سنجش عملکرد و ارزیابی

عملکرد سیستم CAD پیشنهادی، با محاسبه‌ی ماتریس پراکنش سنجیده می‌شود. این ماتریس به وفور در مسائل طبقه‌بندی آماری بکار می‌رود. این ماتریس حالت جدولی با دو سطر و دو ستون دارد و شامل پارامترهای زیر است :

مثبت صحیح (TP): پولیپی داخل فریم حاوی پولیپ شناسایی می‌شود.

مثبت کاذب (FP): پولیپی داخل فریم تصویر بدون پولیپ شناسایی می‌شود.

منفی صحیح (TN): هیچ پولیپی در فریم تصویر بدون پولیپ شناسایی نمی‌شود.

منفی کاذب (FN): یک پولیپ در فریم حاوی پولیپ از دست می‌رود (شناسایی نمی‌شود).

با توجه به پارامترهای فوق، حالت‌های رایج انحراف عملکردی از جمله دقت<sup>۱۴</sup> امتیاز F1، دقت تکرارپذیر<sup>۱۵</sup> مشخص‌کنندگی<sup>۱۶</sup>، نرخ خطا، حساسیت<sup>۱۷</sup> و نرخ مثبت کاذب تعریف می‌شوند :

$$\text{دقت} \quad Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% , \quad (9)$$

$$\text{نرخ خطا} \quad Error\ rate = \frac{FP + FN}{TP + TN + FP + FN} \times 100\% , \quad (10)$$

$$\text{دقت تکرارپذیر} \quad PPV\ (precision) = \frac{TP}{TP + FP} \times 100\% , \quad (11)$$

$$\text{حساسیت} \quad Sensitivity\ (TPR, recall) = \frac{TP}{TP + FN} \times 100\% , \quad (12)$$

$$\text{امتیاز F1} \quad F1\ score = 2 \times \left( \frac{precision \times recall}{precision + recall} \right) , \quad (13)$$

$$\text{مشخص‌کنندگی} \quad Specificity = \frac{TN}{TN + FP} \times 100\% , \quad (14)$$

Accuracy<sup>۱</sup>Precision<sup>۱</sup>Specificity<sup>۱</sup>Sensitivity<sup>۱</sup>

$$\text{نرخ مثبت کاذب} \quad FPR = \frac{FP}{FP + TN} , \quad (15)$$

امتیاز F1 معادل با میانگین وزنی معادله (۱۱) برای پیش بینی و (۱۳) برای بازیابی می‌باشد. حساسیت یعنی نسبت مقادیر مثبت بدرستی پیش بینی شده به تعداد کل حالت‌های مثبت، مشخص‌کنندگی یعنی نسبت مقادیر منفی بدرستی پیش بینی شده به تعداد کل حالت‌های منفی. همچنین منحنی مشخصه عملیاتی گیرنده<sup>۸</sup>، از طریق معادلات (۱۲) و (۱۵) قابل حصول است.

## فصل سوم

# پیاده سازی

در این فصل به نحوه پیاده سازی بخش های مدل پیشنهادی برای تشخیص پولیپ در عکس های کلونو سکوپي پرداخته شده است. در فصل قبل روش های به کار رفته معرفی شدند. در این فصل، پیاده سازی آن به زبان پایتون و با کتابخانه کراس<sup>۱</sup> شرح داده خواهد شد. همچنین برای تسریع فرایند یادگیری و ذخیره سازی بهتر نتایج از فضای ابری گوگل کولب<sup>۲</sup> استفاده شده است.

---

<sup>۱</sup> Keras

<sup>۲</sup> Google Colab



## ۳-۱ فراخوانی کتابخانه ها و توابع

به طور کلی در این پروژه از کتابخانه های OpenCV ، tensorflow ، matplotlib و sklearn استفاده شده است.

```
import cv2
import os
import numpy as np
import argparse
from matplotlib import pyplot as plt
from PIL import Image, ImageFilter
from skimage.io import imread_collection
from scipy import ndimage, misc
import glob
import pandas as pd
import random
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
import tensorflow.keras as keras
from keras.optimizers import Adam
import math
from shutil import copyfile
from keras.models import load_model
from tqdm.notebook import tqdm
from sklearn.decomposition import PCA
from scipy.spatial import distance
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.datasets import make_classification
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
```

شکل ۳-۱ فراخوانی کتابخانه ها و توابع

## ۳-۲ خواندن مجموعه داده و تقسیم آن به داده آموزشی و آزمایشی

در این بخش ابتدا آدرس محل نگهداری هر طبقه از مجموعه داده طبق شکل ۳-۲-۳ به ترتیب در org\_polyp و org\_normal ذخیره شده است.

```
# load the location of dataset
org_polyp = os.path.realpath('kvasir-dataset/kvasir-dataset/polyps/')
org_normal = os.path.realpath('kvasir-dataset/kvasir-dataset/normal-cecum/')
```

شکل ۳-۲ ذخیره آدرس مجموعه داده

در این قسمت از کد آدرس جدید تصاویر پیش از تقسیم بندی تعریف شده است. از آنجایی که مجموعه داده دارای تعداد یکسانی از دو طبقه است، تقسیم داده های آزمایشی و آموزشی به صورت متعادل انجام شده است. در `rnd_polyp` و `rnd_normal` آدرس ها ذخیره میشوند تا در نهایت با دستور `shuffle` به صورت تصادفی تصاویر هر پوشه در `train` و `test` ریخته شود.

```
# new folder address of train and test sets
trn_polyp = os.path.realpath('dataset/train/polyps/')
trn_normal = os.path.realpath('dataset/train/normal-cecum/')
val_polyp = os.path.realpath('dataset/test/polyps/')
val_normal = os.path.realpath('dataset/test/normal-cecum/')
# split train and test of images into two folders
dataset_size = 1000
split_rate = 0.7
rnd_polyp = [trn_polyp]*int(dataset_size*split_rate/2) + [val_polyp]*int(dataset_size/2-dataset_size*split_rate/2)
rnd_normal = [trn_normal]*int(dataset_size*split_rate/2) + [val_normal]*int(dataset_size/2-dataset_size*split_rate/2)
```

شکل ۳-۳ ذخیره آدرس جدید تصاویر مجموعه داده

تصاویر مجموعه داده که در دو پوشه `polyps` و `normal_cecum` قرار دارند، به دو مجموعه آموزشی و آزمایشی با در نظر گرفتن ۹۰ درصد مجموعه داده برای قسمت آموزشی جدا شده اند. در پوشه `dataset` در دو زیر پوشه `train` و `test` ریخته میشوند. هر کدام دو بخش `polyps` و `normal_cecum` دارند. در تصویر ۳-۴ پوشه مربوط به `polyps` در `train` و `test` ساخته شده است. تصاویر دارای پولیپ بر اساس `rnd_polyp` تقسیم میشوند.

```
# copy the images with polyp to train and test folders
try:
    os.makedirs(os.path.join(trn_polyp))
    os.makedirs(os.path.join(val_polyp))
except Exception:
    pass
files = os.listdir(os.path.join(org_polyp))
# shuffle to split the polyp images randomly
random.shuffle(rnd_polyp)
for filename, trg in zip(os.listdir(os.path.join(org_polyp)), rnd_polyp):
    scr = os.path.join(org_polyp, filename)
    dst = os.path.join(trg, filename)
    copyfile(scr, dst)
# copy the normal images to train and test folders
try:
    os.makedirs(os.path.join(trn_normal))
    os.makedirs(os.path.join(val_normal))
except Exception:
    pass
files = os.listdir(os.path.join(org_normal))
# shuffle to split the normal images randomly
random.shuffle(rnd_normal)
for filename, trg in zip(os.listdir(os.path.join(org_normal)), rnd_normal):
    scr = os.path.join(org_normal, filename)
    dst = os.path.join(trg, filename)
    copyfile(scr, dst)
# the result dataset with splited folders of train and test
DATA_DIR = 'dataset'
```

شکل ۳-۴ تقسیم تصاویر حاوی پولیپ

### ۳-۳ پیش پردازش تصاویر

اندازه اولیه تصاویر در مجموعه داده Kvasir  $720 \times 576$  است. برای آموزش بهتر شبکه به  $224 \times 224$  تغییر میکند. اندازه دسته ۱۶ در نظر گرفته است. مجموعه آموزشی به دسته هایی با نام `batches` به مدل برای آموزش داده می شود. دسته های ارزیابی از روی مجموعه آزمایشی ساخته شده و در `val-batches` ذخیره میشود.

```

SIZE = (224, 224)
BATCH_SIZE = 16
num_train_samples = sum([len(files) for r, d, files in os.walk(TRAIN_DIR)])
num_valid_samples = sum([len(files) for r, d, files in os.walk(VALID_DIR)])
num_train_steps = math.floor(num_train_samples/BATCH_SIZE)
num_valid_steps = math.floor(num_valid_samples/BATCH_SIZE)
gen = keras.preprocessing.image.ImageDataGenerator()
val_gen = keras.preprocessing.image.ImageDataGenerator()
batches = gen.flow_from_directory(TRAIN_DIR, target_size=SIZE,
                                  class_mode='categorical', shuffle=True, batch_size=BATCH_SIZE)
val_batches = val_gen.flow_from_directory(VALID_DIR, target_size=SIZE,
                                          class_mode='categorical', shuffle=True, batch_size=BATCH_SIZE)

```

شکل ۳-۵ پیش پردازش تصاویر

## ۴-۳ طراحی مدل

برای طراحی مدل شبکه پیچشی از مدل ResNet50 به عنوان مدل یادگیری انتقالی استفاده شده است. این مدل در ابتدا شبکه به عنوان مدل پایه با نام `base_model` قرار میگیرد. اندازه ورودی شبکه طبق مرحله پیش پردازش تصاویر  $224 \times 224$  با ۳ کانال رنگی در نظر گرفته شده است. در مرحله اول آموزش لایه ی ResNet50 آموزش داده نمیشود پس `trainable` آن برابر با False شده است. پس از لایه پایه یک لایه `globalAveragePooling` پیش از سه لایه کاملاً متصل قرار میگیرد. در آخرین لایه از `softmax` با دو گرهِ خروجی مربوط به دو طبقه قرار گرفته است.

طبق خلاصه مدل در شکل ۴-۳، مدل اولیه تنها ۴ هزار پارامتر آموزش پذیر دارد.

```
base_model = ResNet50(weights='imagenet', include_top=False)
base_model.trainable = False
inputs = keras.Input(shape=(224, 224, 3))
x = base_model(inputs, training=True)
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dropout(0.3)(x)
outputs = keras.layers.Dense(2, activation="softmax")(x)
model = keras.Model(inputs, outputs)
model.summary()
```

Model: "model\_6"

Layer (type)	Output Shape	Param #
input_6 (InputLayer)	[(None, 224, 224, 3)]	0
resnet50 (Functional)	(None, None, None, 2048)	23587712
global_average_pooling2d_2 (	(None, 2048)	0
dropout_2 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 2)	4098

Total params: 23,591,810  
 Trainable params: 4,098  
 Non-trainable params: 23,587,712

شکل ۴-۳ طراحی مدل

## ۵-۳ آموزش مدل

در مرحله اول مدل اولیه که لایه انتقالی آن آموزش پذیر نبود، آموزش داده میشود، شکل ۷-۳. در آموزش این مرحله با ۲۰ اپیک<sup>۱</sup> صورت گرفته است. برای جلوگیری از بیش‌برازش مدل دستورات توقف زود هنگام تعریف شده است. طبق این دستورات اگر تا ۵ اپیک خطای داده تست کاهش نیابد آموزش مدل متوقف میشود.

```
# the first time model is trained by freezed resnet50 model
model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])

early_stopping = EarlyStopping(patience=5)
checkpointer = ModelCheckpoint('resnet50_best.h5', verbose=1, save_best_only=True)
model.fit(batches, steps_per_epoch=num_train_steps, epochs=20,
        callbacks=[early_stopping, checkpointer],
        validation_data=val_batches, validation_steps=num_valid_steps)
model.save('resnet50_final.h5')
```

```
56/56 [=====] - 162s 3s/step - loss: 0.3137 - accuracy: 0.8586 - val_loss: 0.31978
Epoch 00006: val_loss improved from 0.31978 to 0.30935, saving model to resnet50_best.h5
Epoch 7/20
56/56 [=====] - 162s 3s/step - loss: 0.2910 - accuracy: 0.8835 - val_loss: 0.30935
Epoch 00007: val_loss improved from 0.30935 to 0.30565, saving model to resnet50_best.h5
Epoch 8/20
56/56 [=====] - 162s 3s/step - loss: 0.2643 - accuracy: 0.9005 - val_loss: 0.30565
Epoch 00008: val_loss improved from 0.30565 to 0.27851, saving model to resnet50_best.h5
Epoch 9/20
56/56 [=====] - 162s 3s/step - loss: 0.2695 - accuracy: 0.8903 - val_loss: 0.27851
Epoch 00009: val_loss improved from 0.27851 to 0.26112, saving model to resnet50_best.h5
Epoch 10/20
56/56 [=====] - 162s 3s/step - loss: 0.2436 - accuracy: 0.9038 - val_loss: 0.26112
```

شکل ۷-۳ آموزش مدل در مرحله اول

سپس در مرحله دوم آموزش، مدل پایه ResNet50 آموزش پذیر شده است. مدل مانند مرحله قبل با دستور fit آموزش داده میشود.

<sup>۱</sup> epoch

```
# the second time all layers of model are trained
base_model.trainable = True
model.compile(optimizer=Adam(lr=0.000001), loss='categorical_crossentropy', metrics=['accuracy'])

early_stopping = EarlyStopping(patience=5)
checkpointer = ModelCheckpoint('resnet50_fine_best.h5', verbose=1, save_best_only=True)
model.fit(batches, steps_per_epoch=num_train_steps, epochs=20,
        callbacks=[early_stopping, checkpointer],
        validation_data=val_batches, validation_steps=num_valid_steps)
model.save('resnet50_fine_final.h5')
```

```
56/56 [=====] - 633s 11s/step - loss: 0.2612 - accuracy: 0.8903 - val_loss: 0.31095
Epoch 00006: val_loss did not improve from 0.31095
Epoch 7/20
56/56 [=====] - 628s 11s/step - loss: 0.1986 - accuracy: 0.9152 - val_loss: 0.26831
Epoch 00007: val_loss improved from 0.31095 to 0.26831, saving model to resnet50_fine_best.h5
Epoch 8/20
56/56 [=====] - 629s 11s/step - loss: 0.1691 - accuracy: 0.9457 - val_loss: 0.23754
Epoch 00008: val_loss improved from 0.26831 to 0.23754, saving model to resnet50_fine_best.h5
Epoch 9/20
56/56 [=====] - 627s 11s/step - loss: 0.1552 - accuracy: 0.9400 - val_loss: 0.21154
Epoch 00009: val_loss improved from 0.23754 to 0.21154, saving model to resnet50_fine_best.h5
```

شکل ۸-۳ آموزش مدل در مرحله دوم

## ۶-۳ استخراج بردار ویژگی‌ها

برای استخراج بردارهای ویژگی از روی مدل آموزش داده شده در مرحله قبلی توابعی طراحی شده است که در شکل ۹-۳ قابل مشاهده است. تابع preprocess در ورودی آدرس تصویر را گرفته و تصویر مرتبط با آن را طبق اندازه ورودی شبکه در خروجی می‌دهد. تابع feature\_extractor آدرس تصویر را در ورودی می‌گیرد. این تابع تصویر را به مدل استخراج ویژگی می‌دهد. سپس در خروجی برداری به اندازه ۲۰۴۸، بردار ویژگی استخراجی، را قرار می‌دهد.

```

def preprocess(image_path):
    img = image.load_img(image_path, target_size=(224, 224))
    # Convert image to numpy array of 3-dimensions
    x = image.img_to_array(img)
    # Add one more dimension
    x = np.expand_dims(x, axis=0)
    # preprocess the images using preprocess_input() from ResNet50 module
    x = preprocess_input(x)
    return x
# Function to extract features of a given image into a vector of size (2048, )
def feature_extractor(image):
    image = preprocess(image) # preprocess the image
    fea_vec = fea_ext_model.predict(image) # Get the feature vector for the image
    fea_vec = np.reshape(fea_vec, fea_vec.shape[1]) # reshape from (1, 2048) to (2048, )
    return fea_vec

```

شکل ۹-۳ توابع استخراج ویژگی

با حذف سه لایه کاملاً متصل، مدل استخراج ویژگی ساخته شده است. در `data` بردار ویژگی هر تصویر با برچسب آن ریخته شده است. اطلاعات `data` سپس در دو بردار `X` و `y` مربوط به داده های آموزشی ذخیره میشود، شکل ۱۰-۳.

```

last = model.layers[-3].output
fea_ext_model = keras.Model(model.input, last)
polyps = [file for file in glob.glob('dataset/train/polyps/*.jpg')]
normals = [file for file in glob.glob('dataset/train/normal-cecum/*.jpg')]
data = []
for img in tqdm(polyps):
    data.append((feature_extractor(img), 1))
for img in tqdm(normals):
    data.append((feature_extractor(img), 0))

random.shuffle(data)
X = []
y = []
for d in data:
    X.append(d[0])
    y.append(d[1])
X = np.array(X)
y = np.array(y)
X_df = pd.DataFrame(X)
y_df = pd.DataFrame(y)
X_df.to_csv('kvasir_fea_extracted_X.csv', index=False, float_format='%f')
y_df.to_csv('kvasir_fea_extracted_y.csv', index=False)

```

شکل ۱۰-۳ استخراج ویژگی

طبق شکل ۳-۱۱، همانند آنچه برای داده آموزشی صورت گرفت، بردار های ویژگی برای داده تست نیز محاسبه میشود.

```
polyps_test = [file for file in glob.glob('dataset/test/polyps/*.jpg')]
normals_test = [file for file in glob.glob('dataset/test/normal-cecum/*.jpg')]
data_test = []
for img in tqdm(polyps_test):
    data_test.append((feature_extractor(img), 1))
for img in tqdm(normals_test):
    data_test.append((feature_extractor(img), 0))

random.shuffle(data_test)
X_test = []
y_test = []
for d in data_test:
    X_test.append(d[0])
    y_test.append(d[1])
X_test = np.array(X_test)
y_test = np.array(y_test)
```

شکل ۳-۱۱ استخراج ویژگی داده تست

### ۳-۷ تحلیل مؤلفه های اصلی

بردار ویژگی استخراج شده به طول ۲۰۴۸ است. این بردارها به روش تحلیل مؤلفه های اصلی با کاهش ۱۰ درصد واریانس طول اشان کاهش میابد. با این عمل داده ها برای ورود به مدل بعدی آماده می شوند.  $X_{total}$  شامل کلیه بردار های ویژگی آموزشی و تست است. این بردارها همزمان با دستور `fit_transform` به بردار با طول کوتاهتری تبدیل میشوند.

```
X_total = np.concatenate([X, X_test])
pca1 = PCA(n_components=1000)
pca1.fit(X_total)
n_comp = pca1.explained_variance_[pca1.explained_variance_ > 0.1].shape[0]
print(n_comp)
pca = PCA(n_components=n_comp)
x_total = pca.fit_transform(X_total)
x_pca = x_total[:900]
X_test = x_total[900:]
```

شکل ۳-۱۲ تحلیل مؤلفه های اصلی



## ۸-۳ داده‌افزایی بردارهای ویژگی آموزشی

به جهت یادگیری بهتر مدل یادگیری جمعی، تعداد بردارهای ویژگی آموزشی به روش نزدیک‌ترین همسایه افزایش یافته است. طبق این روش ابتدا نزدیک‌ترین بردار با برچسب یکسان به هر بردار پیدا شده و سپس با روش برون‌یابی و درون‌یابی بردار ویژگی جدید با همان برچسب ساخته شده و به مجموعه اضافه می‌شود.

```

n = len(x_pca)
for i in range(n):
    # find knn
    min = 10000000
    idx = -1
    min2 = 10000000
    idx2 = -1
    for j in range(n):
        if (i!=j) and (y[i]==y[j]):
            dst = distance.euclidean(x_pca[i],x_pca[j])
            if dst < min:
                min2 = min
                idx2 = idx
                min = dst
                idx = j
            elif dst < min2:
                min2 = dst
                idx2 = j
    # add interpolation
    new = 0.5*(x_pca[idx]-x_pca[i])+x_pca[i]
    x_pca = np.append(x_pca, [new], axis = 0)
    y = np.append(y,[y[i]], axis = 0)
    # add outerpolation
    new = -0.5*(x_pca[idx]-x_pca[i])+x_pca[i]
    x_pca = np.append(x_pca, [new], axis = 0)
    y = np.append(y,[y[i]], axis = 0)

```

شکل ۱۳-۳ داده‌افزایی

### ۳-۹ طبقه بندی با روش یادگیری جمعی

در این قسمت، مدل یادگیری جمعی AdaBoost با تخمین گر پایه درخت تصمیم طراحی شده است. طبق شکل ۳-۱۴، این مدل با داده های آموزشی  $X_{train}$  و برچسب  $y_{train}$  با دستور `fit` آموزش داده شد.

```
X_train = x_pca
y_train = y
sc = StandardScaler()
X_train_std = sc.fit_transform(X_train)
dtree = DecisionTreeClassifier(criterion='gini', random_state=40)
adbclassifier = AdaBoostClassifier(base_estimator=dtree,
                                   n_estimators=400,
                                   learning_rate=0.000005,
                                   algorithm = 'SAMME',
                                   random_state=40)
adbclassifier.fit(X_train, y_train)
```

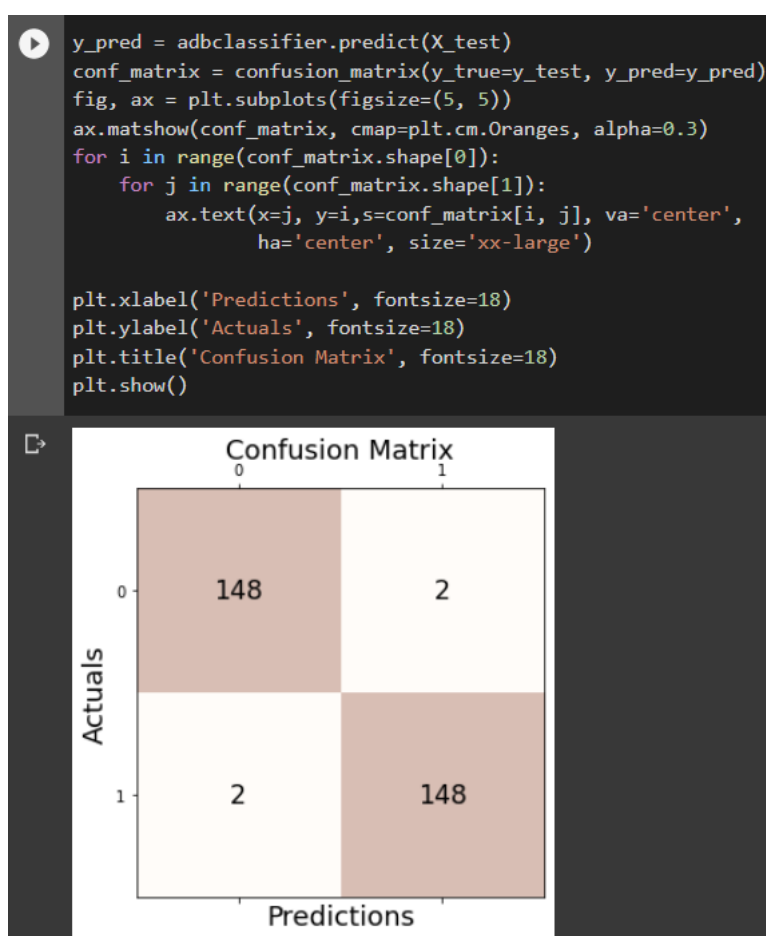
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using y = column\_or\_1d(y, warn=True)

```
AdaBoostClassifier(algorithm='SAMME',
                   base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                                           class_weight=None,
                                                           criterion='gini',
                                                           max_depth=None,
                                                           max_features=None,
                                                           max_leaf_nodes=None,
                                                           min_impurity_decrease=0.0,
                                                           min_impurity_split=None,
                                                           min_samples_leaf=1,
                                                           min_samples_split=2,
                                                           min_weight_fraction_leaf=0.0,
                                                           presort='deprecated',
                                                           random_state=40,
                                                           splitter='best'),
                   learning_rate=5e-06, n_estimators=400, random_state=40)
```

شکل ۳-۱۴ یادگیری جمعی با AdaBoost

### ۱۰-۳ سنجش عملکرد مدل

طبق شکل ۱۵-۳، ابتدا برچسب بردارهای ویژگی مجموعه تست توسط مدل AdaBoost پیش‌بینی می‌شود. از روی مقادیر پیش‌بینی شده و مقادیر واقعی برچسب‌ها ماتریس پراکنش با دستور `confusion_matrix` ساخته شد. این ماتریس توسط توابع `matplotlib` در خروجی نمایش داده شده است.



شکل ۱۵-۳ سنجش عملکرد مدل

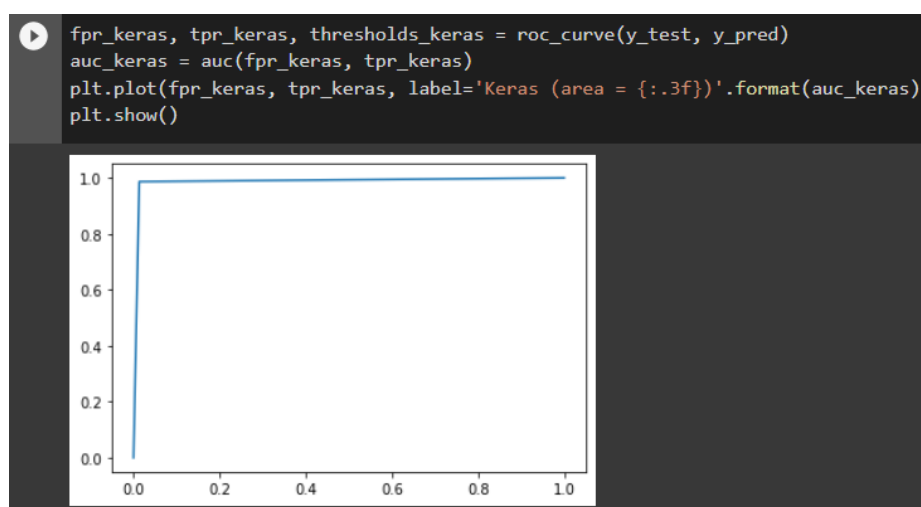
با استفاده با ماتریس پراکنش به دست آمده در دستورات قبل میتوان به راحتی میتوان حالت های رایج انحراف عملکردی را برای داده تست محاسبه کرد. در شکل ۳-۱۶ نحوه محاسبه هر کدام از سنجشگر های عملکرد و خروجی آن ها نمایش داده شده است.

```
tn = conf_matrix[0, 0]
tp = conf_matrix[1, 1]
fn = conf_matrix[1, 0]
fp = conf_matrix[0, 1]
print('Specificity: %.3f' % float(tn/(tn+fp)))
print('Sensitivity: %.3f' % float(tp/(tp+fn)))
print('FPR: %.3f' % float(fp/(fp+tn)))
print('Recall: %.3f' % recall_score(y_test, y_pred))
print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
print('F1 Score: %.3f' % f1_score(y_test, y_pred))
error_rate = 1- accuracy_score(y_test, y_pred)
print('Error: %.3f' % error_rate)
```

Specificity: 0.987  
Sensitivity: 0.987  
FPR: 0.013  
Recall: 0.987  
Accuracy: 0.987  
F1 Score: 0.987  
Error: 0.013

شکل ۳-۱۶ سنجشگر های مدل

با استفاده از کد شکل ۳-۱۷ میتوان نمودار ROC مربوط به مدل را نیز رسم نمود. از این مدل در فصل نتایج برای مقایسه مدل پیاده سازی شده با مقالات مرتبط استفاده شده است.



شکل ۳-۱۷ منحنی ROC

## فصل چهارم

## نتایج

### ۴-۱ عملکرد سیستم تشخیص رایانه‌ای (CAD)

در این بخش، نتایج ارزیابی عملکرد الگوریتم پیشنهادی در سیستم CAD را ارائه می‌دهیم. در جدول ۴، عملکرد جدیدترین و بهترین روش‌ها از جمله روش مقاله ای که برای پیاده سازی مدل استفاده شده، با پروژه حاضر مقایسه می‌شوند. پیاده سازی الگوریتم معرفی شده، فقط روی مجموعه داده Kvasir آزمایش شده است (۷۰٪ داده‌ها به صورت تصادفی برای آزمایش و ۳۰٪ برای تست بکار می‌روند).

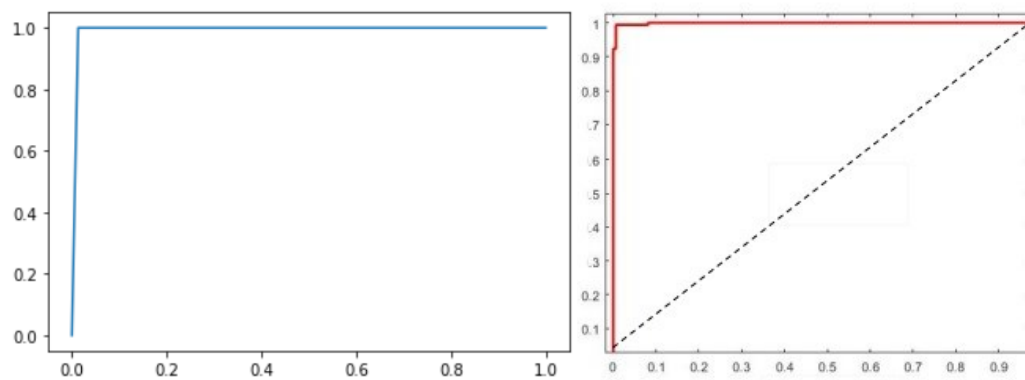
در تصمیم گیری پزشکی، منحنی ROC برای ارزیابی، یا به نمایش درآوردن عملکرد طبقه بندی بکار می‌رود [۵۸]. ROC یک منحنی احتمالاتی است و طی آن حساسیت (مساحت زیر منحنی) نسبت به FPR (نرخ منفی صحیح)، در آستانه‌های مختلف ترسیم می‌شود. از سوی دیگر، AUC شاخص تفکیک پذیری است و توانایی مدل برای تمایزدهی دسته‌ها را نشان می‌دهد. همچنین AUROC بالاتر (مساحت زیر منحنی مشخصه عملیاتی گیرنده) بیانگر عملکرد بهتر مدل است. در شکل ۴-۱ عملکرد سیستم مربوطه نشان داده شده است. برای این مدل، دقت، نرخ خطا، شاخص F1، حساسیت و مشخص کنندگی عمل طبقه بندی تصاویر پولیپ روده بزرگ، به ترتیب برابر با ۹۸٫۷٪، ۰٫۰۱۳٪، ۹۸٫۷٪، ۹۸٫۷٪ و ۹۸٫۷٪ هستند.

جدول ۴-۱ مقایسه عملکرد بین تحقیقات قبلی و روش پیاده سازی شده در مجموعه داده‌های مختلف از تصاویر کولونوسکوپی /

آندوسکوپی

روش	پایگاه داده	تکنیک	دقت	خطا	F1	حساسیت	مشخص‌کنندگی	FPR
پیاده سازی شده	مجموعه داده Kvasir	پیش پردازش CNN TFL ResNet50 PCA داده‌افزایی جمعی (AdaBoost)	۹۸,۷	۰,۰۱۳	۹۸,۷	۹۸,۷	۹۸,۷	۰,۰۱۳
تحقیق مرجع [۶۰]	مجموعه داده Kvasir	پیش پردازش CNN TFL ResNet50 PCA جمعی (AdaBoost)	۹۸,۶۴	۱,۳۶	۹۸,۶۲	۹۷,۹	۹۹,۳۸	۰,۰۰۶۲
ویتنب رگ ۲۰۱۹ [۲۰]	ETIS-LaribPolypD CVC- B و ClinicDB و Bayreuth	RCNN پوششی ResNet101 TFL	-	-	۸۳,۳	۸۷	-	-
لئو ۲۰۱۹ [۲۱]	ETIS-LaribPolypD CVC- B و ClinicDB و CVC-ColonDB	پیش پردازش چهارچوب SSD Inception-V3	-	-	۷۶,۸	۸۰	-	-
وانی ۲۰۱۹ [۲۲]	CVC-ColonDB فریم‌های ویدئو WCE، تصاویر آندوسکوپی از اندواتلس و شیلی	داده‌افزایی تکنیک یادگیری عمیق (VGG 19) با چهارچوب Keras	۹۴	۵,۵	۹۳	۹۴	-	-
ندیمی ۲۰۲۰ [۲۳]	تصاویر دریافتی از آندوسکوپی کپسول روده بزرگ	داده‌افزایی CNN TFL SGDM	۹۸	۲	-	۹۸	۹۶	-

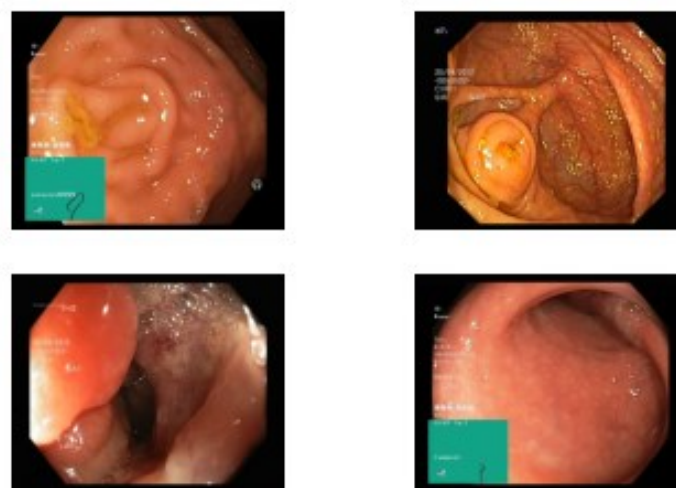
پاتینو برتینو س، ۲۰۲۰ [۲۴]	مجموعه داده خصوصی دانشگاه دوستو	پیش پردازش مدل CNN ۴ لایه VGG 16	۸۳	۱۷	۸۳	۸۶	-	-
--	---------------------------------------	---	----	----	----	----	---	---



شکل ۴-۱ مقایسه منحنی ROC برای طبقه بندی پولیپ، سمت راست منحنی تحقیق مرجع [۶۰] و سمت چپ منحنی مدل پیاده سازی شده

## ۴-۲ قضاوت غلط در سیستم تشخیص

مدل پیاده‌سازی شده، هنوز از مشکل دقت ناکافی یا قضاوت غلط روی تصاویر آندوسکوپی رنج می‌برد. تمامی تصاویر تست شده‌اند و مشخص شد که چندین تصویر از لحاظ بافت، تراکم و شکل دچار اشتباه هستند. بدین ترتیب سیستم ضایعاتی همچون زخم، التهاب و خونریزی را به عنوان پولیپ دسته بندی می‌کند. این مساله بشدت چالش برانگیز است، زیرا تصاویر پولیپ و بدون پولیپ شباهت‌های بسیاری دارند، از جمله شکل ظاهری، رنگ و ساختار سطحی. تمایزدهی این ویژگی‌ها بین آن‌ها بسختی صورت می‌گیرد. شکل ۴-۲ نشان دهنده‌ی تصاویر با دسته بندی غلط است. در ردیف اول شکل ۴-۲، تصویر نرمال به عنوان تصویر دارای پولیپ در مدل پیش‌بینی شده است. در ردیف دوم شکل، تصویر دارای پولیپ در دسته نرمال طبقه بندی شده است. علت این است که ساختار روده‌ی بزرگ ناموزون و تاخورده است، و گاهی اوقات با توجه به شکل و بافت، تصاویر به اشتباه پولیپ تشخیص داده می‌شوند.



شکل ۴-۲ تصاویر با دسته بندی غلط، سطر اول تصاویر با قضاوت اشتباه به عنوان پولیپ، سطر دوم تصاویر با قضاوت اشتباه به نرمال



## فصل پنجم

### بحث و بررسی

با در نظر گرفتن نمودارها و نتایج فوق، می‌توان گفت که روش پیاده‌سازی شده با استفاده از ResNet50 و AdaBoost و ترکیب خودکار آن‌ها، عملکرد طبقه‌بندی خوبی را بجا می‌گذارد. استفاده از لایه عمیق‌تر برای استخراج ویژگی، عملکرد طبقه‌بندی را در مقایسه با لایه‌های کم عمق افزایش می‌دهد. علت این است که لایه‌های عمیق‌تر، دارای ویژگی‌های سطوح بالاتری هستند. آن‌ها بر اساس ویژگی‌های سطوح پایین‌تر در لایه‌های پایین‌تر ساخته می‌شوند. همچنین، در پیاده‌سازی این مدل از داده‌افزایی به روش نزدیک‌ترین هم‌سایه استفاده شد. این روش در مقاله مرجع نیامده است اما عملکرد بهتری را نسبت به آن به جای گذاشته است. از سوی دیگر، طبقه‌بندها نقش مهمی در تمایزدهی ویژگی‌ها در مسائل طبقه‌بندی دارند. طبق جدول ۳-۲، AdaBoost به دلیل ماهیت تطبیقی‌اش بهتر از بقیه عمل می‌کند. با اجرای AdaBoost، طبقه‌بند قوی عملکرد کلی را افزایش می‌دهد. همچنین داده‌های بیشتر، عملکرد کلی را ارتقا می‌دهند. در نهایت، این روش نسبت به روش مقاله مرجع به دلیل مرحله داده‌افزایی دارای محاسبات بیشتری است اما در اجرای مختلف دقت‌های خروجی دارای پراکندگی کمتری هستند. بنابراین این مدل پیاده‌سازی شده از ثبات خوبی برخوردار است که آن را برای استفاده در تشخیص‌های پزشکی مناسب می‌کند.

## فصل ششم

# جمع‌بندی و نتیجه‌گیری و پیشنهادات

در این پروژه، روش جدید و خودکار تشخیص پولیپ با استفاده ترکیبی یادگیری انتقالی و یادگیری جمعی پیاده‌سازی شده است. استخراج ویژگی به کمک مدل ResNet50 انجام می‌گیرد. لایه‌های پیش‌بینی مدل ResNet50 به عنوان مدل پایه منتقل شده و لایه‌های کاملاً متصل جدید به آن اضافه شده است. پس از یادگیری مدل ویژگی‌ها از مدل استخراج می‌شود. از طرفی دیگر طبقه‌بندهای تطبیقی تقویتی جمعی، برای یادگیری و دسته‌بندی تصاویر بر اساس مؤلفه‌های اصلی استخراج شده از مجموعه داده، با برچسب‌های مربوطه بکار می‌روند (پولیپ و غیرپولیپ). در آزمایش روی ۱۰۰۰ تصویر، وجود پولیپ با دقت ۹۸٫۷٪، روی مجموعه داده Kvasir شناسایی شده است. نرخ خطای سیستم پیشنهادی در حد ۰٫۰۱۳٪ است. حساسیت، ویژگی و F1 به ترتیب ۹۸، ۹۸ و ۹۸٪ هستند. روش پیاده‌سازی شده در این پروژه روشی موثر برای شناسایی پولیپ در روده بزرگ است. در این روش نرخ خطا به طور چشمگیری پایین است. به خاطر ساختار غیرمنظم روده بزرگ، سیستم تشخیصی هنوز برخی تصاویر را به اشتباه طبقه‌بندی می‌کند.

## پیشنهادهای

در آینده، اثربخشی سیستم تشخیص رایانه‌ای می‌تواند با اعمال تکنیک‌های توجه و کاهش تعداد محاسبات بهبود پیدا کند. پیشنهاد میشود در سیستم‌های آینده متغیری به عنوان درصد احتمال وجود پولیپ نیز گزارش شود. در این صورت قضاوت‌های خطا در نتیجه نهایی تاثیرگذاری کمتری خواهد داشت.

## منابع و مراجع

- Ozawa, T. et al. Automated endoscopic detection and classification of colorectal polyps using convolutional neural networks. *Therap Adv Gastroenterol* 13, (2020). [١]
- Siegel, R. L. et al. Colorectal cancer statistics, 2020. *CA: A Cancer Journal for Clinicians* 70, 145–164 (2020). [٢]
- Paik, D. S. et al. Surface normal overlap: a computer-aided detection algorithm with application to colonic polyps and lung nodules in helical CT. *IEEE Transactions on Medical Imaging* 23, 661–675 (2004). [٣]
- Tajbakhsh, N., Chi, C., Gurudu, S. R. & Liang, J. Automatic polyp detection from learned boundaries. in 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI) 97–100 (2014). doi:10.1109/ISBI.2014.6867818. [٤]
- Stoitsis, J. et al. Computer aided diagnosis based on medical image processing and artificial intelligence methods. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 569, 591–595 (2006). [٥]
- Bose, B. K. Expert system, fuzzy logic, and neural network applications in power electronics and motion control. *Proceedings of the IEEE* 82, 1303–1323 (1994). [٦]
- Gueye, L., Yildirim-Yayilgan, S., Cheikh, F. A. & Balasingham, I. Automatic detection of colonoscopic anomalies using capsule endoscopy. in 2015 IEEE International Conference on Image Processing (ICIP) 1061–1064 (2015). doi:10.1109/ICIP.2015.7350962. [٧]
- Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 91–110 (2004). [٨]
- Chao, W.-L., Manickavasagan, H. & Krishna, S. G. Application of Artificial Intelligence in the Detection and Differentiation of Colon Polyps: A Technical Review for Physicians. *Diagnostics (Basel)* 9, (2019). [٩]
- Shin, Y., Qadir, H. A., Aabakken, L., Bergsland, J. & Balasingham, I. Automatic Colon Polyp Detection Using Region Based Deep CNN and Post Learning Approaches. *IEEE Access* 6, 40950–40962 (2018). [١٠]
- Bernal, J. et al. Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results From the MICCAI 2015 Endoscopic Vision Challenge. *IEEE Transactions on Medical Imaging* 36, 1231–1249 (2017). [١١]

- Tajbakhsh, N., Gurudu, S. R. & Liang, J. Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks. in 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI) 79–83 (2015). doi:10.1109/ISBI.2015.7163821. [١٢]
- Zhang, R., Zheng, Y., Poon, C. C. Y., Shen, D. & Lau, J. Y. W. Polyp detection during colonoscopy using a regression-based convolutional neural network with a tracker. *Pattern Recognition* 83, 209–219 (2018). [١٣]
- Tajbakhsh, N., GURUDU, S. R. & Liang, J. System and methods for automatic polyp detection using convolutional neural networks. (2016). [١٤]
- Brandao, P. et al. Towards a Computed-Aided Diagnosis System in Colonoscopy: Automatic Polyp Segmentation Using Convolution Neural Networks. *J. Med. Robot. Res.* 03, 1840002 (2018). [١٥]
- Yu, L., Chen, H., Dou, Q., Qin, J. & Heng, P. A. Integrating Online and Offline Three-Dimensional Deep Learning for Automated Polyp Detection in Colonoscopy Videos. *IEEE Journal of Biomedical and Health Informatics* 21, 65–75 (2017). [١٦]
- Zhang, R. et al. Automatic Detection and Classification of Colorectal Polyps by Transferring Low-Level CNN Features. From Nonmedical Domain. *IEEE Journal of Biomedical and Health Informatics* 21, 41–47 (2017). [١٧]
- Pogorelov, K. et al. KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. in *Proceedings of the 8th ACM on Multimedia Systems Conference* 164–169 (ACM, 2017). doi:10.1145/3083187.3083212. [١٨]
- Lin, T.-Y. et al. Microsoft COCO: Common Objects in Context. in *Computer Vision – ECCV 2014* (eds. Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T.) 740–755 (Springer International Publishing, 2014). doi:10.1007/978-3-319-10602-1\_48. [١٩]
- Mohammed, A., Yildirim, S., Farup, I., Pedersen, M. & Hovde, Ø. Y-Net: A deep Convolutional Neural Network for Polyp Detection. *arXiv:1806.01907 [cs]* (2018). [٢٠]
- Wittenberg, T., Zobel, P., Rathke, M. & Mühldorfer, S. Computer Aided Detection of Polyps in Whitelight- Colonoscopy Images using Deep Neural Networks. *Current Directions in Biomedical Engineering* 5, 231–234 (2019). [٢١]
- Liu, M., Jiang, J. & Wang, Z. Colonic Polyp Detection in Endoscopic Videos with Single Shot Detection Based Deep Convolutional Neural Network. *IEEE Access* 7, 75058–75066 (2019). [٢١]

- Vani, V. & Prashanth, K. V. M. Polyp Detection in Endoscopy Image Using Deep Learning. *EC Gastroenterology and Digestive System*, 663-672 (2019). [٢٣]
- Nadimi, E. S. et al. Application of deep learning for autonomous detection and localization of colorectal polyps in wireless colon capsule endoscopy. *Computers & Electrical Engineering* 81, 106531 (2020). [٣٤]
- Patino-Barrientos, S., Sierra-Sosa, D., Garcia-Zapirain, B., Castillo-Olea, C. & Elmaghraby, A. Kudo's Classification for Colon Polyps Assessment Using a Deep Learning Approach. *Applied Sciences* 10, 501 (2020). [٤٥]
- Mamonov, A. V., Figueiredo, I. N., Figueiredo, P. N. & Richard Tsai, Y.-H. Automated Polyp Detection in Colon Capsule Endoscopy. *IEEE Transactions on Medical Imaging* 33, 1488–1502 (2014). [٥٦]
- Sharon, H., Elamvazuthi, I., Lu, C.-K., Parasuraman, S. & Natarajan, E. Development of Rheumatoid Arthritis Classification from Electronic Image Sensor Using Ensemble Method. *Sensors* 20, 167 (2020). [٦٧]
- Alpaydin, E. *Introduction to machine learning*. (MIT Press, 2010) [٧٨]
- Polyp - Grand Challenge. [grand-challenge.org](http://grand-challenge.org) [٨٩]  
<https://polyp.grandchallenge.org/EtisLarib/>.
- Cvc-Clinicdb - Polyp - Grand Challenge. <https://polyp.grand-challenge.org/CVCClinicDB/>. [٩٠]
- Endoscopy: Types, preparation, procedure, and risks. [١٠١]  
<https://www.medicalnewstoday.com/articles/153737> (2017).
- Types of Endoscopy. Cancer.Net <https://www.cancer.net/navigating-cancer-care/diagnosing-cancer/tests-and-procedures/types-endoscopy> (2011). [١١٢]
- Marques, S., Bispo, M., Pimentel-Nunes, P., Chagas, C. & Dinis-Ribeiro, M. Image Documentation in Gastrointestinal Endoscopy: Review of Recommendations. *GE Port J Gastroenterol* 24, 269–274 (2017). [١٢٣]
- Rey, J. F., Lambert, R. & ESGE Quality Assurance Committee. ESGE recommendations for quality control in gastrointestinal endoscopy: guidelines for image documentation in upper and lower GI endoscopy. *Endoscopy* 33, 901–903 (2001). [١٣٤]
- Alaskar, H., Hussain, A., Al-Aseem, N., Liatsis, P. & Al-Jumeily, D. Application of Convolutional Neural Networks for Automated Ulcer Detection in Wireless Capsule Endoscopy Images. *Sensors* 19, 1265 (2019). [١٤٥]

- Liaqat, A. et al. Automated ulcer and bleeding classification from wce images using multiple features fusion and selection. *J. Mech. Med. Biol.* 18, 1850038 (2018). [۱۵۶]
- Charfi, S. & Ansari, M. E. Computer-aided diagnosis system for colon abnormalities detection in wireless capsule endoscopy images. *Multimed Tools Appl* 77, 4047–4064 (2018). [۱۶۷]
- Nelikanti, A. Colorectal Cancer MRI Image Segmentation Using Image Processing Techniques. in (2014). [۱۷۸]
- Jeyavathana, R. B., Balasubramanian, D. R. & Pandian, A. A. A Survey: Analysis on Pre-processing and Segmentation Techniques for Medical Images. /paper/A-Survey%3A-A-Analysis-on-Pre-processing-and-Techniques-Jeyavathana-Balasubramanian/9a2becc0f82325c0ff4d3aee625dbe5c58e7e3b8 (2016). [۱۸۹]
- Alginahi, Y. Preprocessing Techniques in Character Recognition. *Character Recognition* (2010) doi:10.5772/9776. [۱۹۰]
- Kibria, S. B. & Hasan, M. S. An analysis of Feature extraction and Classification Algorithms for Dangerous Object Detection. in 2017 2nd International Conference on Electrical Electronic Engineering (ICEEE) 1–4 (2017). doi:10.1109/CEEE.2017.8412846. [۲۰۱]
- Simon, A. State of the Art of Object Recognition Techniques. *Sci. Semin. Neuroscientific Syst. Theory* (2016). [۲۱۲]
- Shie, C.-K., Chuang, C.-H., Chou, C.-N., Wu, M.-H. & Chang, E. Y. Transfer representation learning for medical image analysis. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2015) doi:10.1109/EMBC.2015.7318461. [۲۲۳]
- Shie, C.-K., Chuang, C.-H., Chou, C.-N., Wu, M.-H. & Chang, E. Y. Transfer representation learning for medical image analysis. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2015) doi:10.1109/EMBC.2015.7318461. [۲۳۴]
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 1627–1645 (2010). [۲۴۵]
- Ansari, A. & Bakar, A. A. A Comparative Study of Three Artificial Intelligence Techniques: Genetic Algorithm, Neural Network, and Fuzzy Logic, on Scheduling Problem. in 2014 4th International Conference on Artificial Intelligence with [۴۶]

Applications in Engineering and Technology 31–36 (2014).  
doi:10.1109/ICAJET.2014.15.

Franklin, S. Learning high quality decisions with neural networks in “conscious” software agents. WSEAS Transactions on Systems. [47]

Canziani, A., Paszke, A. & Culurciello, E. An Analysis of Deep Neural Network Models for Practical Applications. (2016). [48]

Pretrained Deep Neural Networks - MATLAB & Simulink - MathWorks United Kingdom. <https://uk.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html>. [49]

Kornblith, S., Shlens, J. & Le, Q. V. Do Better ImageNet Models Transfer Better? in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2656–2666 (IEEE, 2019). doi:10.1109/CVPR.2019.00277. [50]

Whitley, D. Genetic Algorithms and Neural Networks. in Genetic Algorithms in Engineering and Computer Science 191–201 (John Wiley, 1995). [51]

Vink, J. P. & de Haan, G. Comparison of machine learning techniques for target detection. Artif Intell Rev 43, 125–139 (2015). [52]

Lu, H., Yang, L., Yan, K., Xue, Y. & Gao, Z. A cost-sensitive rotation forest algorithm for gene expression data classification. Neurocomputing 228, 270–276 (2017). [53]

Ibrahim, M. F. I. & Al-Jumaily, A. A. PCA indexing based feature learning and feature selection. in 2016 8th Cairo International Biomedical Engineering Conference (CIBEC) 68–71 (2016). doi:10.1109/CIBEC.2016.7836122. [54]

Zhu, M. et al. PCA and Kernel-based extreme learning machine for side-scan sonar image classification. in 2017 IEEE Underwater Technology (UT) 1–4 (2017). doi:10.1109/UT.2017.7890275. [55]

Lu, H., Meng, Y., Yan, K. & Gao, Z. Kernel principal component analysis combining rotation forest method for linearly inseparable data. Cognitive Systems Research 53, 111–122 (2019). [56]

Pang, S., Zhang, Y., Ding, M., Wang, X. & Xie, X. A Deep Model for Lung Cancer Type Identification by Densely Connected Convolutional Networks and Adaptive Boosting. IEEE Access 8, 4799–4805 (2020). [57]

Tartar, A. & Akan, A. Ensemble learning approaches to classification of pulmonary [58]

nodules. in 2016 International Conference on Control, Decision and Information Technologies (CoDIT) 472–477 (2016). doi:10.1109/CoDIT.2016.7593608.

Terrance DeVries, Graham W. Taylor. Dataset Augmentation in Feature Space. [59]  
Workshop track - ICLR (2017)

Win Sheng Liew, Tong Boon Tang, and Cheng-Kai Lu\*. Polyps Detection in [60]  
Colonoscopy Images Based on a Combination of Deep Feature Extraction and  
Ensemble Learning Techniques. (2021)