مقاله پژوهشی

مجله ماشین‌بینایی و پردازش‌تصویر
سال نهم، شماره سوم، پاییز ۱۴۰۱

# دسته‌بندی تصاویر پزشکی ضایعات پوستی با استفاده از شبکه عصبی کپسولی

نرگس حسن‌پور[۱]، امید اسلام[۲]، حدیث محسنی[۳*]

## چکیده

**شبکه‌های عمیق نوعی از روش‌های یادگیری هستند که قابلیت مدل کردن روابط سطح بالای موجود در داده‌ها را دارند. یکی از پرکاربردترین انواع شبکه‌های عمیق، شبکه‌های پیچشی یا کانولوشنی هستند که با بهره‌گیری از لایه‌های کانولوشن بر روی تصاویر قادر به مدل کردن وابستگی‌های مکانی در آن‌ها هستند، اما ساختارهای سلسله-مراتبی مکانی درون تصویر را در نظر نمی‌گیرند. شبکه‌های کپسولی یکی از ایده‌های جدیدی هستند که برای مدل‌سازی ساختار سلسله-مراتبی ویژگی‌ها در تصویر پیشنهاد شده‌اند و در آنها از کپسول یا نورون‌های گروه‌بندی شده به همراه یک الگوریتم مسیریابی پویا استفاده می‌شود. با وجود کارآیی ایده‌ی شبکه‌های کپسولی بر روی مجموعه‌داده‌های ساده، عملکرد این شبکه‌ها بر روی داده‌های پیچیده هنوز در ابهام است. در این مقاله عملکرد این شبکه بر روی مجموعه داده‌ی پیچیده‌ی سرطان پوست مورد بررسی قرار گرفته است که به دلیل اهمیت تشخیص ضایعات پوستی در پزشکی، پیچیدگی تصاویر، تعداد زیاد آنها و نامتعادل بودن دسته‌ها در آن انتخاب شده است. برای استخراج بهتر تنوع موجود در ضایعات پوستی، تغییراتی در لایه‌های ابتدایی شبکه داده شد و به دلیل عدم توازن در مجموعه‌داده‌ی ذکر شده، تغییراتی در تابع هزینه‌ی شبکه اعمال شد. تأثیر استفاده از توابع فعال‌سازی مختلف در شبکه نیز مورد بررسی قرار گرفت. نتایج به دست آمده نشان می‌دهد ایده‌ی شبکه کپسولی با انجام تنظیمات متناسب می‌تواند بر روی مجموعه‌داده‌های پیچیده نیز به نحو مطلوبی مورد استفاده قرار گیرد.**

## کلیدواژه‌ها

**شبکه کپسولی، دسته بندی، تصاویر پزشکی، تابع هزینه، تابع فعال‌سازی**

## ۱ مقدمه

شبکه‌های عصبی عمیق یک نوع شبکه عصبی مصنوعی با لایه‌های متعددی بین ورودی و خروجی هستند و تلاش دارند تا روابط سطح بالای موجود در داده‌ها را مدل نمایند و کاربردهای متعددی در زمینه پزشکی، ماشین‌های خودکار، دسته‌بندی تصاویر و ... دارند. یادگیری عمیق که به دو دسته‌ی با نظارت و بدون نظارت تقسیم می‌شود، در شبکه‌های متعددی مورد استفاده قرار گرفته است.

یکی از مهم‌ترین شبکه‌های عمیق، شبکه عصبی پیچشی یا کانولوشن (CNN) است. این شبکه یک الگوریتم یادگیری عمیق است که بیشتر بر روی داده‌های چند بعدی مانند تصاویر، مورد استفاده قرار می‌گیرد و در مقایسه با دیگر الگوریتم‌های دسته‌بندی در شبکه‌های عمیق به پیش پردازش[1] کمتری بر روی داده‌های

این مقاله در تیرماه ۱۴۰۰ دریافت، در آبان‌ماه بازنگری و سپس پذیرفته شد.

[۱] کارشناس ارشد هوش مصنوعی دانشگاه شهید باهنر کرمان
رایانامه: hasanpour.narges@eng.uk.ac.ir
[۲] کارشناس نرم‌افزار دانشگاه شهید باهنر کرمان
رایانامه: omid.es73@gmail.com
[۳] استادیار بخش مهندسی کامپیوتر دانشگاه شهید باهنر کرمان
رایانامه: hmohseni@uk.ac.ir

مولف مسئول: حدیث محسنی
DOR: 20.1001.1.23831197.1401.9.3.5.0

[1] Pre-processing

ورودی نیاز دارد که در واقع با آموزش دیدن به اندازه کافی، توانایی فراگیری ویژگی‌های داده ورودی را به دست می‌آورد. معماری این شبکه مشابه با الگوی اتصال نورون‌ها در مغز انسان است و از سازمان‌دهی قشر بصری[1] در مغز الهام گرفته شده است. هر نورون به محرک‌ها تنها در منطقه محدودی از میدان بصری که تحت عنوان میدان تأثیر[2] شناخته شده است پاسخ می‌دهد و مجموعه‌ای از این میدان‌ها برای پوشش دادن کل ناحیه بصری با یکدیگر هم‌پوشانی دارند. امروزه شبکه‌های CNN به دلیل کارآیی بالای که دارند، یکی از علل اصلی همه‌گیر شدن شبکه‌های یادگیری عمیق هستند. با این حال، آن ها دارای یک سری محدودیت‌ها و نقایص اساسی هستند. به طور کلی، یک شبکه عصبی کانولوشن از سه نوع لایه اصلی تشکیل می‌شود که عبارتند از لایه کانولوشن، لایه ادغام[3] و لایه تماماً متصل[4] که هر کدام از این لایه‌های مختلف وظایفی را انجام می‌دهند. استفاده از لایه‌های کانولوشن باعث می‌شود که وابستگی سطری و ستونی پیکسل‌های تصویر در لایه اول و همچنین وابستگی ویژگی‌های سطح بالاتر در لایه‌های بالاتر در نظر گرفته شود. با این حال، با استفاده از لایه ادغام در لایه‌های بالاتر، از قدرت یادگیری شبکه با توجه به وابستگی مکانی ویژگی‌ها کاسته می‌شود. به عبارت دیگر در لایه‌های انتهایی شبکه، مجموعه‌ای از ویژگی‌ها وجود دارند که مشخص نیست، ویژگی استخراج شده دقیقاً به کدام قسمت از تصویر مربوط است. استفاده از لایه ادغام در این شبکه‌ها صرفاً تضمین کننده این موضوع است که اگر تصویر در یک زاویه دید، کمی تغییر کند، شبکه می‌تواند همچنان پاسخ مناسبی را در مورد این تصویر ارائه کند، اما عیب این شبکه‌ها این است که ساختارهای سلسله-مراتبی مکانی[5] بین عناصر تشکیل دهنده‌ی اشیاء درون تصویر را در نظر نمی‌گیرند. معمولا آخرین لایه‌های یک شبکه عصبی کانولوشن را لایه‌های تماماً متصل تشکیل می‌دهند که یکی از کاربردهای اصلی آنها استفاده به عنوان طبقه‌بند است. مجموعه ویژگی‌های استخراج شده در لایه‌های کانولوشنی و ادغام در نهایت به صورت یک بردار ویژگی به یک طبقه‌بند تمام متصل داده می‌شود تا طبقه یا دسته خروجی صحیح را شناسایی کند.

یکی از جدیدترین ایده‌ها و رویکردها در شبکه‌های یادگیری عمیق کانولوشنی، که در سال ۲۰۱۷ معرفی شد، شبکه عصبی کپسولی[6] (CapsNet) [۱] است که هدف از ارائه آن بهبود عملکرد شبکه‌های عصبی کانولوشنی و رفع نقایص آنها در مدل کردن روابط سلسله مراتبی مکانی است. نکته کلیدی در این ایده آن است که نمود اشیاء در مغز به زاویه دید بستگی ندارد و روابط بین اشیاء سه بعدی با مفهومی به نام وضعیت[7] که عملا ترکیبی از انتقال[8] و چرخش[9] است بیان می‌گردد. برای انجام صحیح طبقه‌بندی و شناسایی تصاویر، حفظ روابط سلسله مراتبی وضعیت بین اجزاء هر شیء مهم است و موقعیت نسبی اشیاء و اجزای آنها باید مدنظر قرار گیرند. این مفهوم کلیدی، اهمیت تئوری کپسول‌ها را بیش از پیش روشن می‌سازد.

در رویکرد شبکه‌های کپسولی، نورون‌ها در قالب لایه‌های گروه‌بندی شده به منظور شناسایی اشیای تصویر به کار گرفته می‌شوند. به بیان دیگر، هر کپسول به منظور شناسایی یک ویژگی منحصر به فرد در یک تصویر طراحی شده است به گونه‌ای که بتواند در سناریوهای مختلفی همچون تشخیص اجسام در زاویه‌های مختلف مورد استفاده قرار گیرد. زمانی که چند کپسول در یک لایه موفق می‌شوند، جسمی را شناسایی کنند، آن‌ها کپسولی که در سطح بالاتری قرار دارد را فعال کرده و این روند ادامه پیدا می‌کند تا زمانی که شبکه بتواند درباره آنچه مشاهده کرده است، قضاوت کند.

یکی دیگر از خصوصیاتی که شبکه‌های کپسولی را از شبکه‌های کانولوشن قبلی متمایز می‌کند، استفاده از بردار بجای اسکالر برای ذخیره و انتقال اطلاعات بین نورون‌ها در شبکه است. عموماً شبکه‌های CNN برای داشتن عملکرد مطلوب، نیاز به حجم زیادی از داده‌ی آموزشی دارند. اما در برخی از کاربردها مانند داده‌های پزشکی، عملاً مشکل محدودیت در تعداد داده وجود دارد و شبکه‌های قوی‌تری مورد نیاز است تا با حجم کم داده هم به نتایج قابل قبولی برسند. شبکه‌های کپسولی سعی در برطرف کردن این مشکل با استفاده از فرم برداری به جای نمایش خروجی نورون‌ها به صورت اسکالر یا عددی دارند. در واقع کپسول‌ها همه‌ی اطلاعات مهم درباره‌ی وضعیت ویژگی مورد نظر در داده را در قالب یک بردار خلاصه و تجمیع می‌کنند و هر بردار با خود خاصیت اندازه و جهت را به همراه خواهد داشت.

ما در این تحقیق قصد داریم شبکه‌های کپسولی و کاربرد آن در مسائل طبقه‌بندی در تصاویر پیچیده‌ی پزشکی را با دقت بیشتری مورد بررسی قرار دهیم.

## ۲ مروری بر کارهای گذشته

استفاده از شبکه‌های کانولوشن یا CNN انقلاب عظیمی در شبکه‌های عصبی یادگیری عمیق ایجاد کرده است که از نمونه‌های آن می‌توان به تجزیه و تحلیل داده‌های پزشکی که از اهمیت بالایی برخوردار هستند، اشاره کرد. به عنوان مثال، بهبود کارآیی در روش‌های رادیولوژی [۲] و کاهش دوز دارویی در MRI مغز با میزان کنتراست زیاد، بدون کاهش قابل توجه در کیفیت تصویر [۳] از نمونه این تحلیل‌های پزشکی با استفاده از CNN هستند. اما از آنجایی که هدف از این مقاله بررسی کاربرد شبکه‌های کپسولی به

[1] Visual Cortex
[2] Receptive Field
[3] Pooling
[4] Fully Connected
[5] Spatial Hierarchical Structures
[6] Capsule Networks (CapsNet)
[7] Pose
[8] Translation
[9] Rotation

عنوان نمونه‌ای بهبود یافته از CNN است، تمرکز اصلی در این قسمت به مرور تعدادی از مقالات در حوزه تحلیل تصاویر پزشکی با استفاده از CapsNet معطوف می‌شود.

یکی از تحقیقات صورت گرفته در زمینه طبقه‌بندی تصاویر MRI مغز با شبکه عصبی CapsNet ، [۴] است که در آن تصاویر مغز در سه دسته طبقه‌بندی شده‌اند و دقت طبقه‌بندی بر روی معماری شبکه CapsNet به میزان ۸۶,۵۶% بوده است، در صورتی که شبکه مبتنی بر CNN به دقت ۷۲,۱۳% دست یافته است.

در مقاله ارائه شده در [۵]، عملکرد CapsNet بر روی یک مجموعه از تصاویر 2D و 3D سرطان ریه مورد بررسی قرار گرفته است. نتایج این مقاله نشان می‌دهند که در مواجهه با داده‌های آموزشی محدود، شبکه CapsNet بهتر از شبکه‌های کانولوشن معمولی عمل می‌کند.

با توجه به گسترش روز افزون COVID-19 تحقیقات بسیار گسترده‌ای بر روی مجموعه‌داده‌های موجود در این زمینه انجام می‌شود. مقاله [۶] با تغییر ساختار شبکه کپسولی و تابع هزینه بدون اینکه از قبل آموزش دیده باشد، به دقت ۹۵,۷ % و با آموزش دادن آن از قبل به دقت ۹۸,۳% دست یافته است. هم‌چنین [۷] در زمینه COVID-19 با شبکه کپسولی DenseCapsNet به دقت ۹۹,۳۳% رسیده است، در حالی که خود شبکه‌ی DenseNet121 بر روی این تصاویر به دقت ۸۷,۷۸% دست می‌یابد.

یکی از مجموعه‌داده‌های پیچیده‌ای که در زمینه تصاویر پزشکی وجود دارد، مجموعه‌داده‌های مربوط به سرطان پوست هستند. تشخیص سرطان پوست به دلیل پیچیدگی‌های ظاهری و انواع مختلف ضایعات پوستی کار دشواری است که حتی با روش‌های آزمایشگاهی غیر تهاجمی هم دقت بالایی ندارد. از نمونه کارهای پیشین در زمینه تشخیص ضایعات پوستی می‌توان به مقالات [۸] و [۹] اشاره کرد که از روش‌های کلاسیک یادگیری مانند خوشه‌بندی k-means، SVM و AdaBoost برای طبقه‌بندی ضایعات پوستی استفاده کرده‌اند. مقالات [۱۰] و [۱۱] هم از مجموعه‌های مختلفی از ویژگی‌ها از جمله نوع ضایعه، بافت، رنگ و غیره و شبکه‌های عصبی برای ساخت یک سیستم تشخیص قوی استفاده کرده‌اند.

در زمینه استفاده از روش‌های یادگیری عمیق برای تشخیص سرطان پوست، [۱۲] یک CNN با بیش از ۵۰ لایه برای طبقه‌بندی ضایعات بدخیم در مجموعه داده‌های چالش ISBI 2016 پیشنهاد داده است. مقاله [۱۳] از شبکه عصبی کانولوشن عمیق برای طبقه‌بندی تصاویر ضایعات پوستی استفاده کرده است. [۱۴] الگوریتمی را با استفاده از SVM همراه با یک رویکرد شبکه عصبی کانولوشن عمیق برای طبقه‌بندی تصاویر بالینی سرطان پوست پیشنهاد داده است و [۱۵] با استفاده از یک شبکه عصبی کانولوشن عمیق، تصاویر بیماری‌های پوستی در مجموعه‌داده ISIC 2016 را با کمک شبکه عصبی U-Net طبقه‌بندی کرده است.

مقاله [۱۶] عملکرد دو شبکه عصبی کانولوشن را بر روی مجموعه‌داده ISIC 2017 بررسی کرده است. در مقاله [۱۷]، روشی ارائه شده است که با استفاده از شبکه‌های CNN، به دقت ۹۰,۸% بر روی مجموعه‌داده ISIC 2016 دست یافته است .مقاله [۱۸] ترکیب دو شبکه عصبی، شبکه CNN و SVM را بر روی مجموعه‌داده‌ی ISIC 2016 را بررسی کرده است.

مقاله [19] با استفاده از ترکیب سه شبکه از پیش آموزش دیده‌ی VGG16 ، AlexNet و ResNet-18، ویژگی‌های تصاویر سرطان پوست را به دست آورده و از آنها برای طبقه‌بندی تصاویر با روش SVM استفاده کرده است. مقاله [۲۰] شبکه عصبی VGG16 را با شبکه عصبی کپسولی ترکیب کرده و عملکرد شبکه عصبی مورد نظر را بر روی مجموعه‌داده سرطان پوست بررسی کرده است.

در نهایت، [۲۱] از مجموعه‌داده‌ی ISIC در سال‌های ۲۰۱۸، ۲۰۱۹ و ۲۰۲۰ برای آموزش شبکه خود استفاده کرده است. این تحقیق از ۱۸ مدل و از شبکه‌های EfficientNet B3, B4, B5, B6, B7, SE-ResNeXt-101 , ResNeSt-101 که از قبل بر روی مجموعه‌داده‌ی بزرگی مانند ImageNet آموزش دیده‌اند، استفاده کرده با ترکیب آنها با متادیتا و استفاده سایزهای مختلف عکس در ورودی، به بهترین نتیجه یعنی ۹۴% دست یافته است.

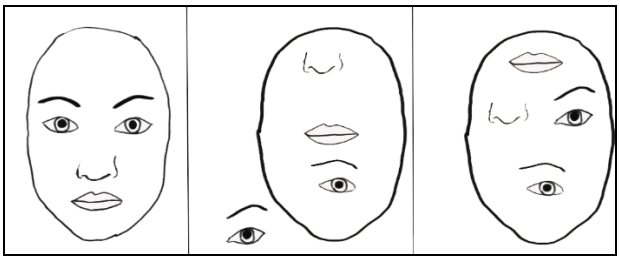
مروری بر کارهای انجام شده در زمینه تشخیص ضایعات پوستی نشان می‌دهد که هنوز جای تحقیقات بسیاری برای دستیابی به نتایج مطلوب در این زمینه وجود دارد. برای مثال، چالشی که هر سال بر روی مجموعه‌داده‌ی پیچیده‌ی ISIC برگزار می‌شود، موکد ضرورت انجام تحقیقات بیشتر در این زمینه است. به دلیل اینکه CapsNet یکی از جدیدترین ایده‌های مطرح شده در ساختار شبکه‌های کانولوشن است و عملکرد آن بر روی مجموعه‌داده‌های بزرگ و پیچیده به لحاظ ساختار و نامتعادل بودن داده‌ها به اثبات نرسیده است، تمرکز این مقاله بر استفاده از این شبکه عمیق بر روی مجموعه داده‌ی ISIC قرار داده شده است.

در این مقاله سعی شده که تأثیر توابع هزینه متفاوت و از طرف دیگر توابع فعال‌سازی متفاوت در شبکه کپسولی مورد بررسی قرار گیرد و شبکه در حالت‌های مختلف بر روی این مجموعه‌داده‌ی پیچیده آموزش داده شود تا به شناخت رفتار آن کمک کند. هم‌چنین با تغییرات جزئی در معماری شبکه و اضافه کردن تعدادی لایه کانولوشنی پیش از مرحله رمزگشا، به استخراج بهتر ویژگی‌های مؤثر از داده‌های پیچیده کمک می‌شود.

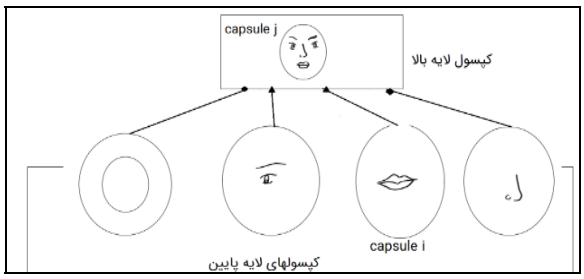
در ادامه مقاله، در بخش ۳، ساختار شبکه‌های کپسولی مورد بررسی قرار می‌گیرد. در بخش ۴، کارهای انجام شده و تغییرات پیشنهاد شده در ساختار این شبکه با جزئیات ذکر می‌شوند. در بخش ۵، مجموعه‌داده‌ی استفاده شده معرفی شده و شبکه در حالت‌های مختلف با این مجموعه‌داده آموزش می‌بینند و نتایج به دست آمده مورد بررسی قرار می‌گیرد. در نهایت بخش نتیجه‌گیری به جمع‌بندی نتایج به دست آمده در بخش ۵ شرح داده می‌شود.

## ۳ ساختار شبکه کپسولی

در شبکه کپسولی، هر کپسول شامل گروهی از نورون‌ها است که حضور و موقعیت یک شی خاص را در یک مکان مشخص پیش‌بینی می‌کنند. خروجی کپسول، بردار فعال‌سازی است که جهت آن اطلاعات موقعیت جسم را مانند مکان و مقیاس آن را نشان می‌دهد و طول بردار فعال‌سازی نشان دهنده‌ی احتمال وجود شیء مورد نظر در مکان خاص است. به طور مثال اگر یک جسم را بچرخانیم، بردارهای فعال‌سازی به همان نسبت تغییر می‌کنند ولی طول آنها ثابت می‌ماند. به بیان دیگر، کپسول‌ها سعی می‌کنند ویژگی‌های پیچیده‌تری را در ورودی خود تشخیص دهند. به عنوان مثال چهره‌ی انسان از چشم، ابرو، دهان و بینی تشکیل شده است. در شکل ۱، شبکه‌های عصبی کانولوشنی می‌توانند وجود چهره را در یک تصویر تشخیص دهند چون چهره شامل هر کدام از این اجزا (دهان و بینی و...) هست، اما روابط مکانی بین اجزای چهره را در نظر نمی‌گیرند. به عبارت دیگر اگر هر کدام از این اجزاء به طور مناسب در مکان خود قرار نگیرند (به طور مثال جای دهان و بینی در تصویر جابجا شود)، باز هم شبکه‌های کانولوشنی تصویر را چهره تشخیص می‌دهند در صورتی که تصویر کاملا به هم ریخته است.



**شکل ۱ شبکه عصبی کانولوشنی روابط مکانی بین هر کدام از اجزاء چهره (دهان،بینی،چشم و ابرو) را در نظر نمی‌گیرد و هر دو تصویر (تصویر سمت چپ و تصویر سمت راست) را چهره تشخیص می‌دهد، چون هر دو تصویر از اجزای یکسانی تشکیل شده‌اند.**




**شکل ۲ شبکه عصبی کپسولی با ایجاد بردار فعال‌سازی بین کپسول‌های لایه پایین (چشم،ابرو و دهان و...) و کپسول لایه بالاتر (چهره) و استفاده از طول و جهت بردارها و جستجوی توافقی آرای بین کپسول‌ها، سعی می‌کند که کپسول چهره را با روابط مکانی صحیح هر کدام از اجزای در لایه بالاتر تشخیص دهد.**

در شکل ۲ شبکه عصبی کپسولی به هر کدام از اجزاء چهره اهمیت می‌دهد به این صورت که هر کدام از اعضای چهره(مانند دهان، چشم، ابرو و بینی) را کپسول‌های لایه اول در نظر می‌گیرد و بر اساس بردار فعال‌سازی، طول و جهت آن و همچنین با جستجوی توافقی آرای بین هر کدام از کپسول‌ها، سعی می‌کند که کپسول چهره را با روابط مکانی صحیح هر کدام از اجزای در لایه بالاتر تشخیص دهد.

### ۳-۱ معماری شبکه کپسولی

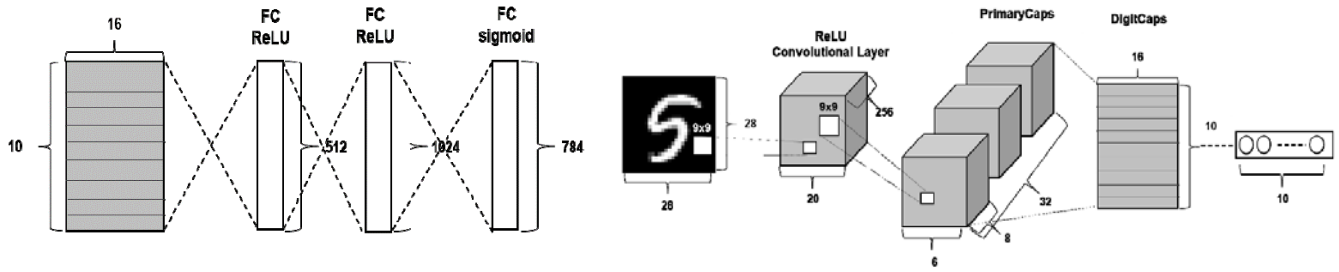
از لحاظ معماری، هر شبکه کپسولی دارای دو بخش اصلی رمزنگار[1] و رمزگشا[2] است. بخش رمزنگار وظیفه تبدیل تصویر ورودی به یک بردار را دارد که این بردار، نماینده ویژگی‌ها و اطلاعات موجود در تصویر ورودی است. بخش رمزگشا که فقط در زمان آموزش شبکه استفاده می‌شود، وظیفه ساخت تصویر اصلی از روی بردار ساخته شده توسط رمزنگار را دارد تا مطمئن شود که بردار تولید شده توسط رمزنگار یک توصیف کننده خوب از تصویر ورودی است. حالت پایه و اصلی این شبکه که در [۱] ارائه شده است، معماری شبکه را بر اساس مجموعه‌داده MNIST بنا نهاده است که این مجموعه‌داده شامل تصاویر ۲۸*۲۸ از اعداد دست‌نویس انگلیسی است و در آن ۱۰ دسته وجود دارد. مسلماً بسته به نوع داده و کاربرد، تعداد لایه‌ها در هر بخش از شبکه قابل تنظیم است.

#### ۳-۱-۱ بخش رمزنگار

در معماری شبکه کپسولی که در [۱] ارائه شده، بخش رمزنگار شبکه، دارای سه لایه‌ی اصلی است. همان‌طور که در شکل ۳-الف مشخص است بخش رمزنگار شبکه کپسولی شامل لایه کانولوشنی، لایه کپسولی-اصلی و لایه کپسولی-رقمی است. وظیفه این لایه‌ها این است که با شناسایی و ترکیب ویژگی‌های استخراج شده توسط کپسول‌ها بتوانند ویژگی‌های با اهمیت‌تری را بدست آورند و هر کدام از روابط بین اجزای را با استفاده از جستجوی توافقی بین رای‌ها حفظ کنند:

**لایه ۱- لایه کانولوشن:** وظیفه این لایه شناسایی ویژگی‌های ابتدایی تصویر ورودی است. در [۱]، این لایه دارای ۲۵۶ هسته[3] با ابعاد ۹ * ۹ و گام[4] معادل یک و تابع فعال‌سازی ReLU[5] است. است.

[1] Encoder
[2] Decoder
[3] Kernel
[4] Stride
[5] Rectified Linear Unit




**شکل ۳ معماری شبکه کپسولی . الف) بخش رمزنگار شبکه کپسولی شامل سه لایه است. لایه ۱: کانولوشن، لایه ۲: لایه کپسولی-اصلی، لایه ۳: لایه کپسولی-رقمی. هدف اصلی این کپسول‌ها استخراج ویژگی‌های مهم تصویر ورودی و روابط آنهاست. ب) بخش رمزگشا شبکه کپسولی که فقط در زمان آموزش استفاده می‌شود، شامل سه لایه تماما متصل است و وظیفه بازسازی تصویر ورودی را بر عهده دارد.**

**لایه ۲- لایه کپسولی-اصلی[1]** : وظیفه این لایه این است که ویژگی‌های ابتدایی شناسایی شده توسط لایه اول را دریافت کرده و ترکیباتی از این ویژگی‌ها را تولید کنند. در [۱]، در این لایه ۳۲ گروه از کپسول‌های هشت بعدی وجود دارد. یعنی هر کپسول دارای ۸ هسته‌ی مختلف با گام برابر ۲ و سایز ۹*۹ است که بر روی ورودی‌های ۲۵۶*۲۰*۲۰ اعمال می‌شوند و نهایتاً نتیجه را به تنسورهایی با ابعاد ۶*۶*۸*۳۲ تغییر اندازه می‌دهد.

**لایه ۳- لایه کپسولی-رقمی[2]** : در این لایه که تماماً متصل و همان لایه خروجی نیز هست، به اندازه‌ی تعداد دسته‌ها کپسول داریم. بر اساس پیاده‌سازی شبکه در [۱]، بردار ورودی ۸ بعدی هر کپسول توسط ماتریس وزن ۱۶*۸ آن به فضای خروجی ۱۶ بعدی آن کپسول نگاشت می‌شود. هر بعد از بردار فعال‌سازی ۱۶ بعدی ویژگی‌های خاصی از جسم را نشان می‌دهد و هر کدام از ۱۶ بعد می‌توانند منجر به ایجاد یک ویژگی شوند. از آنجا که در بخش رمزنگار فرض بر انتخاب فقط یک دسته است، لایه کپسولی-رقمی باید تغییرات در ضخامت، مقیاس، انحراف و ... در ورودی خود را از طریق همین ماتریس وزن یاد بگیرد.

### ۲-۱-۳- بخش رمزگشا

این بخش از شبکه که از سه لایه تمام-متصل تشکیل شده است (شکل ۳-ب)، فقط در مرحله آموزش استفاده می‌شود. رمزگشا، به عنوان یک تنظیم کننده مورد استفاده قرار می‌گیرد. به این معنی که خروجی صحیح لایه کپسولی-رقمی که یک بردار ۱۶ بعدی است را دریافت کرده و می‌آموزد که این بردار را به تصویر ورودی متناظر با آن بازسازی کند. تابع هزینه این کار، فاصله اقلیدسی بین تصویر بازسازی شده و تصویر ورودی (اصلی) است. رمزگشا کپسول‌ها را مجبور می‌کند که ویژگی‌هایی را بیاموزند که برای بازسازی تصویر اصلی مفید هستند.

## ۲-۳ اتصال لایه‌ها و مسیریابی در شبکه

برای هر مجموعه‌ی ۶*۶ از کپسول‌ها در لایه‌ی پایین (لایه‌ی کپسولی-اصلی)، طبق رابطه ۱ یک ماتریس وزن (W) با اندازه ۱۶*۸ در نظر گرفته می‌شود و در ورودی (u) ضرب می‌شود. ماتریس وزن نشان دهنده این است که هر یک از نورون‌های کپسول هشت بعدی تا چه میزان رأی به هر یک از نورون‌های کپسول ۱۶ بعدی می‌دهند. $\hat{u}_{j|i}$ بردار پیش‌بینی خروجی کپسول j براساس کپسول i است.

$$\hat{u}_{j|i} = W_{ij} u_i \qquad (۱)$$

که اندیس i و j به ترتیب نشان‌دهنده لایه‌ی پایین و لایه‌ی بالا (لایه‌ی کپسولی-رقمی) هستند.

در مرحله بعد بین لایه‌ی کپسولی پایین و بالا، از الگوریتم مسیریابی پویا[3] (شکل ۴) استفاده می‌شود که به آن مسیریابی یا رأی دهی نیز گفته می‌شود. این الگوریتم بین هر کپسول در لایه i و لایه j یک ضریب اولیه $b_{ij}$ (ضریب مشارکت) با مقدار صفر تعریف می‌کند. سپس طبق رابطه ۲ با استفاده از تابع غیرخطی Softmax، ضریب مسیریابی بین لایه پایین و بالا ($C_{ij}$) محاسبه می‌شود.

$$c_{ij} = \text{softmax}(b_{ij}) = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \qquad (۲)$$

اگر مقدار $C_{ij}$ برابر یک باشد، یعنی تمام اطلاعات کپسول i ام به کپسول j ام منتقل می‌شوند. سپس این مقادیر را در بردارهای پیش‌بینی ضرب می‌کنیم و به ازای هر کپسول لایه j همه مقادیر ضرب شده در لایه i را با هم جمع می‌کنیم. در بار اول اجرای الگوریتم، همه‌ی بردارهای پیش‌بینی در ۰,۵ ضرب می‌شوند (چون مقدار Softmax(0)=0.5 است) و با هم جمع زده می‌شوند یعنی همه کپسول‌های لایه i برای بدست آوردن مقادیر لایه j با ضریب ۰,۵ رأی داده‌اند. از آنجایی که مقدار این حاصل‌جمع و طول بردار خروجی کپسول ممکن است بزرگ‌تر از یک شود، طبق رابطه ۳ از

[1] PrimaryCaps Layer
[2] DigitCaps Layer
[3] Dynamic Routing

**Algorithm ۱** Routing

**procedure** Routing( r،l)

**for all** capsule $i$ in layer $l$ and capsule j in layer $(l+۱)$ **do**

$b_{ij} = ۰$

**for** $r$ iterations **do**

**for all** capsule $i$ in layer $l$ **do**

$c_i = softmax(b_i)$

**for all** capsule $j$ in layer $(l+۱)$ **do**

$s_j = \sum_i c_{ij}\hat{u}_{(j|i)}$

**for all** capsule $j$ in layer $(l+۱)$ **do**

$v_j = squash(s_j)$

**for all** capsule $i$ in layer $l$ and capsule $j$ in layer $(l+۱)$ **do**

$b_{ij} = b_{ij} + \hat{u}_{(j|i)}.v_j$

**return** $v_j$

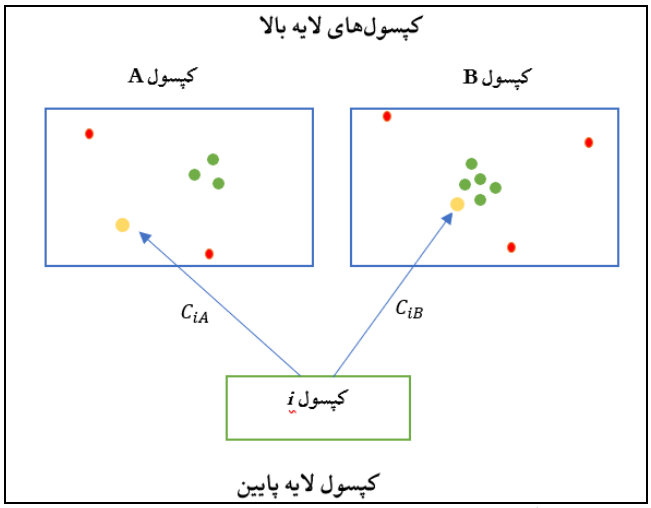
**شکل ۴ الگوریتم مسیریابی پویا [۱]. این الگوریتم طی سه مرحله(r) رأی دهی و بر اساس آراء کپسول‌های لایه پایین (i)، کپسولی را از سطح پایین انتخاب کند که بیشترین شباهت با کپسول لایه بالا (j) را دارد، و این کار را چند بار تکرار می‌کند تا به همگرایی(توافق بین آراء) برسد.**

تابع Squash برای نرمال‌سازی آن استفاده می‌کنیم تا خروجی کپسول‌ها همواره عددی در بازه [۰,۱] باشد. در واقع این عدد نشان‌دهنده احتمال وجود ویژگی اختصاص یافته به این کپسول در تصویر ورودی باشد.

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \quad (۳)$$

در این رابطه $s_j$ همان مجموع محاسبه شده در کپسول و $v_j$ خروجی یا رأی نرمال شده کپسول j ام است. بر اساس این رابطه، اگر همه کپسول‌های لایه i در مورد یک ورودی اتفاق نظر داشته باشند، $s_j$ بزرگ شده و $v_j$ به یک نزدیک می‌شود و برعکس.

شکل ۵ نشان دهنده‌ی عملکرد الگوریتم مسیریابی بین کپسول‌های لایه پایین و کپسول‌های لایه بالا است. کپسول سطح پایین باید تصمیم بگیرد که خروجی خود را چگونه به هر کدام از کپسول‌های سطح بالاتر ارسال کند و این تصمیم را با استفاده از ضرب ضریب $C_{ij}$ در بردارهای پیش‌بینی $\hat{u}_{(j|i)}$ اتخاذ می‌کند. مطابق شکل ۵، کپسول‌های لایه بالاتر بردارهای ورودی بسیاری را از لایه پایین دریافت کرده‌اند (تمام نقاط قرمز رنگ و سبز رنگ). وجود نقاط در کنار هم به این معنا است که پیش‌بینی کپسول‌های سطح پایین نزدیک به یکدیگر است (خوشه‌های سبزس رنگ در هر دو کپسول A و B). بنابراین، در این شکل پیش‌بینی کپسول سطح پایین (نقطه زرد رنگ) در کپسول A، دور از خوشه‌ی پیش‌بینی‌های صحیح (یعنی سبز رنگ) قرار گرفته است، اما در کپسول B، نزدیک به خوشه‌ی پیش‌بینی‌های صحیح قرار گرفته است. بنابراین کپسول لایه پایین با کپسول لایه بالای B سازگارتر است و تعلق بیشتری به آن دارد. بنابراین کپسول لایه پایین سعی می‌کند ضریب مسیریابی خود را طوری تنظیم کند که نسبت به کپسول B مقدار بیشتری داشته باشد.




**شکل ۵ الگوریتم مسیریابی پویا، اطلاعات کپسول‌های لایه پایین را به لایه‌های بالاتر ارسال کند. کپسول iام در لایه پایین به کپسولی از لایه‌ی بالاتر تعلق می‌گیرد که بیشترین شباهت را با آن داشته باشد، یعنی مثلا در اینجا به دلیل نزدیک بودن پیش‌بینی کپسول لایه پایین به کپسول B در لایه‌ی بالا، ضریب C_iB افزایش می‌یابد [۵].**

تجربه نشان داده که اگر الگوریتم مسیریابی پویا تنها یک‌بار انجام شود، خروجی برگشت شده از تابع squash مناسب نخواهد بود، چون این خروجی‌ها در یک مرحله رأی دهی همگرا نمی‌شوند. بنابراین ضرایب مشارکت دوباره به روزرسانی می‌شوند، به این شکل که رأی‌های نرمال شده را در بردارهای پیش‌بینی مربوطه ضرب می‌کنیم و با ضریب مشارکت قبلی جمع می‌کنیم. یعنی که کپسول‌ها پس از یک دور رأی دادن از رأی‌های بقیه آگاه می‌شوند و دوباره رأی‌های خود را تغییر می‌دهند. همان‌طور که در [۱] ذکر شده است، پس از سه دور اجرای این الگوریتم رأی‌ها همگرا شده و اجرای بیشتر این الگوریتم تأثیری در خروجی ندارد.

پس از تکمیل مرحله آموزش و مسیریابی در بخش رمزنگار، به ازای هر ورودی در شبکه، در تمامی کپسول‌های خروجی (دسته‌های موجود در داده‌ها)، بردار خروجی تولید می‌شود که انتظار می‌رود که بردار متناظر به دسته‌ی صحیح، بزرگترین طول را داشته باشد. بخش بعد به بررسی این موضوع می‌پردازد.

## ۳-۳ بررسی خروجی شبکه با استفاده از تابع Margin Loss

همان‌طور که در بخش پیش شرح داده شد، در لایه کپسولی-عددی به تعداد دسته‌های موجود در داده‌های ورودی، کپسول خروجی وجود دارد که هر یک از این کپسول‌ها برداری را متناظر با ورودی اعمال شده به شبکه تولید می‌کنند. قاعدتاً طول بردار تولید شده توسط هر کپسول نشان‌دهنده‌ی میزان تعلق داده‌ی ورودی به دسته متناظر با آن کپسول است. در [۱]، یک تابع هزینه به صورت رابطه ۴ (تابع Margin Loss) پیشنهاد شده است که مقدار آن برای همه بردارهای خروجی محاسبه می‌شود.

$$L_k = \underbrace{T_k \max(0, m^+ - \|v_k\|)^2}_{\text{Class present}} + \underbrace{\lambda\,(1 - T_k)\,\max(0, \|v_k\| - m^-)^2}_{\text{Class not present}} \qquad (۴)$$

$$T_k = \begin{cases} 1 & \text{digit of class k present} \\ 0 & \text{other} \end{cases}$$

$$m^+ = 0.9\,,\, m^- = 0.1\,,\, \lambda = 0.5$$

در این رابطه، $T_k$ یک ضریب است که مقدار آن برای دسته صحیح در خروجی برابر با یک و برای بقیه دسته‌ها برابر با صفر است و $\|v_k\|$ اندازه بردار خروجی از کپسول مربوط به دسته‌ی k ام است. تابع هزینه Margin Loss شامل سه پارامتر $m^+, m^-, \lambda$ است. $m^+$ مرز بالا این تابع هزینه است و $m^-$ مرز پایین آن است که در [۱] به ترتیب به ۰٫۹ و ۰٫۱ مقداردهی شده‌اند. $\lambda$ ضریب یادگیری اولیه با مقدار پیش فرض ۰٫۵ برای پایداری[1] مدل است.

بر اساس این تابع، مقدار این تابع هزینه برای خروجی هر کپسول محاسبه می‌شود. برای مثال اگر کپسول k همان دسته صحیح باشد ($T_k = 1$) و اندازه بردار خروجی کپسول k ام عددی بزرگتر از ۰٫۹ و در بقیه کپسول‌ها عددی کمتر از ۰٫۱ باشد، مقدار $\max(0, m^+ - \|v_k\|)^2$ و $(1 - T_k)\max(0, \|v_k\| - m^-)^2$ برابر با صفر خواهد شد و مقدار نهایی تابع هزینه صفر می‌شود، یعنی شبکه این ورودی را به درستی تشخیص داده است. اما اگر اندازه بردار خروجی در کپسول صحیح کمتر از ۰٫۹ باشد، $T_k \max(0, m^+ - \|v_k\|)^2$ برابر با صفر نخواهد بود و در نتیجه تابع هزینه مقداری غیر صفر خواهد داشت که نشان دهنده‌ی تشخیص اشتباه در شبکه است.

## ۴ کارهای انجام شده

همان‌طور که پیشتر ذکر شد، ایده‌ی شبکه‌های کپسولی یک ایده جدید در دنیای شبکه‌های عمیق است که سعی در رفع نواقص مربوط به شبکه‌های کانولوشنی دارد. هر چند CapsNet در مجموعه‌داده‌های ساده مانند MNIST به بالاترین دقت و عملکرد نسبت به دیگر شبکه‌های عمیق رسیده است [۱]، اما هنوز تلاش‌های بسیاری برای بررسی عملکرد این شبکه بر روی داده‌های پیچیده‌تر در حال انجام است (مانند [۴] و [۵]). یکی از دلایل این موضوع نحوه استخراج اطلاعات از تصاویر در کپسول‌ها و میزان زیاد آنها است که باعث پیچیدگی شبکه و مشکلات آموزش آن می‌شود. در نتیجه، شبکه‌های کپسولی هنوز در مرحله تحقیق و توسعه هستند و نتایج اثبات شده بسیار کمی در مورد آنها وجود دارد و این لزوم تحقیقات بیشتر برای شناسایی بهتر این شبکه‌ها را تأیید می‌کند.

در این مقاله، هدف بررسی شبکه کپسولی بر روی مجموعه‌داده‌ی سرطان پوست ISIC منتشر شده در سال ۲۰۲۰[2] است که مجموعه‌داده بسیار پیچیده و مشکل است. در دنیای پزشکی، یکی از شایع‌ترین سرطان‌ها، سرطان پوست است، مخصوصاً سرطان نوع ملانوم که ۷۵٪ مرگ‌های ناشی از سرطان پوست به دلیل ابتلا به آن اتفاق می‌افتد. در نتیجه تشخیص به موقع و دقیق این بیماری می‌تواند در درمان این افراد بسیار مؤثر باشد.

وجود برخی عوامل در این مجموعه‌داده باعث می‌شود که تشخیص بیماری در آن کاری دشوار محسوب شود. برای مثال می‌توان به وجود شباهت در رنگ، شکل و بافت ضایعات خوش‌خیم و بدخیم، کیفیت متفاوت در تصاویر و یا تصاویری که دارای مو هستند، اشاره کرد. اما یکی از چالش‌های اساسی در این مجموعه‌داده که کار آموزش شبکه را بسیار دشوار می‌کند، نامتعادل بودن داده‌ها بین دسته‌های مختلف در این مجموعه داده است. در واقع، تمرکز اصلی این مقاله بر بررسی رفتار و عملکرد شبکه

[1] Stability

[2] https://www.kaggle.com/c/siim-isic-melanoma- classification

کپسولی در مواجهه با داده‌های نامتوازن و چگونگی بهبود شبکه قرار گرفته است.

در گام اول، تغییراتی در لایه‌ها و پارامترهای شبکه داده شد تا شبکه سازگاری بیشتری با مجموعه‌داده‌ی ذکر شده داشته باشد. به این منظور، دو لایه کانولوشنی قبل از لایه‌ی کپسولی-اصلی اضافه شد تا شبکه بتواند ویژگی‌های بیشتری را تشخیص دهد، سپس تغییرات به دو روش اعمال شد:

- **روش اول (Capsnet_v1)**
  - لایه کانولوشن اول با هسته‌ی ۲۵*۲۵، گام با اندازه ۳ و خروجی با اندازه ۱۲۸
  - لایه کانولوشن دوم با هسته‌ی ۱۶*۱۶، گام با اندازه ۲ و خروجی با اندازه ۱۲۸
  - لایه کپسولی-اصلی با هسته‌ی ۹*۹، گام با اندازه ۱ و خروجی با اندازه ۱۲۸

- **روش دوم (Capsnet_v2)**
  - لایه کانولوشن اول با هسته‌ی ۳۶*۳۶، گام با اندازه ۲ و اندازه خروجی ۶۴
  - لایه کانولوشن دوم با هسته‌ی ۱۶*۱۶، گام با اندازه ۲ و اندازه خروجی ۱۲۸
  - لایه کپسولی-اصلی با هسته‌ی ۹*۹، گام با اندازه ۱ و خروجی با اندازه ۲۵۶

پس از انجام بررسی‌های مختلف بر روی شبکه کپسولی و بر اساس عملکرد این شبکه بر روی این مجموعه‌داده‌ی پیچیده، این نتیجه به دست آمد که تابع هزینه[1] شبکه به دلیل عدم توازن در داده‌های دسته‌های مختلف، باید تغییر کند و قاعدتاً باید از توابعی استفاده شود که برای داده‌های نامتوازن تعریف شده‌اند. البته در استفاده از این توابع، فرض بر این بوده است که دسته‌بندی تصاویر در این شبکه قرار است به صورت باینری و در دو دسته ضایعات خوش‌خیم و بدخیم انجام می‌شود.

### تابع هزینه Binary Cross Entropy

یکی از توابع هزینه‌ای که می‌تواند در مورد داده‌های باینری استفاده شود، تابع Binary Cross Entropy است. این تابع هر یک از احتمالات پیش بینی شده را با خروجی دسته واقعی ($y_i$) مقایسه می‌کند که می‌تواند ۰ یا ۱ باشد. این تابع احتمالات را بر اساس فاصله از مقدار مورد انتظار، محاسبه می‌کند و بدان معنی است که به چه میزان از مقدار واقعی، نزدیک یا دور هستیم. در این تابع که به صورت رابطه ۵ تعریف می‌شود

$$H(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i)) \quad (۵)$$

$y_i$ نشان دهنده‌ی برچسب دسته‌ی صحیح برای داده‌ی iام است و $p$ نشان دهنده احتمال تخمین زده شده برای تعلق داده به دسته‌ی صحیح توسط شبکه است. مقدار تابع باید به ازای تمامی نمونه‌ها n=1,2,……,N محاسبه و از حاصل مجموع آن‌ها میانگین گرفته شود.

### تابع هزینه Focal loss

چون طبقه‌بندی دو دسته‌ای است و داده‌های این مجموعه‌داده نیز به صورت نامتعادل هستند، به عنوان یک ایده، جایگزینی تابع هزینه قسمت رمزنگار شبکه با تابع Focal loss طبق [۲۲] مورد بررسی قرار گرفته است. در واقع تابع Focal loss همان نسخه‌ی بهبود یافته از تابع Cross Entropy است که ضریب $(1-p_t)^{\gamma}$ به آن اضافه شده است. در هنگام آموزش شبکه، این تابع به همه‌ی دسته‌بندی‌های نادرست، وزن بیشتری نسبت به دسته‌بندی‌های صحیح می‌دهد، بنابراین سعی می‌کند مشکل عدم تعادل در داده‌ها را بررسی کند. تعریف این تابع در رابطه ۶ آمده است:

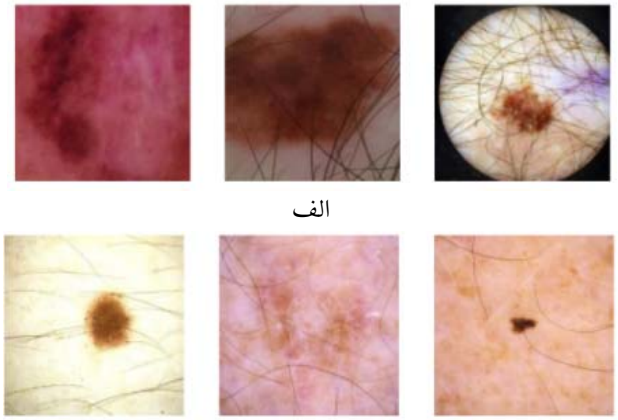
$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (۶)$$

$$p_t \in [0,1], \alpha_t \in [0,1], \gamma \geq 0$$

که در آن، $p_t$ نشان دهنده‌ی احتمال تخمین زده شده توسط مدل برای تعلق داده به دسته‌ی صحیح است و مقداری بین صفر و یک دارد. در نتیجه برای داده‌هایی که با احتمال بالایی درست تخمین زده شده‌اند ($p_t > 0.5$)، ضریب $(1-p_t)^{\gamma}$ عدد کوچکی است و مقدار تابع هزینه کاهش می‌یابد. اما در مورد داده‌هایی که دسته‌ی درست آنها با احتمال پایینی تخمین زده شده است، این ضریب بزرگ‌تر است و هزینه این داده‌ها در شبکه افزایش می‌یابد. پارامتر $\gamma$ هم برای کنترل انحنا و شیب تابع هزینه است (معمولاً مقداری بین صفر و پنج دارد) و هر چه کم‌تر باشد، تمرکز شبکه بر روی داده‌هایی که به درستی دسته‌بندی نشده‌اند، بیشتر خواهد بود. برای برطرف کردن مشکل عدم تعادل داده‌ها بین دو دسته نیز یک راه متداول استفاده از یک ضریب وزنی است. $\alpha$ یک ضریب وزنی است که مقداری بین صفر و یک دارد و برای دسته‌ای که تعداد داده‌ی کم‌تری دارد، بالاتر در نظر گرفته می‌شود تا خطای حاصل از دسته‌بندی اشتباه در دسته‌ی کوچک اهمیت بیشتری پیدا کند. طبق [۲۲] و بر اساس آزمایش‌هایی که صورت گرفته است، بهترین مقدار پارمترهای $\alpha = 0.25$ و $\gamma = 2$ است.

در قسمت بعد و پس از معرفی مجموعه‌داده‌ی استفاده شده به صورت کامل، در مورد عملکرد شبکه با استفاده از توابع هزینه‌ی ذکر شده بحث خواهد شد.

[1] Loss function



الف

ب

**شکل ۶ نمونه ای از تصاویر مجموعه داده‌ی ISIC2020 . الف) تصاویر داده‌های ملانوما (ضایعات بدخیم)، ب) تصاویر داده‌های ضایعات خوش خیم.**

## ۵ مجموعه‌داده و نتایج پیاده‌سازی

### ۵-۱ مجموعه‌داده‌ی ISIC

مجموعه‌داده‌ی ISIC 2020 دارای ۳۳۱۲۶ داده‌ی آموزشی و ۱۰۹۸۲ داده برای تست است که دسته‌ی قطعی داده‌های تست، تاکنون به صورت عمومی منتشر نشده است. شکل ۶ تعدادی از نمونه‌های این مجموعه‌داده را نشان می‌دهد. داده‌های آموزش این مجموعه‌داده نامتعادل بوده و از کل داده‌های آموزشی این مجموعه‌داده، فقط ۵۸۴ داده (کمتر از یک درصد) ملانوما یا ضایعه بدخیم هستند و بقیه خوش‌خیم محسوب می‌شوند. از طرفی هر بیمار ممکن است چندین بار آزمایش داده باشد (داده‌های منحصر بفرد ۶۰۰۰ داده‌ی آموزشی و ۴۰۰۰ داده‌ی تست است).

برای کاهش مشکل عدم توازن در دسته‌های مختلف در داده و بهبود نحوه محاسبه تابع هزینه در زمان آموزش در شبکه، دو راه‌حل مورد بررسی قرار گرفته است:

۱. افزایش تعداد نمونه‌ها در دسته‌ی اقلیت و توازن در توزیع دسته‌ها: می‌توان از داده‌های موجود در نسخه‌های قبلی این مجموعه داده هم استفاده کرد. در اینجا ما از داده‌های سال ۲۰۱۹[1] استفاده کردیم. این مجموعه‌داده شامل ۲۵۳۳۱ تصویر برای مرحله آموزش است که از این تعداد ۴۵۲۲ تصویر مربوط به ضایعات بدخیم هستند.

۲. تعریف وزن برای دسته‌ها در تابع هزینه به صورتی که اگر کلاس اقلیت دچار پیش بینی اشتباه شد، وزن دسته اقلیت در محاسبه تابع هزینه بیشتر باشد.

پس در نهایت از ترکیب دو مجموعه‌داده‌ی مربوط به سال‌های ۲۰۲۰ و ۲۰۱۹، ۵۸۴۵۷ تصویر برای آموزش شبکه استفاده شد که از این تعداد ۵۱۰۶ تصویر (کمتر از ۱۰ درصد) متعلق به دسته ضایعات بدخیم و بقیه تصاویر ضایعات خوش‌خیم بودند.

البته برای ارزیابی شبکه در حین آموزش، تعداد ۳۰۰۰ تصویر به صورت تصادفی به عنوان تصاویر اعتبارسنجی انتخاب شدند و بقیه برای آموزش شبکه مورد استفاده قرار گرفتند. برای مرحله تست شبکه از همان ۱۰۹۸۲ تصویر تست موجود در مجموعه‌داده‌ی سال ۲۰۲۰ استفاده شد.

هم‌چنین به دلیل نامتوازن بودن مجموعه‌داده و کم بودن تعداد نمونه داده‌های ضایعات بدخیم یا ملانوما، با استفاده از روش داده‌سازی[2] تصاویری به وسیله چرخش تصادفی به اندازه ۹۰ یا ۱۸۰ درجه در جهات مختلف تولید شدند و برای جلوگیری از بیش برازش[3] در شبکه، جایگزین تصاویر قبلی شدند.

از طرف دیگر، این مجموعه‌داده دارای یک سری متادیتا است که استفاده از متادیتاها هم می‌تواند اطلاعات مفیدی از داده را بدهد و در بهبود خروجی شبکه مؤثر باشد. در این داده‌ها، اطلاعاتی همچون سن، جنسیت، محل ضایعه ( دست، سر، گوش و....) و نوع ضایعه مشخص شده است. با توجه به اینکه بسیاری از الگوریتم‌های یادگیری ماشین نمی‌توانند مستقیماً روی این جنس از داده ها کار کنند، از روش رمزگذاری one hot برای اضافه کردن این اطلاعات به داده های اصلی استفاده شده است. این اطلاعات از متادیتا از دو لایه تماما متصل با ابعاد ۵۱۲ و ۱۲۸ عبور می‌کنند و به شبکه کپسولی متصل می‌شوند تا خروجی شبکه مشخص شود.

چون سایز عکس‌های مجموعه‌داده‌ی سرطان پوست متفاوت و بسیار بزرگ (۱۰۲۴*۱۰۲۴) است و عملاً امکان پیاده‌سازی شبکه کپسولی بر روی تصاویر با چنین سایزی وجود ندارد، به عنوان پیش پردازش، تصاویر به سایز ۱۲۸*۱۲۸ تغییر سایز داده شدند. اعمال شبکه بر روی سایزهای کوچک‌تر (مثلاً ۲۸ *۲۸) باعث از بین رفتن ویژگی‌های مهم تصویر می‌شود.

### ۵-۲ نتایج پیاده‌سازی

تمامی اجراها و بررسی‌های بر روی سایت kaggle[4] و با استفاده از سخت‌افزاری با مشخصات TPU v3-8 و با NVIDIA TESLA P100 GPU انجام شده است. تعداد دوره‌های هر اجرا به صورت پیش فرض ۵۰ دور در نظر گرفته شده است. به صورت تجربی این نتیجه به دست آمد که کمتر یا بیشتر کردن تعداد دورها تأثیر قابل توجهی در نتیجه ندارد.

ابتدا به بررسی اثر توابع هزینه بر عملکرد شبکه کپسولی بر روی مجموعه‌ی‌داده ISIC می‌پردازیم. نمودارهای شکل ۷ نشان دهنده‌ی میزان دقت شبکه کپسول اصلی (ذکر شده در [۱]) با تابع هزینه‌های متفاوت بر روی داده‌های آموزشی و اعتبارسنجی است. همان‌طور که در این شکل مشخص است، ظاهراً در مرحله‌ی آموزش شبکه، دقت دو تابع Focal Loss و Margin Loss شبیه به هم و دقت تابع BinaryCrossEntropy از آنها دیگر کم‌تر است.

[1] https://challenge.isic-archive.com/data

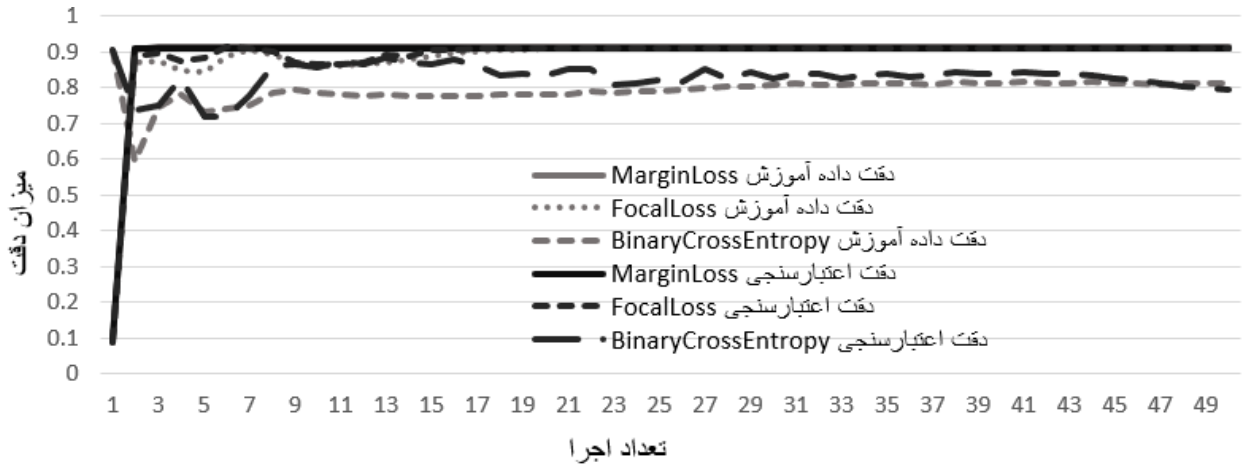
[2] Augmentation
[3] Overfitting
[4] https://www.kaggle.com/docs/tpu

**جدول ۱ نتایج تغییر تابع هزینه‌ی شبکه کپسولی اصلی**

| تابع هزینه | دقت داده‌های آموزش % | دقت داده‌های اعتبارسنجی % | دقت داده‌های تست در سایت kaggle % | دقت داده‌های تست در سایت kaggleبا متادیتا % |
|---|---|---|---|---|
| Margin Loss | ۹۱,۲۲ | ۹۱,۱۵ | ۷۲,۱۴ | ۷۵,۳۷ |
| Focal Loss | ۹۱,۲۴ | ۹۱,۱۷ | ۸۳,۳۱ | ۸۹,۷۹ |
| Binary Cross Entropy | ۸۱,۰۸ | ۷۹,۵۸ | ۸۴,۹۶ | **۹۲,۱۳** |

**جدول ۲ نتایج تغییر تابع فعال‌سازی لایه کانولوشن و لایه کپسولی-اصلی در بخش رمزنگار شبکه کپسولی و استفاده از تابع هزینه‌ی BinaryCrossEntropy**

| تابع فعالسازی | دقت داده‌های آموزش % | دقت داده‌های اعتبارسنجی % | دقت داده‌های تست در سایت kaggle % | دقت داده‌های تست در سایت kaggleبا متادیتا % |
|---|---|---|---|---|
| Tanh | ۸۳,۷۶ | ۸۲,۳۲ | ۸۱,۲۱ | ۹۱,۱۷ |
| Selu | ۸۶,۷۲ | ۸۳,۲۶ | ۷۸,۷۸ | ۸۰,۰۱ |
| Swish | ۸۴,۹۰ | ۸۱,۹۰ | ۸۳,۷۱ | ۹۱,۶۱ |
| Relu | ۸۱,۰۸ | ۷۹,۵۸ | **۸۴,۹۶** | **۹۲,۱۳** |

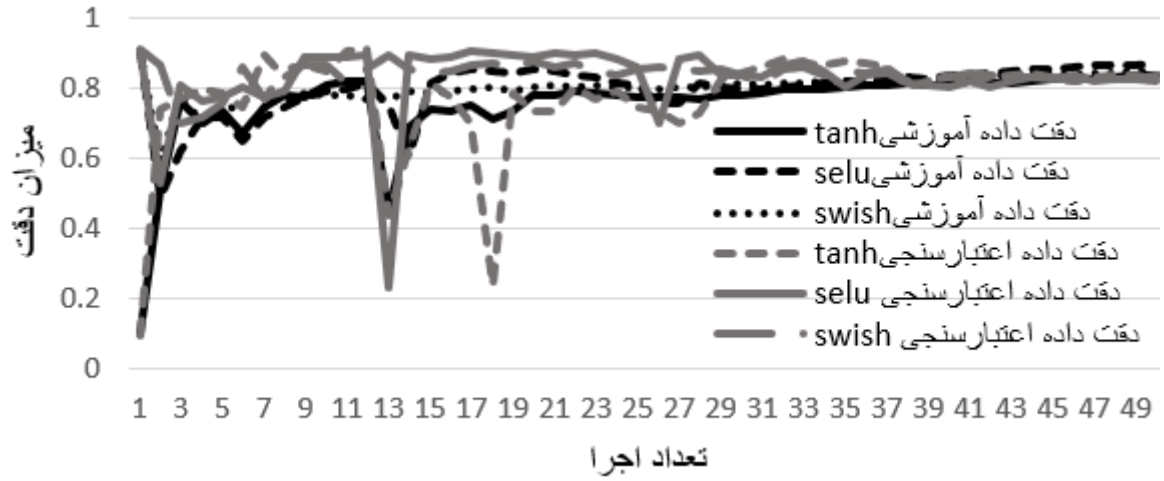



**شکل ۷ دقت شبکه کپسولی بر روی داده‌های آموزش و اعتبارسنجی با توابع هزینه‌ی مختلف(Margin Loss، Focal Loss و Binary Cross Entropy).**

اما با توجه به جدول ۱، دقت شبکه‌ی آموزش دیده با استفاده از تابع BinaryCrossEntropy روی داده‌های تست از دو تابع دیگر بیشتر است و بر روی ۱۰۹۸۲ داده‌ی تست طبق جدول ۱ به ۹۲,۱۳ رسیده است. در واقع تابع هزینه‌ی MarginLoss صرفاً سعی می‌کند شبکه را به نحوی آموزش دهد که احتمال داده‌های درست و نادرست در آن به ترتیب بیشتر از ۰,۹ و کمتر از ۰,۱ باشد و به دلیل استفاده از تابع max، با تمام داده‌هایی که احتمال تعلق‌شان به یک دسته بالاتر از ۰,۹ یا پایین‌تر از ۰,۱ باشد رفتار یکسانی دارد. در حالی که در تابع BinaryCrossEntropy میزان دقیق احتمال تعلق هر داده به هر کلاس تأثیرگذار است و به همین دلیل میزان هزینه در آن به صورت دقیق‌تری محاسبه می‌شود. از طرف دیگر به دلیل اینکه در تابع MarginLoss هزینه در داخل هر batch به صورت جداگانه محاسبه می‌شود، احتمال بیش‌برازش شبکه در آن بیشتر است و به نظر می‌رسد که عملکرد ضعیف‌تر آن نسبت به تابع BinaryCrossEntropy بر روی داده‌های تست به همین دلیل است [۲۳].

همچنین نتایج در جدول ۲ نشان می‌دهد که بر روی داده‌های تست، استفاده از تابع ReLU به عنوان تابع فعال‌سازی (که در [۱] مطرح شده است) بهترین نتیجه را دارد. در واقع میزان دقت شبکه با توابع فعال‌سازی Tanh، SELU، ReLU، SWISH[1] بررسی شد که نتایج آن هم بر روی داده‌های آموزشی و هم بر روی داده‌های اعتبارسنجی، در شکل ۸ نشان داده شده است. از طرفی تابع ReLU ورودی منفی را صفر می‌کند بنظر می‌رسد شبکه با مقادیر مثبت عملکرد بهتری در تشخیص دارد و همان‌طور که از نتایج بر می‌آید تغییر ReLU به توابع فعال‌سازی دیگر روی عملکرد شبکه تأثیر منفی می‌گذارد. در بررسی اثر این توابع فعال‌سازی در شبکه از تابع هزینه BinaryCrossEntropy استفاده شده است.

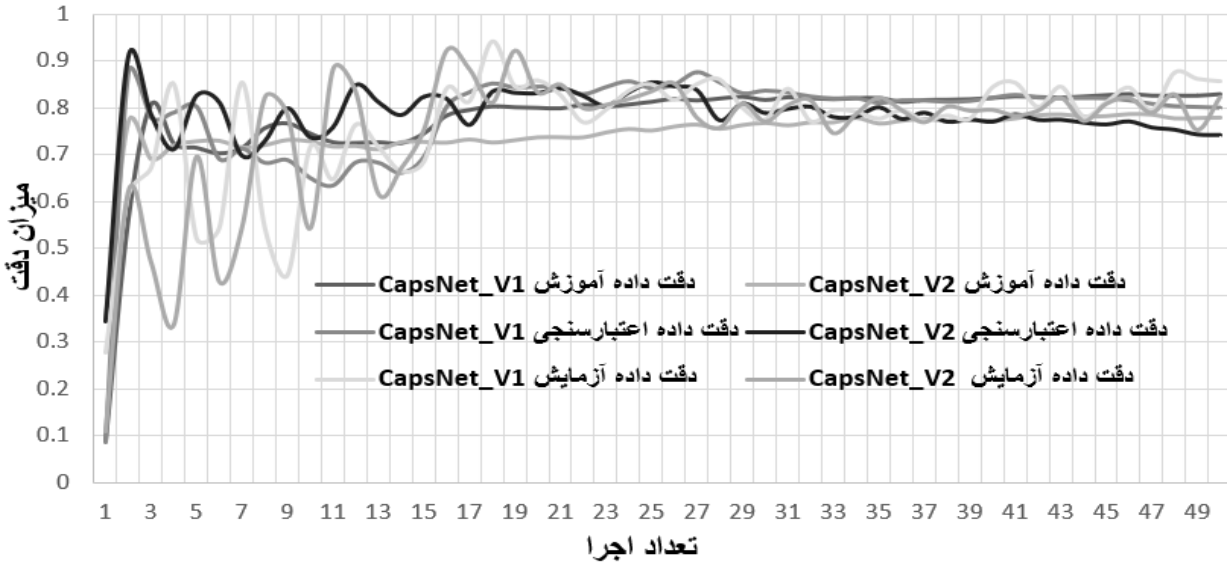
[1] https://www.tensorflow.org/apidocs/python/tf/keras/activations




**شکل ۸ دقت شبکه کپسولی با توابع فعالسازی مختلف ( Tanh ،SELU ،ReLU ، SWISH ) و استفاده از تابع هزینه‌ی Binary Cross Entropy.**

**جدول۳ نتایج تغییر لایه‌های شبکه مطابق با دو روش پیشنهادی و استفاده از تابع هزینه‌ی BinaryCrossEntropy**

| | دقت داده‌های آموزش % | دقت داده‌های اعتبارسنجی % | دقت داده‌های تست در سایت kaggle % | دقت داده‌های تست در سایت kaggleبا متادیتا % |
|---|---|---|---|---|
| روش capsnet_v1 | ۸۳,۰۷ | ۸۰,۱۰ | **۸۴,۳۳** | **۹۲,۲۴** |
| روش capsnet_v2 | ۷۷,۸۹ | ۷۴,۳۰ | ۸۳,۸۶ | ۹۱,۵۱ |




**شکل ۹ دقت دو شبکه کپسولی بهبود یافته‌ی capsnet_v1 و capsnet_v2 بر روی داده های آموزشی، اعتبارسنجی و تست**

پس از بررسی عملکرد شبکه اصلی کپسولی با استفاده از توابع هزینه و فعال‌سازی متفاوت، حال عملکرد این شبکه بر اساس تغییر در تعداد و نوع لایه‌ها بر اساس دو ترکیبی از شبکه که در بخش ۴ ذکر شد، مورد بررسی قرار می‌گیرد. شکل ۹ عملکرد دو شبکه کپسولی تغییر یافته (capsnet_v1 و capsnet_v2) را بر روی داده‌های آموزشی، اعتبارسنجی و تست نشان می‌دهد. همچنین بر اساس نتایج جدول ۳، می‌توان مشاهده کرد که دقت روش capsnet_v1 هم بر روی داده‌های آموزشی و هم بر روی داده‌های تست بالاتر است و ظاهراً این تغییرات در شبکه با داده‌های استفاده شده سازگارتر است.

جدول ۴ خلاصه‌ای از کارها و تحقیقات انجام شده بر روی مجموعه‌داده‌ی سرطان پوست را در مقایسه با یکدیگر نشان می‌دهد. در این جدول که معیارهای ارزیابی متفاوتی در آن لحاظ شده است،

**جدول ۴ مقایسه نتایج کار شبکه کپسولی بهبود یافته با کارهای پیشین**

| | روش | مجموعه داده | تعداد تصاویر | ACC(%) | Recall(%) | SP(%) | AUC(%) | زمان |
|---|---|---|---|---|---|---|---|---|
| [۱۶] | CNN +FNN | ISIC 2017 | ۲۷۵۰ | ۷۶,۰۰ | ۸۲,۰۰ | ۶۲,۰۰ | – | – |
| [۱۷] | FCNN | ISIC 2017 | ۲۱۵۰ | ۷۲,۰۰ | – | – | ۶۵,۸۰ | – |
| [۱۵] | Ensemble NN + SVM | ISIC 2016 | ۱۲۷۹ | ۷۶,۰۰ | ۸۲,۰۰ | ۶۲,۰۰ | – | – |
| | | | | | | | | |
| [۱۸] | CNN + SVM | ISIC 2016 | ۱۲۷۹ | ۸۵,۲۰ | ۵۰,۴۰ | ۹۳,۰۰ | ۷۸,۰۰ | – |
| [۱۹] | CNN + SVM | ISIC 2016 + 2017 | ۲۰۳۷ | – | – | – | ۹۰,۶۹ | – |
| [۲۰] | CNN + CAPS | ISIC 2019 | ۵۵۰۹ | ۹۲,۴۰ | ۹۰,۹۰ | ۹۲,۸۰ | ۸۹,۶۰ | – |
| [۲۱] | 18 Mode of EfficientNet, SE-ResNeXt-101, ResNeSt-101 | ISIC 2018 + ISIC 2019 + ISIC 202 | بیش از ۵۸۰۰۰ داده | **۹۴,۴۲** | – | – | **۹۸,۴۵** | زمان آموزش برای هر شبکه بین ۱۵ تا ۴۵ ساعت متغیر بوده است. |
| شبکه طراحی شده (capsnet_v1) | Modified Capsnet | ISIC 2019 +ISIC 2020 | ۵۸۴۵ ۷ | ۹۲,۲۴ | ۸۵,۰۹ | ۷۵,۳۲ | ۸۲,۴۵ | کمتر از ۲ ساعت |

**جدول ۵ نتایج طبقه‌بندی شبکه کپسولی با تغییر تابع هزینه( Margin Loss , BinaryCrossEntropy) شبکه بر روی داده‌های قطعه‌بندی شده**

| تابع هزینه | دقت داده‌های آموزش % | دقت داده‌های اعتبارسنجی % | دقت داده‌های تست در سایت kaggle % | دقت داده‌های تست در سایت kaggle با متادیتا % |
|---|---|---|---|---|
| Margin Loss | ۹۱,۲۲ | ۹۱,۱۵ | ۷۶,۰۳ | ۸۱,۲۰ |
| BinaryCrossEntropy | ۷۶,۱۴ | ۷۴,۷۴ | ۸۱,۱۲ | **۹۱,۵۳** |

بهترین نتایج توسط مرجع [۲۱] به دست آمده که این مقاله از شبکه‌های از قبل آموزش دیده و از ۱۸ مدل مختلف برای آموزش دادن شبکه‌ی خود استفاده کرده است و به همین دلیل به نتایج بهتری دست یافته است، اما زمان اجرای این شبکه در مقایسه با شبکه کپسولی طراحی شده به طور قابل توجهی بالا می‌باشد، در حالی که دقت این دو مدل تفاوت چندانی با هم ندارد.

بالا بودن زمان آموزش در یک شبکه می‌تواند مشکل بزرگی برای پژوهشگرانی باشد که سخت‌افزار لازم را برای آموزش شبکه‌ها در اختیار ندارند.

آخرین ایده‌ای که برای بهبود عملکرد شبکه کپسولی بر روی مجموعه‌داده‌ی پیچیده‌ی سرطان پوست مورد بررسی قرار گرفته است، استفاده از قطعه‌بندی داده‌ها و تعیین منطقه ضایعه به عنوان پیش پردازش بوده است. چون تصاویر این مجموعه‌داده از نظر بافت و شکل مشابه هستند، قبل از ورود این تصاویر به شبکه اصلی مهم است که بتوانیم ضایعه پوستی را به درستی تشخیص دهیم. در این مرحله، قطعه‌بندی تصاویر با استفاده از شبکه‌های MobilenetV2 و UNet صورت گرفت. هم‌چنین بهینه‌ساز این شبکه Adam و تابع هزینه SparseCategoricalCrossentropy از Keras در نظر گرفته شد[1]. جدول ۵ نتایج به دست آمده از عملکرد شبکه را بر روی داده‌های قطعه‌بندی شده نشان می‌دهد. بر اساس این نتایج به نظر می‌آید که قطعه‌بندی تصاویر بهبودی در عملکرد شبکه ایجاد نمی‌کند و به نظر می‌رسد که خود شبکه در لایه‌های مختلفی که دارد قابلیت تشخیص منطقه ضایعه را داشته است.

در مجموع آزمایشات انجام شده، می‌توان به این نتیجه رسید که اگر چه هنوز تحقیقات کافی بر روی شبکه کپسولی به دلیل جدید بودن و داشتن ساختاری متفاوت با شبکه‌های قبل از خود و زمان‌بر بودن آموزش در آن انجام نشده است و کارآیی این شبکه بر روی مجموعه‌داده‌های پیچیده مورد ابهام است، اما می‌توان با ایجاد تغییرات مناسب در شبکه کپسولی در کاربردهای متفاوتی از این شبکه استفاده کرد.

[1] https://www.tensorflow.org/tutorials/images/seg mentation

## نتیجه‌گیری

شبکه‌های کپسولی به عنوان نوعی خاص از شبکه‌های کانولوشنی مطرح شده‌اند که در آنها از سازماندهی لایه‌ها و نورون‌ها در قالب یک ساختار کپسولی استفاده شده است. با وجود اینکه به نظر می‌رسد که این شبکه‌ها قابلیت بالایی در مدل کردن یک سری تغییرات موجود در داده‌ها دارند، اما به دلیل ساختار متفاوتی که در این شبکه‌ها وجود دارد، آموزش و رفتار آنها در مواجهه با داده‌های پیچیده در ابهام است. در این مقاله، سعی شد که رفتار شبکه کپسولی بر روی مجموعه‌داده‌ی حجیم و پیچیده‌ی سرطان پوست مورد بررسی قرار گیرد. در واقع رفتار شبکه با استفاده از توابع هزینه مختلف و توابع فعال‌سازی متفاوت در شبکه بررسی شد. هم‌چنین اثر اضافه کردن لایه‌های کانولوشنی در ابتدای شبکه به منظور کمک به استخراج ویژگی‌های بهتر از داده‌های پیچیده با نتایج مثبتی روبرو شد. در نهایت می‌توان عنوان کرد که با ایجاد تغییرات مناسب بر روی پارامترهای موجود در این شبکه‌ها به منظور تطابق بیشتر آنها با داده‌های ورودی، می‌توان امید داشت که بتوان از قابلیت‌های این شبکه‌ها بر روی داده‌های مختلف با خصوصیات متفاوت استفاده کرد.

## مراجع

[۱] Sabour, S., Frosst, N., Hinton, G. E. “Dynamic Routing Between Capsules”, Advances in Neural Information Processing Systems (NIPS), pp. 3859–3869, 2017.

[۲] Lee YH. “Efficiency Improvement in a Busy Radiology Practice: Determination of Musculoskeletal Magnetic Resonance Imaging Protocol Using Deep Learning Convolutional Neural Networks”, Journal of Digital Imaging, vol. 31(5), pp. 604-610, 2018.

[۳] Gong. E., Pauly J.M., Wintermark, M., Zaharchuk, G. “Deep Learning Enables Reduced Gadolinium Gose for Contrast-Enhanced Brain MRI”. Journal of Magnetic Resonance Imaging, vol 48(2), pp. 330–340. 2018.

[۴] Afshar, A., Mohammadi, K. N. Plataniotis. “Brain Tumor Type Classification Via Capsule Networks.” 25th IEEE International Conference on Image Processing (ICIP), pp. 3129-3133, 2019.

[۵] Mobiny, A. Van Nguyen, H. “Fast Capsnet for Lung Cancer Screening.” International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 741–749, 2018.

[۶] Afshar, P., Heidarian, SH, “COVID-CAPS: A Capsule Network-Based Framework for Identification of COVID-19 Cases from X-ray Images”. Pattern Recognition Letters, vol. 138, pp. 638-643, 2020.

[۷] Quan, H., Xu, X., “DenseCapsNet: Detection of COVID-19 X-ray Images Using a Capsule Network”, Comput Biol Med, vol. 133, pp.1-11, 2020.

[۸] Dorj, U. O., Lee, K.-K., Choi, J.-Y., Lee, M. J. Applications. “The Skin Cancer Classification Using Deep Convolutional Neural Networks.”, Multimedia Tools and Applications, vol. 77(8), pp. 9909-9924, 2018.

[۹] Esteva, A., et al. “Dermatologist-Level Skin Cancer With Deep Neural Networks.” Nature, vol. 542, pp. 115-118. 2017.

[۱۰] Xu, L., Ren, J. S., Liu, C., Jia, J. “Deep Convolutional Neural Network for Image Deconvolution.” Adv in neural inform processing sys(NIPS), pp. 1790-1798. 2014.

[۱۱] Qayyum, A., Anwar, S. M., Awais, M., Majid, M. “Medical Image Retrieval Using Deep Convolutional Neural Network.” Neurocomputing, vol. 266, pp. 8-20, 2017.

[۱۲] Dorj, U.O., et al. “The Skin Cancer Classification Using Deep Convolutional Neural Networks.” Multimedia Tools and Applications, vol. 77(8), pp. 9909-9924, 2018.

[۱۳] Brinker, T. J., et al. “Skin Cancer Classification Using Convolutional Neural Networks:” systematic review. Journal of medical Internet research, vol. 20(10), 2018.

[۱۴] Rezvantalab, A., Safigholi, H., Karimijeshni. S., “Dermatologist Level Dermoscopy Skin Cancer Classification Using Different Deep Learning Convolutional Neural.” Networks algorithms. ArXiv preprint, 2018.

[۱۵] Codella, Noel CF, et al. “Deep Learning Ensembles for Melanoma Recognition in Dermoscopy Images.” IBM Journal of Research and Development, vol. 61(4/5), 2017.

[۱۶] Mirunalini, P., Chandrabose, A., Gokul, V., Jaisakthi, S. “Deep Learning for Skin Lesion Classification.” arXiv:1703.04364, 2017.

[۱۷] Li, Y., Shen, L. “Skin Lesion Analysis Towards Melanoma Detection Using Deep Learning Network.” Sensors, vol. 18(2), pp. 556. 2018.

[۱۸] Majtner, T., Yildirim-Yayilgan., S. Hardeberg., J. Y. “Combining Deep Learning and Hand-Crafted Features for Skin Lesion Classification”, Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6. 2016.

[۱۹] Mahbod, A., Schaefer, G., Wang, C., Ecker, R., Ellinge, I. “Skin Lesion Classification Using Hybrid Deep Neural Networks”, International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1229–1233. 2019.

[۲۰] Boaro, J. M. C., dos Santos, P. T. C., Rocha, C. V. M., Fontenele, T., Junior, G. B., de Almeida, J. D. S., Rocha, S, “Hybrid Capsule Network Architecture Estimation for Melanoma Detection”, International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 93-98.2020.

[۲۱] Ha, .Q, Liu, .B, Liu, .F. “Identifying Melanoma Images using EfficientNet Ensemble: Winning Solution to the

SIIM-ISIC Melanoma Classification Challenge", CoRR abs/2010.05351 (2020), 2020.
[۲۲] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P., "Focal Loss for Dense Object Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, pp. 318-327, 2020.
[۲۳] Li Z., Kamnitsas K., Glocker B, "Overfitting of Neural Nets Under Class Imbalance: Analysis and Improvements for Segmentation", MICCAI, vol 11766, 2019.

**نرگس حسن‌پور** مدرک کارشناسی خود را در رشته مهندسی کامپیوتر گرایش نرم افزار در دانشگاه شهید باهنر کرمان در سال ۱۳۹۵ دریافت کرد. ایشان هم اکنون دانشجوی کارشناسی ارشد مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک در دانشگاه شهید باهنر کرمان است. زمینه‌های پژوهشی مورد علاقه وی یادگیری‌عمیق، یادگیری‌ماشین، بینایی‌ماشین و پردازش‌تصویر است.

**امید اسلام** مدرک کارشناسی خود را در رشته مهندسی کامپیوتر گرایش نرم افزار در دانشگاه شهید باهنر کرمان در سال ۱۳۹۶ دریافت کرد. ایشان هم اکنون در زمینه‌های تحقیقاتی و پژوهشی مشغول به کار است. زمینه‌های مورد علاقه ایشان داده‌کاوی، یادگیری عمیق، یادگیری ماشین و پردازش تصویر است.

**حدیث محسنی** دانش‌آموخته‌ی دکترای هوش مصنوعی در سال ۱۳۹۲ از دانشگاه صنعتی شریف است. ایشان در حال حاضر استادیار بخش مهندسی کامپیوتر دانشکده فنی دانشگاه شهید باهنر کرمان است که در زمینه‌های پژوهشی یادگیری عمیق و کاربردهای آن در پردازش الگو و پردازش تصویر و سیگنال‌های پزشکی مشغول به فعالیت است.