

Journal of Soft Computing and Information Technology (JSCIT)

Babol Noshirvani University of Technology, Babol, Iran

Journal Homepage: jscit.nit.ac.ir

Volume *, Number *, Spring ٢٠**, pp. **-**

Received: **/**/**, Revised: **/**/**, Accepted: **/**/**



Fully Connected to Fully Convolutional: Road to Yesterday

Mahmood Amintoosi

Department of Computer Science, Hakim Sabzevari University, Sabzevar, Iran.

m.amintoosi@hsu.ac.ir

Abstract- In the last decade, several convolutional networks have been developed for the semantic segmentation, which have shown excellent performance in recognizing and labeling objects in images. Most of these networks involve large-scale architectures that can detect tens or hundreds of predefined classes. With the exception of fully convolutional networks, most applications use architectures that, after convolutional layers, use a common classifier to classify the extracted features. In this paper, the method of converting a network, which as classifier, has two flatten and dense layers (fully connected), to a fully convolutional network is described. The main advantage of this method is the ability to work on inputs of variable size and produce an output map instead of a number, which is the advantage of fully convolutional networks. Newer models of the Deep Learning area generally use training images in which areas of interest are determined by masks; but in the proposed method only labeled images are given to the network. The details of the proposed method are expressed in the form of a new problem of classification of boards with calligraphy of Shekasteh-Nastaliq and Suls, and classification of apple leaf diseases (as two-class problems) and the problem of identifying hand written Persian digits. For this purpose, first a convolutional network with the last fully connected layer is designed and trained for square images. Then a new fully convolutional model is defined based on the previous model and the weights of the previous model are fed to the new model. The only difference between the two models is in the last layer, but the new model will be able to work on input images of any size. Experimental results show the efficiency of the proposed approach.

Keywords- Deep Learning, Convolutional Neural Networks, Image Classification, Object Detection, MultiLayer Perceptron.

تمام متصل به تمام پیچشی: پلی به گذشته

محمود امین طوسی

دانشکده ریاضی و علوم کامپیوتر، دانشگاه حکیم سبزواری

m.amintoosi@hsu.ac.ir

چکیده- در یک دهه‌ی گذشته شبکه‌های پیچشی متعددی برای قطعه‌بندی معنایی تصاویر ابداع شده‌اند که عملکرد بسیار خوبی در تشخیص و برچسب‌زنی اشیاء از خود نشان داده‌اند. عمده‌ی این شبکه‌ها متضمن معماری‌های با اندازه‌ی بزرگ هستند که توانایی آشکارسازی ده‌ها یا صدها دسته‌ی از قبل مشخص را داشته باشند. در بیشتر کاربردها از معماری‌هایی استفاده می‌شود که پس از چند لایه‌ی پیچشی از یک طبقه‌بند معمول برای طبقه‌بندی ویژگی‌های استخراج شده‌ی شبکه استفاده می‌شود. در این نوشتار روش تبدیل یک شبکه که به عنوان طبقه‌بند، دو لایه‌ی مسطح و چگال (تمام متصل) دارد، به یک شبکه تمام پیچشی بیان شده است. مزیت اصلی این شیوه، قابلیت کارکرد بر روی ورودی‌های با اندازه متغیر و تولید یک نقشه خروجی به جای یک عدد می‌باشد که همان مزیت شبکه‌های تمام پیچشی است. در مدل‌های جدید حوزه‌ی یادگیری عمیق عموماً از تصاویر آموزشی که در آنها نواحی موردنظر با ماسک مشخص شده‌اند استفاده می‌شود، اما در شیوه‌ی پیشنهادی در این نوشتار فقط تصاویر برچسب‌دار (مشخص‌کننده طبقه‌ی کل تصویر) به شبکه داده می‌شود. جزییات روش کار در قالب مسئله‌ی جدید طبقه‌بندی و شناسایی تابلوهای با رسم‌الخطهای شکسته نستعلیق و ثلث، شناسایی برگ سالم از مریض سبب (به عنوان مسائل دو کلاسه) و مسئله‌ی شناسایی ارقام فارسی بیان شده است. به این منظور ابتدا یک شبکه پیچشی با لایه آخر تمام متصل طراحی و بر روی تصاویر مربعی آموزش داده می‌شود. سپس مدل تمام پیچشی جدیدی بر اساس مدل قبلی تعریف شده و وزنهای مدل قبلی به مدل جدید کپی می‌شود. تنها تفاوت دو مدل در لایه آخر است، اما مدل جدید قابلیت کار بر روی تصاویر ورودی با هر اندازه را خواهد داشت. نتایج آزمایشات کارایی این شیوه را نشان داده است (کد برنامه در <https://github.com/mamintoosi/FC2FC>).

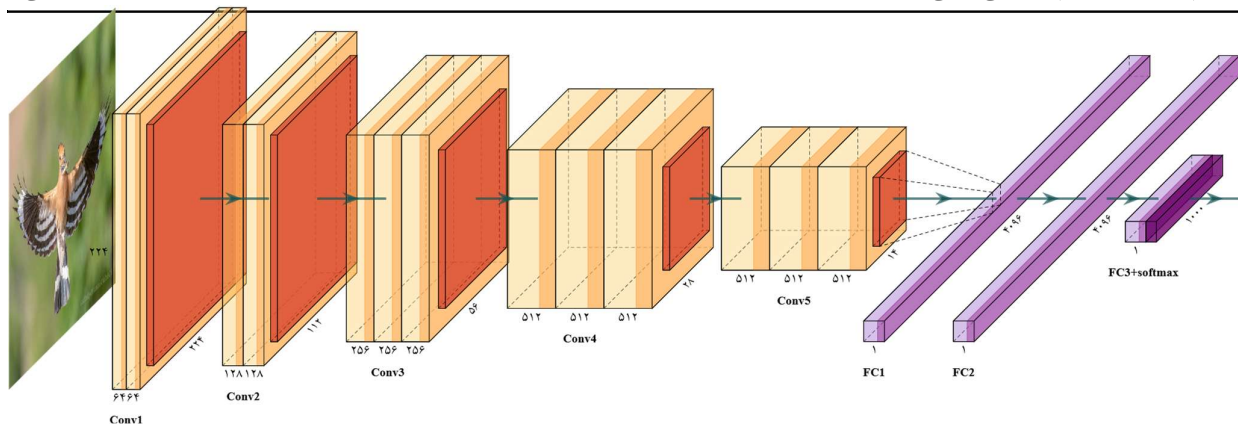
واژه‌های کلیدی: یادگیری عمیق، شبکه‌های عصبی پیچشی، طبقه‌بندی تصویر، شناسایی اشیاء، پرسپترون چند لایه

۱- مقدمه

دارند. این مجموعه داده مربوط به فضاهای داخل می‌باشد. چند نمونه مجموعه داده دیگر در ادامه آمده است:

- مجموعه داده^۵ ImageNet شامل ده میلیون تصویر برچسب زده شده از بیش از ۱۰۰۰ دسته.
- مجموعه داده پاسکال^۶ با بیش از ده هزار تصویر شامل ۲۳۳۷۴ ناحیه علامت زده شده و حجم داده‌های حدود یک و نیم گیگابایت^[۱۲] که شامل طبقه‌های مختلف مانند سگ، گربه و آسمان است. مجموعه داده‌های PASCAL-Context که برچسب‌زنی اضافه‌تری نسبت به برچسب‌های پاسکال دارد و توسط روزبه متقی و دیگران^[۱۳] آماده شده است.
- مجموعه داده میکروسافت COCO^[۱۴] با ۲.۵ میلیون تصویر.

در سالیان اخیر شبکه‌های عصبی پیچشی^۱ کارایی چشمگیری در شناسایی الگوهای پیچیده و آشکارسازی اشیاء به نمایش گذاشته‌اند [۱،۲]. روشهای [۳]FCN، [۴]R-CNN، [۵]Fast R-CNN، [۶]Faster R-CNN، [۷]MaskR-CNN، [۸،۹]YOLO و [۱۰]SegNet از جمله تحقیقات پر ارجاع و کارا در این حوزه هستند. داشتن مجموعه دادگان بزرگ که در آن اشیاء مختلف مورد شناسایی، توسط یک کادر یا ماسک در هر تصویر مشخص شده باشند^۲ از الزامات مدل‌های پیچیده‌ی جدید این حوزه هستند. به عنوان نمونه، مجموعه داده NYU Depth Dataset^۳ [۱۱] که در گیت‌هاب^۴ مربوط به مقاله [۳] به آن اشاره شده، در نسخه خام، ۴۲۸ گیگابایت و داده‌های برچسب زده شده آن ۸.۲ گیگابایت حجم



شکل ۱: معماری مدل VGG16 [۱۵] (چارچوب مدل با استفاده از ابزار سایت [PlotNeuralNet](#) رسم شده است. حق نشر تصویر هدهد متعلق به مجید مؤمنی مقدم عضو هیات علمی گروه زیست شناسی دانشگاه حکیم سبزواری است.)

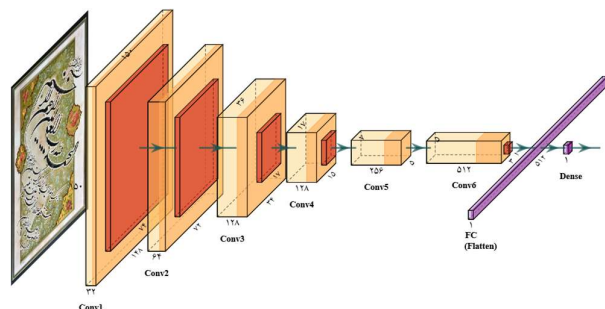
یکسان قابلیت کار دارند، به عنوان مثال اگر بردار ویژگی حاصل از آخرین لایه پیچشی یک بردار هزارتایی باشد، ورودی طبقه‌بندی، هزارتایی خواهد بود. اگر به واسطه‌ی تغییر اندازه‌ی تصویر ورودی به شبکه پیچشی، اندازه این بردار تغییر کند، طبقه‌بندی قادر به عمل نخواهد بود. هدف اصلی این نوشتار بیان نحوه جایگزینی لایه آخر کاملاً متصل^{۱۲} با یک لایه پیچشی است، به نحوی که قابلیت عمل بر روی تصاویر با اندازه متغیر فراهم گردد. ایده جایگزینی لایه آخر تمام متصل با لایه‌ی پیچشی در FCN [۲] بکار برده شده است. بخش قابلیت عملکرد بر روی تصاویر با اندازه متغیر هم در MaskRCNN [۷] مورد استفاده قرار گرفته است. در این نوشتار نحوه تلفیق این دو و ایجاد مدل‌های تمام پیچشی با اندازه‌ی کوچک به تفصیل بیان شده و برنامه‌های منبع باز این نوشتار روش کار به صورت عملی را روشن خواهند کرد. در این شیوه نیازی به تصاویر آموزشی که در آن نمونه‌های مورد طبقه‌بندی با ماسک مشخص شوند، نیست. گرچه که شیوه مدنظر ربطی به اندازه مدل ندارد، اما در کاربردهای مورد بررسی، مدل‌هایی کوچک طراحی شده است که آموزش آنها به سرعت قابل انجام باشد، قابلیت‌های مدنظر در نوشتار را داشته و موضوع قابل درک‌تر باشد.

به نظر می‌رسد عواملی همچون حجم زیاد مدل‌های از قبل آموزش دیده در حوزه یادگیری عمیق، معماری‌های متعدد، پلتفرم‌های مختلف و پارامترهای فراوان، شاید موجب شده باشد که حرکت از روش‌های معمول طبقه‌بندی به شیوه‌های یادگیری عمیق سخت به نظر برسد. آشنایی با شیوه‌ی مورد اشاره در این نوشتار موجب خواهد شد که بسیاری از راه حل‌ها و مجموعه داده‌گان قبلی آماده شده توسط محققین، به راحتی در بستر یادگیری عمیق، قابل استفاده شده و قابلیت عمل با تصاویر ورودی با اندازه متغیر را هم داشته باشند. برای کاربردهای فراوانی، مجموعه داده‌های آموزشی توسط محققین آماده شده است، اما قابلیت استفاده در شیوه‌های

این مجموعه داده‌های بزرگ عموماً در ایجاد معماری‌هایی مانند VGG [۱۵]، AlexNet [۱۶] و DarkNet^۱ [۹] بکاربرده شده‌اند؛ که مبنای بیشتر مدل‌های مرتبط است. به این مدل‌ها، مدل از قبل آموزش دیده اطلاق می‌شود. شکل ۱ ساختار مدل VGG16 را نشان می‌دهد که بالغ بر ۱۳۰ میلیون پارامتر دارد. در [۱۷] از مدل AlexNet برای تصدیق هویت مبتنی بر اثر انگشت و تصویر چهره فرد استفاده شده است. داشتن حدود ۶۰ میلیون پارامتر در مدل مورد استفاده به عنوان یک نقطه ضعف مدل برشمرده شده است. ایجاد مجموعه داده‌های آموزشی که اشیاء مدنظر در هر تصویر با یک کادر یا ماسک برچسب‌گذاری شوند، مستلزم هزینه و زمان‌بر است [۱۸]. ساده‌تر کردن تهیه داده‌های آموزشی با عنوان شناسایی اشیاء با نظارت مختصر^{۱۰} یکی از زیرمسائل شبکه‌های پیچشی در بینایی ماشین محسوب می‌شود.

نسخه‌های مختلف R-CNN و YOLO همگی مبتنی بر یک مدل از قبل آموزش دیده هستند. اندازه این مدل‌ها معمولاً بسیار بزرگ بوده و شامل میلیون‌ها پارامتر هستند. برای آموزش چنین مدل‌هایی حجم زیاد داده و توان پردازشی بالا مورد نیاز است. لذا در بسیاری از تحقیقات بعدی این حوزه، تغییرات مختصری در لایه‌های آخر این مدل‌ها برای کاربردهای خاص صورت می‌پذیرد. به این ترتیب از مدل از قبل آموزش دیده استفاده شده و نیاز به پردازش‌های سنگین برطرف می‌شود. در غیر اینصورت مدل باید از پایه طراحی و آموزش داده شود^{۱۱}.

در بیشتر مدل‌های ارائه شده با شبکه‌های عصبی پیچشی در یادگیری عمیق که با نام شبکه‌های پیچشی عمیق نیز معروف هستند، در آخرین لایه، یک طبقه‌بندی مانند شبکه‌های عصبی پرسپترون یا ماشین بردار پشتیبان بکار گرفته می‌شود. تعویض این طبقه‌بندی با سایر طبقه‌بندی‌ها از جمله نوآوری‌های مرسوم این حوزه است [۱۹]. عموم طبقه‌بندی‌ها بر روی بردارهای ویژگی با طولی



شکل ۲: یک نمونه معماری مدل پیچشی-چگال که بخش طبقه‌بند آن شامل یک لایه ۵۱۲ تایی مسطح (Flatten) و یک لایه چگال (Dense) است که کاملاً متصل هستند (Fully Connected).

این معماری، ساختار مرسوم شبکه‌های پیچشی عمیق است که در ادامه آنرا مدل «پیچشی-چگال» خواهیم نامید^{۱۸}. هدف اصلی آن است که بخش طبقه‌بند این معماری با یک لایه پیچشی به نحوی تعویض شود که تعداد پارامترهای مدل ثابت بماند. ثابت ماندن تعداد پارامترها - در کنار مفهوم پارامترها - این اجازه را به ما خواهد داد که با معماری مدل پیچشی-چگال سیستم آموزش داده شده و سپس پارامترهای این مدل در یک مدل تمام پیچشی^{۱۹} معادل، کپی شود. طراحی و آموزش مدل پیچشی-چگال ساده است و قابلیت کار بر روی بسیاری از مجموعه داده‌گان تصویری را دارد. در ادامه در ابتدا نحوه انجام این کار و سپس نحوه اصلاح ساختار مدل برای پذیرش تصاویر با اندازه‌های متنوع بیان خواهد شد.

۲-۱- تبدیل مدل پیچشی-چگال به مدل تمام پیچشی

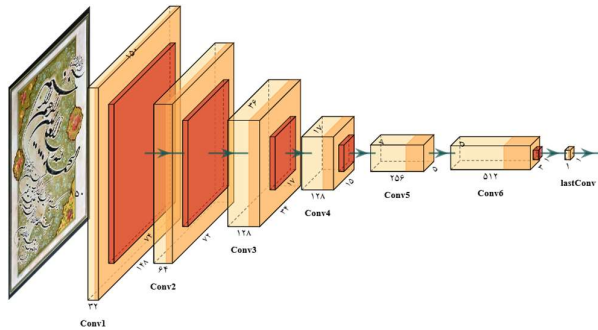
مدل پیچشی-چگال شکل ۲ را در نظر بگیرید. مدل مورد نظر از ۶ لایه پیچشی (Conv1,...,Conv6)، یک لایه کاملاً متصل ۱۲۵ تایی و یک لایه آخر چگال با یک نورون تشکیل شده است. که قابلیت استفاده برای مسائل طبقه‌بندی دو کلاس را دارد. تابع فعالیت هر لایه پیچشی، واحد خطی اصلاح شده^{۲۰} در نظر گرفته شده که با رنگ نارنجی در هر بلوک پیچشی مشخص شده است. لایه های نمونه‌بردار (یا انتخاب بیشینه یا ادغام) با رنگ قرمز و لایه‌های مسطح تمام متصل (FC) و چگال با رنگ بنفش مشخص شده‌اند. تابع فعالیت آخرین لایه، یک تابع سیگموئید است که با رنگ بنفش پررنگ نمایش داده شده است. ورودی مدل یک تصویر مربع ۱۵۰×۱۵۰ است که بعد از گذر از اولین لایه پیچشی با گام یک و اندازه صافی (فیلتر) ۳×۳ به اندازه ۱۴۸×۱۴۸ می‌رسد. اعداد بالا و پایین لایه‌های پیچشی بیانگر اندازه‌های ورودی و خروجی می‌باشند.

جدید همچون Mask RCNN را ندارند. به عنوان مثال اگر مجموعه دادگانی برای تفکیک دو طبقه در حوزه‌ی پردازش تصویر موجود باشد، قاعدتاً تعدادی تصویر از هر دو دسته برای عمل آموزش فراهم شده است (مانند مجموعه تصاویر سگ و گربه در منابع آموزشی یادگیری عمیق)؛ در روش‌هایی مانند Mask RCNN باید دور سوژه‌ی مربوطه در تصاویر آموزشی را با چند ضلعی^{۱۳} یا با یک ماسک مشخص نمود. اما در شیوه‌ی پیشنهادی، تصاویر آموزشی در قالب مرسوم پیشین (که هر تصویر یک برجسب مشخص دارد)، قابلیت استفاده را خواهند داشت. همچنین معماری مدل به گونه‌ای اصلاح خواهد شد که قابلیت عمل بر روی تصاویر ورودی با هر اندازه را داشته و نقشه‌ای از میزان تعلق هر بخش تصویر به دسته‌های مورد عمل را به دست دهد؛ به جای اینکه تصمیم‌گیری بر روی کل تصویر باشد و طبقه‌ی کل تصویر را مشخص کند.

به عنوان کاربردهایی از شیوه‌ی پیشنهادی، مسائل شناسایی دو رسم الخط مشهور **سکه شقیق و ثلث** در تابلو نگاره‌ها، تفکیک برگ‌های سالم و مریض درخت سیب و شناسایی ارقام فارسی در نظر گرفته شده است که در بخش ۳ به تفصیل بیان خواهند شد.

۲-۲ شیوه‌ی پیشنهادی

همان گونه که در بخش قبل اشاره شد، بسیاری از مدل‌های پیچشی عمیق متضمن چند لایه پیچشی و در انتها، لایه(هایی) تمام متصل برای طبقه‌بندی می‌باشند. شکل ۱ معماری مدل VGG16 را نشان می‌دهد که همانگونه که ملاحظه می‌شود در آخر، لایه‌های تمام متصل وجود دارد. ابتدا ۵ لایه پیچشی (هر یک شامل چند لایه پیچشی و یک لایه نمونه‌بردار یا انتخاب بیشینه^{۱۴}) وجود دارد که وظیفه‌ی استخراج ویژگی را به عهده دارند. خروجی آخرین لایه نمونه‌بردار به عنوان ویژگی‌های ورودی یک طبقه‌بند از نوع شبکه‌های عصبی چند لایه در نظر گرفته می‌شوند. این شبکه یک لایه ورودی ۴۰۹۶ تایی دارد که در شکل ۱ با عنوان FC1 یک لایه کاملاً متصل (به لایه بعدی) است. این لایه در معماری‌های معمول با نام لایه مسطح^{۱۵} نامیده می‌شود. سپس لایه پنهان شبکه عصبی با ۴۰۹۶ نورون را شاهدیم و در انتها لایه چگال^{۱۶} آخر را داریم که به تعداد کلاس‌ها نورون داشته و از نوع بیشینه‌ی نرم^{۱۷} است.



شکل ۳: معماری مدل تبدیل شده‌ی تمام پیچشی. مدل پیچشی-چگال معادل این ساختار در شکل ۲ آمده است.

جدول ۲: خلاصه مدل و پارامترهای مدل تمام پیچشی نمایش داده شده در

شکل ۳.

Layer (type)	Output Shape	Param #
Conv1: Conv2D(32,(3,3))	(None, 148, 148, 32)	896
MaxPool1: MaxPooling2D((2,2))	(None, 74, 74, 32)	0
Conv2: Conv2D(64,(3,3))	(None, 72, 72, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 36, 36, 64)	0
Conv3: Conv2D(128,(3,3))	(None, 34, 34, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(None, 17, 17, 128)	0
Conv4: Conv2D(128,(3,3))	(None, 15, 15, 128)	147584
MaxPool4: MaxPooling2D((2,2))	(None, 7, 7, 128)	0
Conv5: Conv2D(512,(3,3))	(None, 5, 5, 256)	295168
Conv6: Conv2D(512,(3,3))	(None, 3, 3, 512)	1180160
MaxPool5: MaxPooling2D((2,2))	(None, 1, 1, 512)	0
lastConv: Conv2D(1,(1,1))	(None, 1, 1, 1)	513
Total params: 1,716,673		

همانگونه که دیده می‌شود، به جای دو لایه آخر مدل قبل، لایه‌ی پیچشی lastConv اضافه شده است که عمل طبقه‌بندی به دو کلاس را انجام خواهد داد. حال در صورت داشتن یک مدل آموزش دیده از نوع پیچشی-چگال مناسب، کافیست شبکه تمام پیچشی معادل آنرا تعریف نموده و وزن‌های مدل قبل را در مدل جدید کپی نمود. همان‌گونه که ذکر شد یک لایه‌ی پیچشی، جایگزین دو لایه‌ی سطح و چگال آخر گردید. یک نکته‌ی اصلی همین موضوع است که این جایگزینی باعث می‌شود در صورت تغییر اندازه‌ی تصویر ورودی، تعداد پارامترهای مدل در تمام پیچشی تغییر نکند، در حالیکه تعداد پارامترها در مدل پیچشی-چگال اولیه، در چنین حالتی تغییر خواهد کرد. این ویژگی باعث می‌شود که مدل جدید، بدون نیاز به آموزش مجدد قابل اجرا بر روی ورودی‌هایی با هر اندازه باشد.

جدول ۱: خلاصه مدل و پارامترهای مدل پیچشی-چگال نمایش داده شده در شکل ۲.

Layer (type)	Output Shape	Param #
Conv1: Conv2D(32,(3,3))	(None, 148, 148, 32)	896
MaxPool1: MaxPooling2D((2,2))	(None, 74, 74, 32)	0
Conv2: Conv2D(64,(3,3))	(None, 72, 72, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 36, 36, 64)	0
Conv3: Conv2D(128,(3,3))	(None, 34, 34, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(None, 17, 17, 128)	0
Conv4: Conv2D(128,(3,3))	(None, 15, 15, 128)	147584
MaxPool4: MaxPooling2D((2,2))	(None, 7, 7, 128)	0
Conv5: Conv2D(512,(3,3))	(None, 5, 5, 256)	295168
Conv6: Conv2D(512,(3,3))	(None, 3, 3, 512)	1180160
MaxPool5: MaxPooling2D((2,2))	(None, 1, 1, 512)	0
FC: Flatten()	(None, 512)	0
Dense: Dense(1)	(None, 1)	513
Total params: 1,716,673		

خروجی این لایه از یک لایه نمونه‌بردار با اندازه‌ی صافی 2×2 عبور داده می‌شود که در نتیجه، اندازه آن نصف خواهد شد. خلاصه مدل و تعداد پارامترهای هر لایه آن در جدول ۱ آمده است. اولین لایه‌ی پیچشی با 3×3 صافی 896 پارامتر خواهد داشت؛ ورودی ما تصویر رنگی سه کاناله است، هر صافی یک بایاس هم دارد، لذا $896 = (1 + 3 \times 3 \times 3) \times 32$ پارامتر داریم. دقت به این نکته الزامی است که در این مدل تعداد پارامترها ربطی به طول و عرض تصویر ورودی ندارد. از همین نکته در بخش بعد استفاده خواهد شد. لایه‌ی نمونه‌بردار بعدی پارامتری ندارد. لایه‌ی پیچشی بعدی که $64 \times 3 \times 3$ صافی 3 را روی 32 تصویر خروجی لایه قبل اعمال می‌کند، 18496 پارامتر دارد: $18496 = (1 + 3 \times 3 \times 3) \times 64$.

آخرین لایه‌ی نمونه بردار (MaxPool5) یک لایه‌ی بدون پارامتر است که خروجی آن 512 تصویر 1×1 می‌باشد. این 512 عدد سطح شده و در قالب ورودی به آخرین لایه (Dense) داده می‌شوند؛ با احتساب بایاس، آخرین لایه 513 پارامتر خواهد داشت. نکته اصلی آن است که به جای دو لایه‌ی سطح و چگال آخر می‌توان یک لایه پیچشی با اندازه صافی 1×1 قرار داد. مشابه با لایه‌ی دوم پیچشی که روی خروجی لایه‌ی قبل از خود اعمال می‌شد و $64 \times 3 \times 3$ فیلتر با اندازه $3 \times 3 \times 3$ داشتیم، در اینجا هم چون خروجی لایه قبل 512 تصویر 1×1 است، لذا اعمال یک فیلتر یک در یک روی $512 \times 1 \times 1$ تصویر ورودی به لایه آخر، با احتساب بایاس، 513 پارامتر خواهد داشت. شکل ۳ ساختار مدل تغییر یافته و جدول ۲ جزئیات مدل حاصله را نشان می‌دهند.

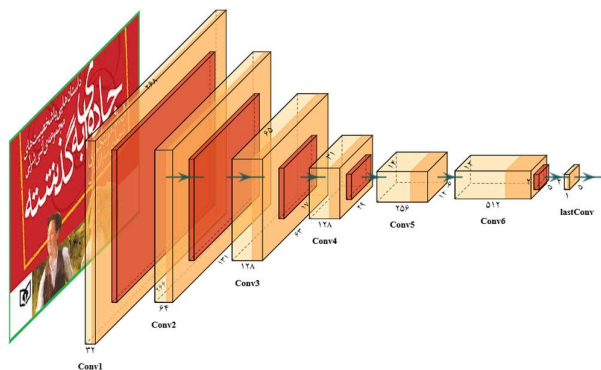
جدول ۳: خلاصه مدل و پارامترهای مدل پیچشی-چگال نمایش داده شده در

شکل ۴ برای تصویر با اندازه ورودی ۱۶۰×۲۶۸ .

Layer (type)	Output Shape	Param #
Conv1: Conv2D(32,(3,3))	(None, 158, 266, 32)	896
MaxPool1: MaxPooling2D((2,2))	(None, 79, 133, 32)	0
Conv2: Conv2D(64,(3,3))	(None, 77, 131, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 38, 65, 64)	0
Conv3: Conv2D(128,(3,3))	(None, 36, 63, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(None, 18, 31, 128)	0
Conv4: Conv2D(128,(3,3))	(None, 16, 29, 128)	147584
MaxPool4: MaxPooling2D((2,2))	(None, 8, 14, 128)	0
Conv5: Conv2D(512,(3,3))	(None, 6, 12, 256)	295168
Conv6: Conv2D(512,(3,3))	(None, 4, 10, 512)	1180160
MaxPool5: MaxPooling2D((2,2))	(None, 2, 5, 512)	0
lastConv: Conv2D(1,(1,1))	(None, 2, 5, 1)	513
Total params: 1,716,673		

۲-۲- اصلاح مدل برای پذیرش تصاویر با اندازه‌های متفاوت

در بخش قبل نحوه تبدیل یک شبکه پیچشی-چگال به شبکه‌ای تمام پیچشی بیان شد. اندازه تصویر ورودی در معماری ذکر شده در شکل ۲ و جدول ۱ ۱۵۰×۱۵۰ فرض شده بود. برداشت اولیه آن است که این شبکه فقط با تصاویری با همین اندازه کار می‌کند؛ در شبکه‌ی پیچشی-چگال به دلیل لایه‌های تمام متصل آخر (لایه‌های مسطح و چگال)، این تلقی درست است. اگر در معماری مدل، اندازه تصویر ورودی تغییر یابد، ۶ لایه اول پیچشی قادر به کار هستند، اما اندازه‌های خروجی‌های آنها (تعداد ویژگی‌ها) تغییر خواهد کرد. این تغییر باعث خواهد شد که در لایه‌ی مسطح، تعداد نورونها تغییر کرده و در نتیجه تعداد وزنها و ساختار مدل از این لایه به بعد دستخوش تغییر گردد. اگر به جای لایه‌های کاملاً متصل آخر، مطابق شیوه‌ی پیشنهادی در بخش قبل، از یک لایه‌ی پیچشی مناسب استفاده شود، شبکه قابلیت کار بر روی تصاویر با اندازه‌ی بزرگتر یا مساوی ۱۵۰×۱۵۰ را خواهد داشت. پس در واقع روش بخش پیش جوابگوی این مشکل هم هست که البته در این بخش به صورت مجزا مورد بررسی قرار گرفته است. فرض کنید اندازه تصویر ورودی ۱۶۰×۲۶۸ است. اگر اندازه ورودی شبکه بر اساس این اندازه‌ها تغییر کند، مدل جدید به صورت شکل ۴ خواهد بود که پارامترهای آن در جدول ۳ نمایش داده شده‌اند. اندازه تصویر ورودی با اندازه تصویر ۱۵۰×۱۵۰ که برای آموزش شبکه استفاده شده تفاوت دارد، مدل



شکل ۴: معماری مدل مورد استفاده تمام پیچشی روی تصویر مستطیلی با اندازه‌ی ۱۶۰×۲۶۸ .

کل ساختار مدل اولیه باید به گونه‌ای باشد که این تبدیل ممکن باشد، یعنی در صورت تغییر اندازه تصویر ورودی، تعداد پارامترهای مدل تبدیل شده، تغییر نکند. اگر مشابه معماری VGG یک یا دو لایه‌ی مسطح قبل از لایه‌ی مسطح آخری وجود داشته باشد، امکان تبدیل - بدون تغییر تعداد وزنها - وجود ندارد. در صورت تبدیل دو لایه‌ی آخر مسطح و چگال به پیچشی، تعداد پارامترهای لایه‌های مسطح قبلی، در صورت تغییر اندازه‌ی ورودی تغییر خواهند کرد و در نتیجه تعداد کل پارامترهای مدل تغییر خواهد کرد که باعث عدم امکان استفاده از وزنه‌های مدل اولیه خواهد شد. از آنجا که روش مدنظر در این مقاله، فقط وزنه‌های مدل اولیه‌ی پیچشی-چگال به مدل تبدیل شده‌ی تمام پیچشی کپی می‌شود، بحث آموزش مجدد مدل مطرح نیست و حتماً باید تعداد پارامترهای دو مدل یکسان باشند. ابزارهای مشهور تنسورفلو و پای‌تورچ توابع لازم برای کپی کردن وزنها از یک مدل به دیگری را دارا می‌باشند. پیاده‌سازی انجام شده برای این مقاله با تنسورفلو^{۲۱} است که از گیت‌هاب مرتبط با این نوشتار^{۲۲} قابل دانلود و اجراست.

در تابع `set_conv_weights` از فایل `FC2FC.py` نحوه کپی وزنه‌های لایه‌ی چگال آخر مدل پیچشی-چگال به لایه‌ی پیچشی مدل تمام پیچشی ذکر شده است. تابع `convert_model` در فایل `FC2FC.py` برای کپی وزنه‌های همه‌ی لایه‌های مشابه با هم در جدول‌های ۱ و ۲ نوشته شده است؛ در این تابع، تابع `set_conv_weights` فراخوانی شده و وزنه‌های لایه‌ی چگال مدل ۲ را به لایه‌ی پیچشی مدل ۳ کپی می‌کند. ساختارهای دو مدل فوق‌الذکر به نحوی تعریف شده بودند که کپی پارامترها امکان‌پذیر باشد. مثال اخیر برای حالت دو کلاس کارایی داشت. در بخش آزمایشات مثال شناسایی ارقام را هم خواهیم داشت که نمایانگر نحوه‌ی عمل برای حالت چند کلاس خواهد بود.

رابطه فوق برای محاسبه w برحسب x ، در صورت تغییر معماری مدل، عوض خواهد شد. به عنوان مثال اگر ساختار شبکه شامل سه لایه‌ی پیچشی + نمونه‌بردار باشد که صافی‌های لایه‌های پیچشی 3×3 با گام یک و صافی‌های لایه‌های نمونه‌بردار 2×2 باشند، w به صورت زیر محاسبه خواهد شد:

$$w = \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} (x - 2) - 2 \right) - 2 \right) \quad (3)$$

$$= \frac{x - 2^4 - 2}{2^3} = \frac{x - 14}{8}$$

درخصوص پیاده سازی، نیازی به آموزش مدل جدید نیست. فقط کافیست اندازه ورودی اولین لایه، متناظر با اندازه تصویر ورودی بروزرسانی شده و پارامترهای مدل پیچشی-چگال به این مدل داده شود. مواردی که گفته شد مربوط به مسائل دو کلاسه بودند، اگر تعداد طبقه‌ها بیشتر از دو باشد، کافیست در لایه آخر پیچشی، تعداد نوروهای 1×1 به تعداد کلاس‌ها در نظر گرفته شده و از تابع فعالیت بیشینه نرم استفاده گردد. اگر k کلاس داشته باشیم، اندازه‌ی آخرین لایه‌ی پیچشی ($h \times w \times k$) خواهد بود. در ادامه با ذکر چند کاربرد، باجزئیات روش کار بیشتر آشنا خواهیم شد.

۳- نتایج آزمایشات

در این بخش سه کاربرد با روش پیشنهادی مدل‌سازی شده‌اند، کاربرد اول، مسئله‌ی طبقه‌بندی و شناسایی دو رسم‌الخط مشهور **مشترک‌ترین ثلث** در تابلو نگاره‌ها و کاربرد دوم تفکیک برگ‌های سالم از میوه درخت سیب است که هر دو مسئله‌هایی دو کلاسه هستند. سومین مورد، شناسایی ارقام فارسی است که دارای ده طبقه می‌باشد. روش پیشنهادی در بستر کراس و تنسورفلو پیاده سازی شده است که از گیت‌هاب نگارنده در آدرس زیر قابل دسترس است:

<https://github.com/mamintoosi/FC2FC>

گرچه که دقت روش در نتایج آزمایشات ذکر شده است، اما هدف از این نوشتار، افزایش دقت یک طبقه‌بند نبوده است؛ هدف اصلی آن است که نحوه‌ی تبدیل یک شبکه‌ی پیچشی-چگال به شبکه عمیق تمام پیچشی و کاربرد این تبدیل نشان داده شود. هر مدل پیچشی-چگال مناسب (با هر دقتی که در دسترس باشد) می‌تواند با این شیوه به یک مدل تمام پیچشی تبدیل شده و قابلیت کار بر روی تصاویر با اندازه‌ی ورودی متغیر را داشته باشد؛ لذا موضوع مقایسه با سایر روش‌های طبقه‌بندی مطرح نیست.

هم مقداری تفاوت دارد اما پارامترهای مدل فرق نکرده است. دلیل این موضوع آن است که مدل نمایش داده شده در شکل ۳ تمام پیچشی بوده و هیچ یک از لایه‌ها و پارامترهای آن وابسته به اندازه تصویر ورودی نیستند. البته در زمان تعریف مدل باید اندازه ورودی آن برحسب اندازه تصویر جدید مشخص شود؛ که در این حالت معماری مدل به صورت شکل ۴ در خواهد آمد. در آخرین لایه که پیش از این یک لایه پیچشی 1×1 داشتیم، اکنون یک لایه پیچشی 2×5 داریم. اگر بر اساس اندازه‌های تصویر ورودی و ساختار شبکه پیش برویم به همین اندازه خواهیم رسید. برای مشخص کردن اندازه این لایه در حالت کلی، فرض کنید طول ضلع شکل مستطیلی ورودی در راستای یکی از محورها x باشد. در اولین لایه‌ی پیچشی ۲ تا کم خواهد شد، سپس با گذر از لایه‌ی نمونه‌بردار تقسیم بر ۲ خواهد شد (تقسیم صحیح) و الی آخر. اگر w پهنای آخرین لایه در راستای همین محور باشد، خواهیم داشت:

$$w = \left\lfloor \frac{\left\lfloor \frac{\left\lfloor \frac{\left\lfloor \frac{x-2}{2} \right\rfloor - 2}{2} \right\rfloor - 2}{2} \right\rfloor - 2}{2} \right\rfloor - 4 \quad (1)$$

عدد ۴ آخر به دلیل دو لایه پیچشی پشت سر هم ۵ و ۶ است که بین این دو، لایه‌ی نمونه‌بردار نداشتیم. از آنجا که اندازه ورودی 150×150 یکی از اندازه‌های مرسوم این حوزه است (و همچنین میانگین اندازه ورودی‌ها در بخش بعد)، در شبکه فرضی مورد بحث، همین اندازه در نظر گرفته شد. خروجی لایه پیچشی قبل از آخرین لایه نمونه‌بردار باید 3×3 باشد که نتیجه نمونه‌برداری 1×1 شود. به همین دلیل بین دو لایه پیچشی ۵ و ۶ لایه نمونه‌بردار قرار داده نشد. با صرف نظر کردن از جزء صحیح، رابطه (۱) به صورت زیر بازنویسی و خلاصه خواهد شد:

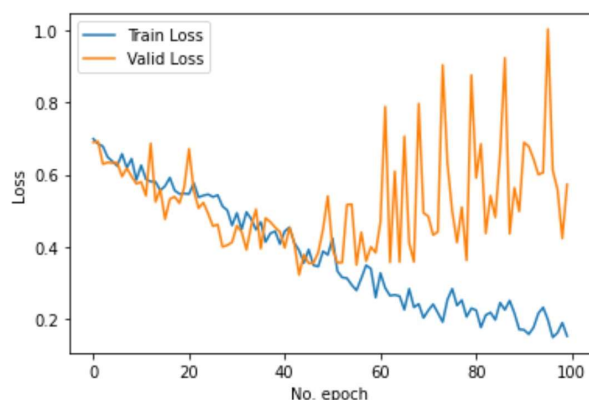
$$w = \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{x-2}{2} - 2 \right) - 2 \right) - 2 \right) - 4 \quad (2)$$

$$= \frac{x - 94}{32}$$

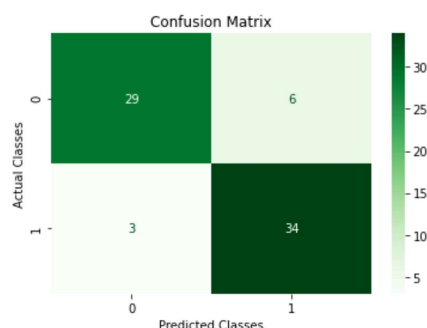
لذا برای تصویر ورودی با طول ضلع $x = 268$ ، w برابر ۵.۴۴ خواهد شد که جزء صحیح آن برابر ۵ خواهد بود. برای ارتفاع تصویر هم مقدار h برابر با ۲ خواهد بود:

$$h = \left\lfloor \frac{y - 94}{32} \right\rfloor = \left\lfloor \frac{160 - 94}{32} \right\rfloor = \left\lfloor 2.31 \right\rfloor = 2$$

به این ترتیب تصویر ورودی 160×268 منجر به آخرین لایه‌ی پیچشی با ابعاد $(h, w) = (2 \times 5)$ خواهد شد. هر یک از این ده نورو، طبقه‌ی ناحیه مرتبط با خود در تصویر ورودی را مشخص خواهند کرد.



شکل ۶: نمودار خطا (زیان) در ۱۰۰ تکرار روی داده‌ها. مدل با کمترین خطا بر روی مجموعه دادگان اعتبارسنجی، به عنوان بهترین مدل انتخاب می‌شود.



شکل ۷: ماتریس درهم ریختگی بر روی مجموعه دادگان آزمون در شناسایی تابلونگارها. در این اجرا میزان دقت ۸۸ درصد بوده است.

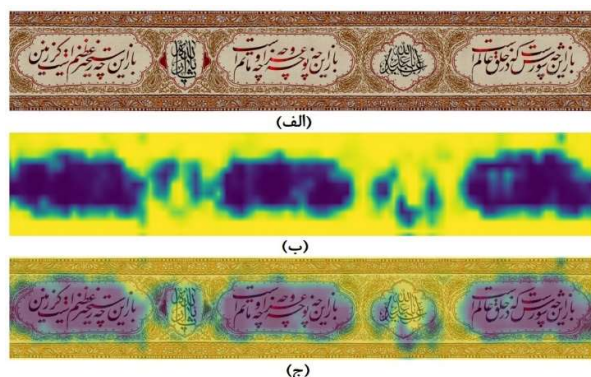
میانگین اندازه تصاویر ورودی 152×151 پیکسل است. از مدل پیچشی-چگال شکل ۲ در این مثال استفاده شده است. روش کلی اجرا، به صورت انتها به انتها^{۲۶} بوده و به جز تبدیل از بازه‌ی [۰، ۲۵۵] به بازه‌ی [۰، ۱] و داده افزایی^{۲۷} پردازش دیگری روی داده‌ها صورت نپذیرفته است. حداکثر زاویه ۴۵ درجه و میزان کم تغییر مقیاس و جابجایی در راستای محورها به عنوان پارامترهای داده‌افزایی در نظر گرفته شده بودند. برای تعیین تعداد آپک^{۲۸} بهینه، با اندازه دسته‌ی^{۲۹} برابر با ۳۲، صد تکرار روی داده‌ها انجام شده است. شکل ۶ نمودار تغییرات خطا^{۳۰} برای داده‌های آموزش و اعتبارسنجی را نشان می‌دهد. مدلی که روی داده‌های اعتبارسنجی کمترین خطا را داشته است به عنوان مدل بهینه انتخاب می‌شود. میزان دقت^{۳۱} روی داده‌های آزمون ۸۸ درصد بوده است. شکل ۷ ماتریس درهم ریختگی برای مدل انتخابی را نشان می‌دهد. طبقه‌ی مربوط به رسم الخط شکسته، کلاس صفر (منفی) و رسم الخط ثلث، کلاس یک (مثبت) در نظر گرفته شده بوده است.



شکل ۸: نمونه تصاویر آموزشی خط شکسته و ثلث که با یک خزنده از گوگل استخراج و پالایش شده‌اند. دو سطر اول مربوط به خط شکسته و دو سطر آخر، نمونه تابلوهایی با خط ثلث را نشان می‌دهد.

۳-۱- طبقه‌بندی تابلونگارهای خوشنویسی

مسئله‌ی پردازش خط و زبان فارسی از دیرباز مطرح بوده و کارهای متنوعی در این حوزه انجام شده است.^{۳۳} اما تا آنجا که نگارنده می‌داند هنوز کاری درخصوص طبقه‌بندی تابلونگارهای خوشنویسی انجام نشده است. شناسایی نوع قلم بکار رفته در متن، به عنوان یک مرحله پیش پردازش در شناسایی بصری حروف فارسی، از اهمیت بالایی برخوردار است. اما آنچه در این نوشتار مدنظر است، شناسایی نوع قلم به دلیل مذکور نیست؛ کاربردی از آن می‌تواند در یک سیستم تحلیل صحنه یا یک سیستم حاشیه نویسی خودکار تصویر^{۳۴} باشد. رسم الخطهای نستعلیق، شکسته و ثلث از جمله‌ی مشهورترین‌ها در خوشنویسی فارسی هستند. رسم الخطهای نستعلیق و شکسته نستعلیق در تابلونگارهای خوشنویسی محبوبیت ویژه‌ای دارند؛ خط ثلث نیز در متون و کتیبه‌های اماکن مذهبی جایگاه خاصی داشته و دارد. در این کاربرد با یک خزنده وب، با جستجوی عبارات «شکسته نستعلیق» و «خطاطی ثلث» در گوگل، حدود هزار تصویر برداشت شد. با ملاحظه بصری، تعداد زیادی از نتایج نامطلوب حذف شدند^{۳۵} و برای هر دسته ۳۵۵ تصویر لحاظ شدند. البته حذف تصاویر با کمی اغماض انجام شد. مثلاً تعداد معدودی از تصاویر شکسته نستعلیق به واقع جزو این خط محسوب نمی‌شدند - مثلاً نستعلیق بودند - اما از دیدگاه کلی شبیه به این خط بودند. برخی تابلوها دارای تصاویر پیش‌زمینه یا انسان هم بودند که اگر نوشته‌ی مربوطه، بخش اصلی تصویر را تشکیل می‌داد حذف نشدند. نمونه‌هایی از تصاویر باقیمانده در شکل ۸ قابل مشاهده هستند. حجم کل تصاویر ۷.۵ مگابایت است. تصاویر به سه دسته‌ی آموزش، اعتبارسنجی و آزمون تقسیم شدند. از مجموع ۷۱۰ تصویر ۸۰ درصد (۵۶۸ تصویر) به عنوان آموزش، ۷۰ تصویر به عنوان اعتبارسنجی و ۷۲ تصویر باقیمانده به عنوان داده آزمون در نظر گرفته شدند.



شکل ۹: خروجی بر روی تصویری شامل هر دو نوع رسم الخط. (الف) تصویر ورودی، (ب) نقشه رنگ حاصل از اعمال شبکه‌ی تمام پیچشی بر روی تصویر ورودی؛ نواحی تیره رنگ مشخص کننده‌ی مکان‌هایی از تصویر هستند که شبکه آنها را متعلق به گروه شکسته نستعلیق دسته‌بندی کرده است و نواحی روشن، به عنوان کتیبه‌های خط ثلث دسته‌بندی شده‌اند. (ج) نگاشت نقشه رنگ بر روی تصویر ورودی؛ تا حدود زیادی نواحی هر دو نوع رسم الخط به درستی شناسایی شده‌اند.

در این بخش یک شبکه‌ی پیچشی-چگال با دقت نزدیک به ۹۰ درصد برای طبقه‌بندی تابلونگارهای خوشنویسی فارسی به دو کلاس شکسته نستعلیق و ثلث طراحی و آموزش داده شد. با کپی کردن وزنه‌های این شبکه‌ی آموزش دیده به یک شبکه‌ی عمیق تمام پیچشی معادل آن، شبکه جدید قابلیت کار بر روی تصاویر ورودی با اندازه‌هایی متفاوت از اندازه تصاویر مربعی آموزشی را پیدا کرده است. کاربرد این تبدیل در شکل ۹ ملاحظه گردید.

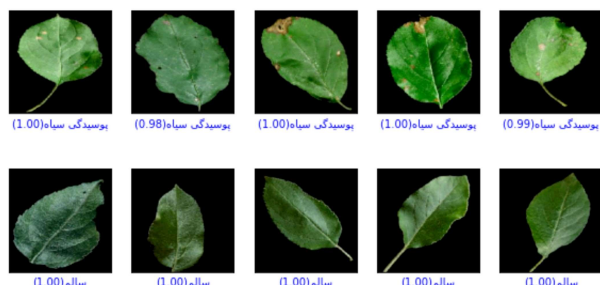
۳-۲- شناسایی برگ‌های سالم و بیمار با مدل تمام پیچشی

کاربرد دیگری از روش پیشنهادی که در این بخش مورد بررسی قرار می‌گیرد، شناسایی برگ‌های سالم از برگ‌های با بیماری «پوسیدگی سیاه»^{۳۲} درخت سیب است. تحقیقات متعددی در این زمینه انجام شده که در بسیاری از کارهای اخیر از یادگیری عمیق برای این مسئله استفاده شده است [۲۰، ۲۱، ۲۲]. در این تحقیقات، عموماً با استفاده از روش انتقال یادگیری^{۳۳} یک مدل معروف از قبل آموزش دیده انتخاب شده و بر روی یک مجموعه‌ی آموزشی از برگ‌های سالم و مریض آموزش داده شده است. عموم این شیوه‌ها مبتنی بر یک مدل پیچشی-چگال هستند که بر روی تصاویر برگ‌های مجزا آموزش دیده و قابلیت کارکرد و شناسایی بر روی تصویر یک برگ را دارند. با شیوه‌ی پیشنهادی در این نوشتار می‌توان یک مدل اینچنینی مناسب را به یک مدل تمام پیچشی تبدیل کرد که قابلیت عمل بر روی تصاویری شامل مجموعه‌ای از برگ‌ها را داشته باشد.

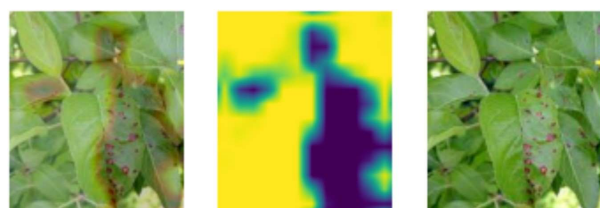


شکل ۸: خروجی روی ده تصویر آزمون. تصاویر آخر و اول از دو سطر اول و دوم، (با برچسب قرمز) به اشتباه طبقه‌بندی شده‌اند. مقادیر اعشاری میزان اطمینان هر مورد را نشان می‌دهد.

شکل ۸ نتیجه‌ی اجرای برنامه‌ی پایتون نوشته شده روی ده تصویر آزمون را نشان می‌دهد. سطر اول نمونه‌های شکسته نستعلیق (شکسته) هستند که همه به جز آخری به درستی تشخیص داده شده‌اند. سطر دوم نمونه‌های متعلق به رسم الخط ثلث هستند که اولی به اشتباه «شکسته» برچسب خورده است. مقادیر اعشاری میزان اطمینان هر مورد را نشان می‌دهد که همان خروجی آخرین لایه پیچشی است. از آنجا که فقط یک نورون در آخرین لایه پیچشی داشته‌ایم، خروجی شبکه عددی بین صفر تا یک بوده است. همین عدد برای کلاس یک در شکل فوق درج شده و یک منهای خروجی، به عنوان ضریب اطمینان مربوط به کلاس صفر نمایش داده شده است. با کپی کردن پارامترهای مدل پیچشی-چگال به مدل تمام پیچشی شکل ۳، مدل قابلیت عمل بر روی ورودی‌های با اندازه‌هایی فراتر از اندازه اولیه 150×150 را پیدا می‌کند. شکل ۹ (الف) یک کتیبه‌ی مربوط به شعر محتشم کاشانی با مطلع «باز این چه شورش است که در خلق عالم است» را نشان می‌دهد که با رسم الخط نستعلیق نگارش شده و مابین ابیات محتشم، نوشته‌هایی با خط ثلث هم دیده می‌شود. شکل ۹ (ب) خروجی شبکه‌ی تمام-پیچشی حاصل از تبدیل مدل پیچشی-چگال به تمام پیچشی را در قالب یک نقشه‌ی رنگ نشان می‌دهد که نقاط تیره‌تر به معنی احتمال تعلق بیشتر به کلاس شکسته نستعلیق و نقاط روشن‌تر به منزله‌ی میزان تعلق بیشتر به کلاس ثلث است. شکل ۹ (ج) نگاشت این نقشه‌ی رنگ بر روی تصویر اولیه را نشان می‌دهد. همان‌گونه که دیده می‌شود نواحی مشابه با تابلونگارهای آموزشی در تصویر به خوبی مشخص شده‌اند. در این کتیبه، ابیات محتشم کاشانی با خط نستعلیق هستند که به خط شکسته شبیه‌تر هستند تا ثلث. ابیات محتشم، جزو دسته‌ی شکسته علامت خورده‌اند که نمایانگر عملکرد مناسب شبکه است.

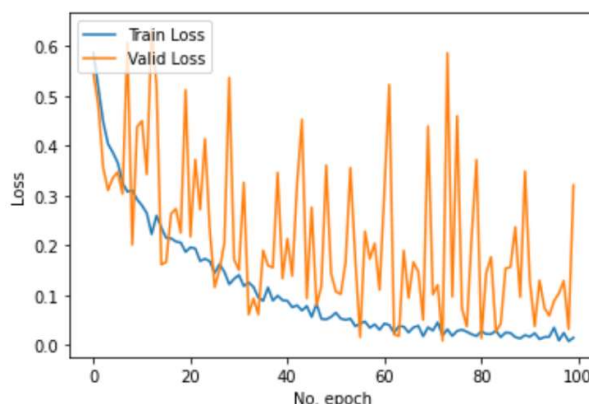


شکل ۱۲: خروجی روی ده تصویر آزمون. مقادیر اعشاری میزان اطمینان هر مورد را نشان می‌دهد.

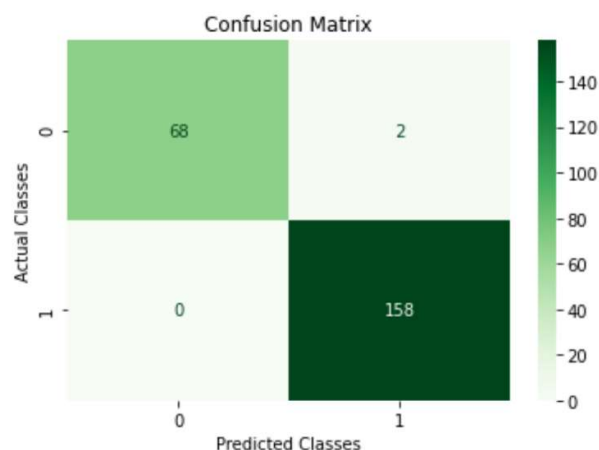


شکل ۱۳: خروجی بر روی تصویر شامل برگهای سالم و مریض. (الف) تصویر ورودی، (ب) نقشه رنگ حاصل از اعمال شبکه‌ی تمام پیچشی، نواحی آبی رنگ مشخص کننده‌ی مکان‌هایی از تصویر هستند که شبکه آنها را متعلق به گروه برگهای مریض دسته‌بندی کرده است و سایر نواحی مرتبط با برگهای سالم هستند. (ج) نگاشت نقشه رنگ بر روی تصویر ورودی. تا حدود زیادی نواحی مرتبط با برگهای مریض به درستی شناسایی شده‌اند.

زیر مجموعه‌ی مورد نظر در بخش دادگان گیت‌هاب این نوشتار قرار داده شده است. داده‌ها به سه گروه آموزش (۱۸۱۲ نمونه)، اعتبارسنجی (۲۲۶ نمونه) و آزمون (۲۲۸ نمونه) تقسیم شده‌اند. مشابه مثال بخش قبل، مدل بر روی داده‌های آموزشی، آموزش داده شد و مدل با کمترین خطا بر روی داده‌های اعتبارسنجی برای ادامه‌ی کار انتخاب گردید. شکل‌های ۱۰، ۱۱ و ۱۲ نمودار خطای روال آموزش، ماتریس درهم ریختگی و خروج مدل نهایی را نشان می‌دهند. دقت مدل بر روی داده‌های آزمون ۰٫۹۹ بوده است. با روش مورد بحث در این نوشتار، مدل پیچشی-چگال آموزش داده شده به یک مدل تمام پیچشی تبدیل می‌شود. شکل ۱۳ خروج مدل حاصله بر روی یک تصویر از برگ‌های یک درخت را نشان می‌دهد^{۲۵}. اگر قرار باشد تصویر ۱۳ (الف) به مدل معمول پیچشی-چگال، داده شود اولاً باید هم اندازه با تصاویر آموزشی شود و در ثانی مدل مذکور، کل این تصویر را به عنوان یک نمونه مورد بررسی قرار می‌دهد؛ در حالی که هدف، بررسی کل این تصویر به عنوان یک نمونه نیست. هدف در بررسی این تصویر، مشخص کردن سالم یا مریض بودن برگهای موجود در تصویر است.



شکل ۱۰: نمودار خطا (زیان) در ۱۰۰ تکرار روی داده‌ها. مدل با کمترین خطا بر روی مجموعه دادگان اعتبارسنجی، به عنوان بهترین مدل انتخاب می‌شود.



شکل ۱۱: ماتریس درهم ریختگی بر روی مجموعه دادگان آزمون در شناسایی برگهای مریض. در این اجرا میزان دقت ۹۹ درصد بوده است.

در اینجا هدف ایجاد یک مدل دقیق‌تر یا سریع‌تر از مدل‌های موجود نیست، لذا بحث مقایسه‌ی دقت یا سرعت مطرح نیست. نکته‌ی اصلی تغییر معماری مدل است که می‌تواند بر روی ورودی‌هایی با اندازه‌های متفاوت (شامل چندین برگ) کار کند. در این بخش از همان مدل بخش قبل (تفکیک تابلونگاره‌های خط شکسته و ثلث) استفاده خواهیم کرد. برنامه‌ی مربوطه با نام `FC2FC_PlantDisease.ipynb` از گیت‌هاب مقاله در دسترس است. مجموعه دادگان مختلفی برای شناسایی امراض گیاهان موجود است. برای این کاربرد از یکی از این مجموعه دادگان عمومی^{۳۴} استفاده خواهیم کرد که حجم کل آن حدود ۵ گیگابایت است. به منظور نمایش نحوه‌ی عمل فقط از بخش مربوط به نمونه برگ‌های سالم و بیمار درخت سیب (نوع پوسیدگی سیاه) استفاده خواهد شد.

جدول ۴: خلاصه مدل و پارامترهای مدل پیچشی-چگال برای تصویر مجموعه داده ارقام فارسی هدی با اندازه ورودی ۲۶×۲۶.

Layer (type)	Output Shape	Param #
Conv1: Conv2D(32,(3,3))	(None, 24, 24, 32)	320
MaxPool1: MaxPooling2D((2,2))	(None, 12, 12, 32)	0
Conv2: Conv2D(64,(3,3))	(None, 10, 10, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 5, 5, 64)	0
Conv3: Conv2D(128,(3,3))	(None, 3, 3, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(None, 1, 1, 128)	0
Flatten1 (Flatten)	(None, 128)	0
Dense1 (Dense)	(None, 10)	1290
Total params: 93,962		

جدول ۵: خلاصه مدل و پارامترهای مدل تمام پیچشی برای تصویر با اندازه ورودی ۱۳۵×۹۰۲.

Layer (type)	Output Shape	Param #
Conv1: Conv2D(32,(3,3))	(None, 133, 900, 32)	320
MaxPool1: MaxPooling2D((2,2))	(None, 66, 450, 32)	0
Conv2: Conv2D(64,(3,3))	(None, 64, 448, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 32, 224, 64)	0
Conv3: Conv2D(128,(3,3))	(None, 30, 222, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(None, 15, 111, 128)	0
lastConv: Conv2D(10,(1,1))	(None, 15, 111, 10)	1290
Total params: 93,962		

در صورت استفاده از مدل های معمول پیچشی-چگال باید برگ های تصویر با روشی دیگر استخراج شده و به مدل داده شود. اما مدل تمام پیچشی می تواند روی هر تصویری با هر اندازه ای کار کند؛ با این فرض که اندازه ی برگ ها در تصویر در حدود اندازه ی آنها در تصاویر آموزشی باشد، مدل جدید می تواند نتایج خوبی را به دست دهد. در این مثال برگ های مریض گروه با برچسب صفر و برگ های سالم گروه یک بوده اند. تصویر ۱۳ (ب) خروج مدل تمام پیچشی را نشان می دهد. نواحی روشن تر تعلق بیشتری به گروه یک (برگ های سالم) و نواحی تیره تر تعلق بیشتری به گروه صفر (برگ های بیمار) دارند. نگاشت این نقشه بر روی تصویر اصلی در شکل ۱۳ (ج) نمایش داده شده است. همان گونه که ملاحظه می شود تا حدود زیادی نواحی تیره تر مشخص کننده ی برگ های بیمار هستند. از آنجا که در این تصویر برگ ها با زوایای مختلف بوده اند و تصاویر آموزشی همه از یک منظر خاص تهیه شده اند، در روال داده افزایی برای آموزش شبکه، تبدیل های خاص (همچون دوران و انعکاس) بر روی تصاویر آموزشی اعمال شده است.

۳-۳- طبقه بندی ارقام فارسی با مدل تمام پیچشی

مدل مورد استفاده در دو مثال قبل با آخرین لایه ی پیچشی با اندازه ی ۱×۱ فقط مناسب طبقه بند دو کلاسه است. در این بخش نحوه ی کار روی مجموعه دادگان ارقام فارسی هدی [۲۳] را خواهیم دید که دارای ده طبقه می باشد. هدف در اینجا نمایش برتری از نظر دقت بر سایر روش ها نیست؛ هدف آن است که نشان داده شود چگونه یک شبکه عمیق مرسوم با لایه ی طبقه بند از نوع چگال چند کلاسه، به یک شبکه تمام پیچشی تبدیل شده و تصاویر ورودی با اندازه های مختلف را بپذیرد.

مشابه بخش پیش، ابتدا یک شبکه پیچشی عمیق معمول با لایه آخر طبقه بند ایجاد نموده و آموزش داده می شود. برای این کاربرد، یک شبکه با ۳ لایه پیچشی به ترتیب با تعداد ۳۲، ۶۴ و ۱۲۸ صافی که بعد از هر یک، یک لایه نمونه بردار قرار داده شده، تعریف شده است. پس از یک لایه ی مسطح ۱۲۸ تایی، در انتها یک لایه ی چگال ده تایی عمل طبقه بندی را انجام می دهد. ۵۴۰۰۰ نمونه برای آموزش و اعتبار سنجی و ۶۰۰۰ نمونه برای تست انتخاب شدند. مدل ده اپک آموزش داده شد. دقت مدل بر روی داده های آزمون ۰.۹۷ بوده است. نمودار خطای روال آموزش و جزئیات کار در برنامه ی مربوطه در گیت هاب این نوشتار موجود است که از درج آنها صرف نظر می شود. نکته اصلی تبدیل مدل پیچشی-چگال به تمام پیچشی است. خلاصه مدل پیچشی-چگال و پارامترهای آن در جدول ۴ آمده است. مدل تمام پیچشی جدیدی بر اساس وزن های مدل آموزش دیده فوق و روش مذکور در بخش های قبل ایجاد می شود. مدل پیچشی-چگال آموزش داده شده با این مدل تمام پیچشی یکی هست، با این تفاوت که دومی با تغییر اندازه ورودی و بدون آموزش مجدد، قابلیت پذیرش ورودی های با اندازه های متفاوت را دارا می باشد. در ادامه نتیجه ی اجرای مدل بر روی تصویری از یک دست نوشته ی حاوی ارقام فارسی با ابعاد ۱۳۵×۹۰۲ را خواهیم دید. مدل جدید تمام پیچشی متناسب با تصویر ورودی با اندازه ی ۱۳۵×۹۰۲ در جدول ۵ آمده است. آخرین لایه ی پیچشی، بعد آخر برابر با ۱۰ دارد که متناظر با تعداد کلاس های مورد طبقه بندی است. شکل ۱۴ (الف) بخشی از یک برگه ی امتحان ریاضی پایه ی هفتم را نشان می دهد. اگر این تصویر به شبکه داده شود، در خروجی ۱۰ نقشه با ابعاد ۱۵×۱۱۱ - متناظر با هر یک از ده رقم - خواهیم داشت.

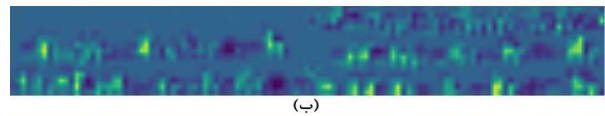
پیچشی با ذکر کاربردهایی بیان گردید. ابتدا مدل مرسوم پیچشی-چگال بر روی داده‌های آموزشی آموزش داده می‌شود. سپس یک مدل مشابه با مدل پیچشی-چگال که لایه‌ی طبقه‌بند کاملاً متصل آن با یک لایه‌ی پیچشی عوض شده است جایگزین می‌شود. مدل جدید تمام پیچشی با همان وزن‌های مدل پیچشی-چگال قادر به کار می‌باشد. شیوه‌ی عملی کپی وزن‌های مدل اول به مدل جدید در برنامه‌ی پایتون مرتبط با این نوشتار به صورت منبع باز در دسترس قرار داده شده است. روش مورد بحث در این نوشتار می‌تواند به عنوان راهکاری برای استفاده از مجموعه داده‌های پیشین، در بستر یادگیری عمیق بکار برده شود. مجموعه داده‌های برجسب‌زده‌ی بسیاری موجود هستند که قابلیت استفاده از شبکه‌های جدید یادگیری عمیق را دارند، اما همچون روش‌های قدیمی‌تر حوزه‌ی شناسایی الگو، مدل آماده شده فقط امکان کار با ورودی‌های با اندازه‌ی ثابت (هم اندازه با نمونه‌های آموزشی) را دارد. از طرفی برخی از مدل‌های جدید حوزه‌ی یادگیری عمیق نیازمند داده‌های آموزشی هستند که ناحیه‌ی اشیاء هدف در آنها به صورت دستی علامت‌گذاری شده باشند؛ که در مجموعه دادگانی که در روش‌های قدیمی‌تر فراهم شده‌اند، عموماً علامت‌گذاری تک به تک اشیاء در هر تصویر انجام نشده است. با راهکار مورد بحث در این نوشتار به عنوان **پلی به گذشته**، می‌توان ضمن استفاده از مجموعه دادگان آموزشی و مدل‌های مرسوم، مدلی داشت که بر روی ورودی‌هایی با هر اندازه قابلیت عملکرد را داشته باشد و بتواند نواحی مرتبط با اشیاء هدف در تصویر ورودی را مشخص کند. در واقع این شیوه امکان استفاده از ابزارهای جدیدتر با داده‌ها و مدل‌های مرسوم گذشته را فراهم می‌کند. دست آوردهای اصلی این نوشتار به شرح زیرند:

- بیان نحوه‌ی تبدیل شبکه‌های پیچشی-چگال به تمام پیچشی
- نمایش قابلیت استفاده از شبکه تبدیل شده با ورودی‌های با اندازه‌هایی فراتر از اندازه تصاویر آموزشی
- استفاده از مجموعه‌های آموزشی قدیمی در شبکه‌های پیچشی جدید
- عدم نیاز به علامت‌گذاری اشیاء هدف در تصاویر
- استفاده از شبکه‌های عصبی پیچشی برای طبقه‌بندی و شناسایی تابلونگارهای دو رسم‌الخط فارسی
- استفاده از روش پیشنهادی برای تفکیک برگ‌های بیمار از سالم در تصاویر دارای چندین برگ.

۶- حاصل عبارت‌های زیر را به صورت تقریبی به دست آورید.

$$\begin{aligned} -71+59 &= -70+50 = -20 & -83-48 &= -10-40 = -120 \\ -41+37-82 &= -30+30-80 = -90 & 37-71 &= 30-70 = -40 \end{aligned}$$

(الف)



۶- حاصل عبارت‌های زیر را به صورت تقریبی به دست آورید.

$$\begin{aligned} -71+59 &= -70+50 = -20 & -83-48 &= -10-40 = -120 \\ -41+37-82 &= -30+30-80 = -90 & 37-71 &= 30-70 = -40 \end{aligned}$$

(ج)

شکل ۱۴: خروجی شناسایی نواحی مرتبط با رقم ۷ بر روی تصویری از یک برگه‌ی ریاضی هفتم. (الف) تصویر ورودی، (ب) نقشه رنگ حاصل از اعمال شبکه‌ی تمام پیچشی بر روی تصویر ورودی؛ نواحی روشن‌تر مشخص‌کننده‌ی مکان‌هایی از تصویر هستند که مطابق خروجی شبکه، در آنجا به احتمال زیاد عدد ۷ وجود داشته است. (ج) نگاشت نقشه رنگ بر روی تصویر ورودی؛ تا حدود زیادی نواحی روشن‌تر مشخص‌کننده‌ی عدد ۷ هستند. عمده نواحی روشنی که در آن ناحیه رقم ۷ در تصویر نیست، تا حدودی مشابه ۷ بوده‌اند، مانند عدد ۲.

شکل ۱۴ (ب) خروجی نقشه‌ی متناظر با رقم ۷ را نشان می‌دهد که نقاط روشن‌تر به معنی احتمال تعلق بیشتر به کلاس ۷ است. شکل ۱۴ (ج) نگاشت این نقشه‌ی رنگ بر روی تصویر اولیه را نشان می‌دهد. همان‌گونه که دیده می‌شود عمده نواحی مرتبط با رقم ۷ به خوبی توسط مدل شناسایی شده‌اند. البته برخی نقاط، مانند عدد ۲ هم به اشتباه به عنوان ۷ شناسایی شده‌اند.

دو کاربرد اول، نمایش بهتری برای ایده‌ی این نوشتار بودند. در مواردی همچون شناسایی ارقام، بهتر است با استفاده از عملیات پیش‌پردازشی، هر کاراکتر استخراج شده و سپس ارقام شناسایی شوند. چنین پیش‌پردازشی برای تصویر اسناد متنی با افکنش‌های عمودی و افقی تصویر به راحتی مقدور است، اما چنین شیوه‌ی سرراستی برای جداسازی نمونه‌های دو کاربرد اول در تصویر، موجود نیست و راه‌حلهایی به مانند روش پیشنهادی می‌تواند در چنین مسائلی راهگشا باشد.

۴- جمع بندی

یادگیری عمیق بستر بسیار مناسبی برای کاربردهای تشخیصی در حوزه‌ی بینایی ماشین فراهم کرده است. مدل‌های مرسوم یادگیری عمیق مبتنی بر چند لایه‌ی پیچشی (کانولوشنی) و در نهایت لایه‌(های) طبقه‌بند هستند. در این مقاله از این مدل‌های مرسوم با عنوان پیچشی-چگال نام برده شد. در این نوشتار نحوه‌ی تبدیل یک شبکه‌ی پیچشی-چگال که لایه‌ی طبقه‌بند آن فقط از یک لایه‌ی مسطح و یک لایه‌ی چگال تشکیل شده است، به شبکه‌ی تمام

- Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.39, no.12, pp.2481–2495, 2017.
- [11] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in *Proceedings of the 12th European Conference on Computer Vision - Volume Part V, ECCV’12*, (Berlin, Heidelberg), p.746–760, Springer-Verlag, 2012.
- [12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision*, vol.88, pp.303–338, June 2010.
- [13] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, “The role of context for object detection and semantic segmentation in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp.740–755, Springer International Publishing, 2014.
- [15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp.1097–1105, Curran Associates, Inc., 2012.
- [17] محمد صبری، محمد شهرام معین، و فرید رزازی، “ارائه روش ترکیبی پویا بر اساس یادگیری عمیق به منظور بهبود کارایی سیستم های تطبیق بیومتری مبتنی بر کارت هوشمند”، نشریه مهندسی برق و مهندسی کامپیوتر، جلد ۱۸، شماره ۱-ب، صفحات ۲۹-۴۱، ۱۳۹۹.
- [18] X. Li, M. Kan, S. Shan, and X. Chen, “Weakly supervised object detection with segmentation collaboration,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [19] S. Sheykhi, S. Meshgini, and Z. Mousavi, “Automatic detection of various epileptic seizures from eeg signal using deeplearning networks,” *Computational Intelligence in Electrical Engineering*, vol.11, no.3, pp.1–12, 2020.
- [20] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, “Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks,” *IEEE Access*, vol.7, pp.59069–59080, 2019.
- [21] X. Chao, G. Sun, H. Zhao, M. Li, and D. He, “Identification of apple tree leaf diseases based on deep learning models,” *Symmetry*, vol.12, p.17, 06 2020.
- [22] M. B. Tahir, M. A. Khan, K. Javed, S. Kadry, Y.-D. Zhang, T. Akram, and M. Nazir, “Recognition of apple leaf diseases using deep learning and variances-controlled features reduction,” *Microprocessors and Microsystems*, p.104027, 2021.
- [23] H. Khosravi and E. Kabir, “Introducing a very large dataset of handwritten farsi digits and a study on their varieties,” *Pattern Recognit. Lett.*, vol.28, pp.1133–1141, 2007.

پاورقی ها

^۱ Convolutional Neural Networks (CNNs/ConvNets)

^۲ Annotated

^۳ https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

^۴ <https://github.com/shelhamer/fcn.berkeleyvision.org>

^۵ <http://www.image-net.org/>

^۶ Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL)

^۷ Visual Object Classes (VOC) <http://host.robots.ox.ac.uk/pascal/VOC/index.html>

^۸ <https://cs.stanford.edu/~roozbeh/pascal-context/>

^۹ <https://pjreddie.com/darknet/>

• نمایش امکان تبدیل به مدل تمام پیچشی بر روی مسئله طبقه بندی و شناسایی ارقام فارسی به عنوان یک مسئله دارای چندین دسته.

از جمله محدودیت های این شیوه می توان به موارد زیر اشاره کرد:

• هر مدل پیچشی-چگالی قابلیت تبدیل به تمام پیچشی را ندارد.

• به جز امکان کار بر روی تصاویر ورودی با اندازه ی متغیر، هر محدودیتی که در مدل اولیه وجود داشته است، در مدل تبدیل شده هم خواهد بود. به عنوان نمونه اگر مدل اولیه، دوران یا تغییر مقیاس نمونه های ورودی را مورد لحاظ قرار نمی داده است، مدل تبدیل شده هم، همان گونه خواهد بود.

روش پیشنهادی برای ایجاد یک مدل یادگیری عمیق از پایه، مناسب است و بسته به کاربرد، مدل نهایی می تواند بسیار کم حجم بوده و زمان آموزش زیادی لازم نداشته باشد. از آنجا که هدف این نوشتار افزایش دقت طبقه بند نبوده است، در خصوص کاربردهای ذکر شده، مقایسه ای با روش های قبل صورت نگرفته است؛ هر روش یادگیری عمیق مبتنی بر یک مدل پیچشی-چگال مناسب، با هر دقتی که وجود داشته باشد، با روش ذکر شده در این نوشتار، می تواند برای کاربردهای با اندازه ی ورودی متغیر به یک مدل تمام پیچشی تبدیل شود.

مراجع

- [1] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A survey of modern deep learning based object detection models,” *Digital Signal Processing*, vol. 126, p. 103514, 2022.
- [2] M. Schellenberg et al., “Semantic segmentation of multispectral photoacoustic images using deep learning,” *Photoacoustics*, vol. 26, p. 100341, 2022.
- [3] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.39, p.640–651, Apr. 2017.
- [4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp.580–587, 2014.
- [5] R. Girshick, “Fast R-CNN,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, (USA), p.1440–1448, IEEE Computer Society, 2015.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.39, no.6, pp.1137–1149, 2017.
- [7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask RCNN,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.42, no.2, pp.386–397, 2020.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.779–788, 2016.
- [9] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” *CVPR*, 2017.
- [10] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image

^{۱۰}Weakly Supervised Object Detection (WSOD)

^{۱۱}Build from scratch

^{۱۲}Fully Connected or Dense

^{۱۳}این کار در مسابقه شناسایی درختان ۲۷مین کنفرانس مهندسی برق ایران توسط نگارنده انجام شده است: <https://github.com/mamintoosi/tree-detection>

^{۱۴}MaxPooling

^{۱۵}Flatten

^{۱۶}Dense

^{۱۷}SoftMax

^{۱۸}با اینکه لایه‌های تمام متصل و چگال (متراکم) یکی هستند، اما به لحاظ اینکه در صورت نامگذاری مدل به پیچشی-تمام متصل، با نوع شبکه تمام پیچشی اشتراک لفظ پیدا می‌کند، به ناچار از عبارت «پیچشی-چگال» استفاده شد.

^{۱۹}Fully Convolutional

^{۲۰}ReLU

^{۲۱}Keras, Tensorflow

^{۲۲}<https://github.com/mamintoosi/FC2FC>

^{۲۳}به عنوان نمونه به تحقیقات احسان الله کبیر مراجعه شود:

<https://www.modares.ac.ir/~kabir>

^{۲۴}Automatic image annotation

^{۲۵}نگارنده دارای مدرک خوش از انجمن خوشنویسان ایران می‌باشد.

^{۲۶}end-to-end

^{۲۷}Data Augmentation

^{۲۸}Epoch

^{۲۹}Batch Size

^{۳۰}Loss

^{۳۱}Accuracy

^{۳۲}Apple Black rot

^{۳۳}Transfer Learning

^{۳۴}https://github.com/digitalepidemiologylab/plantvillage_deeplearning_paper_dataset

^{۳۵}تصویر برگ‌های درخت از سایت دانشگاه ایالت میشیگان اخذ شده است:

https://www.canr.msu.edu/ipm/diseases/black_rot_blossom_end_rot_frogeye_leaf_spot