



## Fully Connected 2 Fully Convolutional: Road to Yesterday

Mahmood Amintoosi

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

[m.amintoosi@hsu.ac.ir](mailto: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 full 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 such a trained network into a full 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 full 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 (as a two-class problem), classification of apple leaf diseases and the problem of identifying hand written Persian digits. For this purpose, first a convolutional network with the last full connected layer is designed and trained for square images. Then a new full 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. The test results show the efficiency of the proposed approach.

**Keywords-** Deep Learning, Convolutional Neural Networks, Image Classification, Object Detection, Multi-Layer Perceptron.

## تمام متصل به تمام پیچشی: پلی به گذشته

محمود امین طوسی

دانشکده ریاضی و علوم کامپیوتر. دانشگاه حکیم سبزواری. سبزوار. ایران.  
[m.amintoosi@hsu.ac.ir](mailto: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. عمدتی این شبکه‌ها متشتم منعماً های با اندازه‌ی بزرگ هستند که توانایی آشکارسازی دهای یا صدای دسته‌ی از قبل مشخص را داشته باشند. در بیشتر کاربردهای استفاده می‌شود که پس از چند لایه پیچشی از یک طبقه‌بند معمول برای طبقه‌بندی ویژگی‌های استخراج شده‌ی شبکه استفاده می‌شود. در این نوشتار روش تبدیل یک شبکه آموزش دیده‌ی اینچنینی 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 حوزه یادگیری عمیق عموماً از تصاویر آموزشی که در آنها نواحی مورد نظر با ماسک مشخص شده‌اند استفاده می‌شود، اما در شیوه‌ی پیشنهادی در این نوشتار فقط تصاویر برچسب‌دار (مشخص‌کننده طبقه‌ی تصویر) به شبکه داده می‌شود. جزییات روش کار در قالب مسئله‌ی جدید طبقه‌بندی و شناسایی تابلوهای با رسم الخطهای شکسته نستعلیق و ثلث، شناسایی برگ سالم از مرضی سیب (به عنوان مسائل دو کلاسه) و مسئله‌ی شناسایی ارقام فارسی و مسئله‌ی شناسایی ارقام فارسی را نشان داده است. به این منظور ابتدا یک شبکه پیچشی با لایه آخر تمام متصل طراحی و بر روی تصاویر مربوطی آموزش داده می‌شود. سپس مدل تمام پیچشی جدیدی بر اساس مدل قبلی تعریف شده و وزنهای مدل قبلی به مدل جدید خورانده 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، پرسپکتورون چند لایه

### ۱- مقدمه

ناحیه علامت زده شده و حجم داده‌های حدود یک و نیم گیگابایت [۱۲] که شامل طبقه‌های مختلفی 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] شامل دو و نیم میلیون نمونه‌ی برچسب زده شده در ۳۲۸ هزار تصویر.

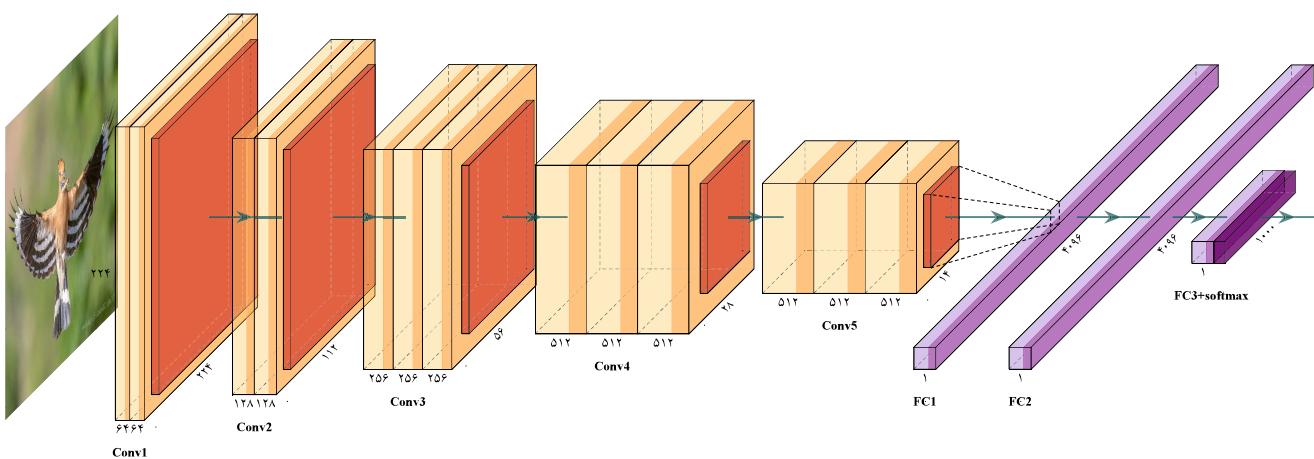
این مجموعه داده‌های بزرگ عموماً در ایجاد معنایی مانند VGG [۱۵]، AlexNet [۱۶] و ResNet [۱۷] باکار برده DarkNet [۹] و <sup>۸</sup> می‌باشد که مبنای بیشتر مدل‌های مرتبط است. به این مدل‌ها مدل از قبل آموزش دیده اطلاق می‌شود. شکل ۱ ساختار مدل VGG16 را نشان می‌دهد که بالغ بر ۱۳۰ میلیون پارامتر دارد. در [۱۸] 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 بر شمرده شده است.

ایجاد مجموعه داده‌های آموزشی که اشیاء مدنظر در هر تصویر با یک

در سالیان اخیر شبکه‌های عصبی پیچشی <sup>۱</sup> کارایی چشمگیری در شناسایی الگوهای پیچیده و آشکارسازی اشیاء به نمایش گذاشته‌اند [۲]. روش‌های <sup>۲</sup>R-CNN [۴]، <sup>۳</sup>Fast R-CNN [۵]، <sup>۴</sup>FCN [۶]، <sup>۵</sup>YOLO [۹]، <sup>۶</sup>Mask R-CNN [۷] و <sup>۷</sup>SegNet [۱۰] 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 <sup>۲</sup> از الزامات مدل‌های پیچیده‌ی جدید این حوزه هستند. به عنوان نمونه، مجموعه داده <sup>۳</sup> NYU Depth Dataset [۱۱] که در گیت‌هاب <sup>۴</sup> مربوط به مقاله [۳] به آن اشاره شده، در نسخه خام، ۴۲۸ گیگابایت و داده‌های برچسب زده شده آن ۲,۸ میلیون تصویر دارند. این مجموعه داده مربوط به فضاهای داخلی می‌باشد. چند نمونه مجموعه داده دیگر در ادامه آمده است:

• مجموعه داده <sup>۵</sup> ImageNet شامل ده میلیون تصویر برچسب زده شده از بیش از ۱۰۰۰ دسته.

• مجموعه داده <sup>۶</sup> پاسکال با بیش از ده هزار تصویر شامل ۲۳۳۷۴



شکل ۱: معماری مدل VGG16 [۱] (چارچوب مدل با استفاده از ابزار سایت <https://github.com/HarisIqbal88/PlotNeuralNet> رسم شده است. حق نشر تصویر هددهد متعلق به مجید مؤمنی مقدم عضو هیات علمی گروه زیست شناسی دانشگاه حکیم سبزواری است).

کوچکی طراحی شده است که درگیر مدل‌های بزرگ نباشیم و موضوع کاپبل درکتر باشد.

به نظر می‌رسد عواملی همچون حجم زیاد مدل‌های از قبل آموختن 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 راهحلها و مجموعه دادگان قبلی آماده شده توسط محققین به راحتی در بستر یادگیری عمیق، با دقت بیشتر قابلیت استفاده مجدد پیدا کرده و قابلیت عمل با تصاویر ورودی با اندازه متغیر را هم داشته باشند. برای کاربردهای فراوانی، مجموعه داده‌های آموختشی خاصی آماده شده است، اما قابلیت استفاده در شیوه‌های جدید همچون Mask RCNN را ندارند. به عنوان مثال اگر مجموعه دادگانی برای تفکیک دو طبقه‌ی تصویری موجود باشد، قاعده‌تاً تعدادی تصویر از هر دو دسته برای عمل آموختش فراهم شده است. در روش‌هایی مانند Mask RCNN باید دور سوزه‌ی مربوطه در تصاویر آموختشی را با چند ضلعی<sup>۱۳</sup> یا با یک ماسک مشخص نمود. اما در شیوه‌ی پیشنهادی، تصاویر آموختشی در قالب مرسوم پیشین، قابلیت استفاده را دارند. 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 تضمیم‌گیری بر روی کل تصویر باشد و طبقه کل تصویر را مشخص کند. به عنوان کاربردهای شیوه‌ی پیشنهادی، مسائل شناسایی دو رسم الخط مشهور **شکر تسلیق** و **ثلث** در تابلو نگاره‌ها، تفکیک برگ‌های سالم و مریض درخت سیب و شناسایی ارقام فارسی در نظر گرفته شده است که در بخش <sup>۲</sup> به تفصیل بیان خواهد شد.

## ۲- شیوه‌ی پیشنهادی

همان‌گونه که در بخش قبل اشاره شد، بسیاری از مدل‌های پیچشی عمیق متناسبن چند لایه‌ی پیچشی و در انتها، لایه‌هایی تمام متصل برای طبقه‌بندی می‌باشند. شکل <sup>۱</sup> معماری مدل VGG16 را نشان می‌دهد که همان‌گونه که ملاحظه می‌شود در آخر، لایه‌های تمام متصل وجود

کادر یا ماسک برچسب‌گذاری شوند، مستلزم هزینه و زمان براست [۱۸]. ساده‌تر کردن تهیه داده‌های آموختشی با عنوان شناسایی اشیاء با نظارت مختصر<sup>۱۰</sup> یکی از زیرمسائل شبکه‌های پیچشی در بینایی ماشین محسوب می‌شود.

نسخه‌های مختلف YOLO و R-CNN همگی مبتنی بر یک مدل از قبیل آموختش دیده هستند. اندازه این مدل‌ها معمولاً بسیار بزرگ بوده و شامل میلیون‌ها پارامتر هستند. برای آموختش 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 قبیل آموختش دیده استفاده شده و نیاز به پردازش‌های سنگین بطرف می‌شود. در غیر اینصورت مدل باید از پایه طراحی و آموختش داده شود<sup>۱۱</sup>.

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 طبقه‌بند مانند شبکه‌های عصبی پرسپترون یا ماشین بردار پشتیبان بکار گرفته می‌شود. تعویض این طبقه‌بند‌ها با سایر طبقه‌بند‌ها از جمله نواوری‌های مرسوم این حوزه است [۱۹]. از آنجا که طبقه‌بند‌های مرسوم بر روی بردارهای ویژگی با طول یکسان قابلیت کار دارند، به عنوان مثال اگر بردار ویژگی حاصل از آخرین لایه پیچشی یک بردار هزارتایی باشد، ورودی طبقه‌بند، هزارتایی خواهد بود و طبقه‌بندها با چنین داده‌هایی آموختش داده خواهد شد و اگر به واسطه‌ی تغییر اندازه تصویر ورودی به شبکه پیچشی، اندازه این بردار تغییر کند، طبقه‌بند قادر به عمل نخواهد بود. هدف اصلی این نوشتار بیان نحوه جایگزینی لایه آخر کاملاً متصل <sup>۱۲</sup> 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 لایه پیچشی در <sup>۳</sup>[FCN] بکار برده شده است. بخش قابلیت عملکرد بر روی تصاویر با اندازه متغیر هم در <sup>۴</sup>[RCNN Mask] مورد استفاده قرار گرفته است. در این نوشتار نحوه تلفیق این دو و ایجاد مدل‌های تمام پیچشی با اندازه 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 آموختشی که در آن نمونه‌های مورد طبقه‌بندی با ماسک مشخص شوند، نیست. گرچه که شیوه مدنظر ربطی به اندازه مدل ندارد، اما در کاربرد مورد بررسی، مدل

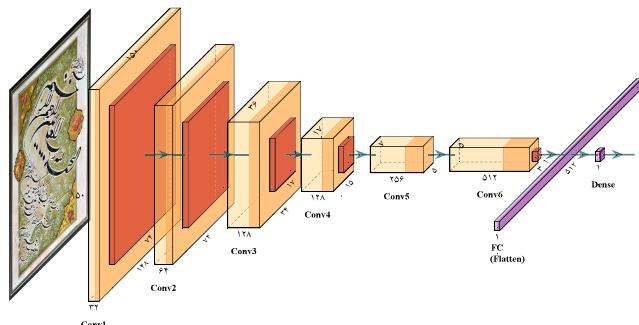
Layer (type)	Output Shape	Param #
Conv1: Conv2D(32, (3,3))	(None, 148, 148, 32)	896
MaxPool1: MaxPooling2D((2,2))	(None, 74, 74, 32)	0
Conv2: Conv2D(64, (3,3))	(None, 72, 72, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 36, 36, 64)	0
Conv3: Conv2D(128, (3,3))	(None, 34, 34, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(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	

آخرین لایهی نمونهبردار (MaxPool5) یک لایهی بدون پارامتر است که خروجی آن  $5 \times 5$  تصویر  $1 \times 1$  میباشد. این  $5 \times 5$  عدد مسطح شده و در قالب ورودی به آخرین لایه (Dense) داده میشوند؛ با احتساب بایاس، آخرین لایهی  $5 \times 1$  پارامتر خواهد داشت.

نکته اصلی آن است که به جای دو لایهی آخر Flatten، Dense می‌توان یک لایه پیچشی با اندازه صافی  $1 \times 1$  قرار داد. مشابه با لایهی دوم پیچشی که روی خروجی لایهی قبل از خود اعمال می‌شد و  $64 \times 64$  فیلتر با اندازه  $3 \times 3$  داشتیم، در اینجا هم چون خروجی لایه قبل  $512 \times 512$  است، لذا اعمال یک فیلتر یک در یک روی  $1 \times 1$  تصویر  $1 \times 1$  است. تصویر ورودی به لایه آخر، با احتساب بایاس،  $513 \times 513$  پارامتر خواهد داشت.

شکل ۲ ساختار مدل تغییر یافته و جدول ۲ جزییات مدل حاصله را نشان می‌دهند. همان‌گونه که دیده می‌شود، به جای دو لایه آخر مدل قبلی، لایهی lastConv اضافه شده است که عمل طبقه‌بندی به دو کلاس را پیچشی 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 قبلی را به مدل جدید منتقل نمود. منظور از مناسب آن است که



شکل ۲: معماری مدل اولیه‌ی پیچشی-چگال مورد استفاده در روش پیشنهادی با یک لایه کاملاً متصل ۵۱۲ تایی (Fully Connected) و لایه‌ی آخر چگال (Dense).

دارد. ابتدا ۵ لايهی پیچشی (هر يك شامل چند لايهی پیچشی و يك لايهی نمونهبردار يا انتخاب بیشینه<sup>۱۴</sup>) وجود دارد که وظیفه‌ی استخراج ويژگی را به عهده دارد. خروجی آخرین لايهی نمونهبردار به عنوان ويژگی‌های ورودی يك طبقه‌بند از نوع شبکه‌های عصبی چند لايه درنظر گرفته می‌شوند. اين شبکه يك لايهی ورودی ۴۰۹۶ تایی دارد که در شکل ۱ با عنوان FCI یك لايه کاملاً متصل (به لايه بعدی خود) است. اين لايه در معماری‌های معمول با نام لايه مسطح<sup>۱۵</sup> نامیده می‌شود. سپس لايه پنهان شبکه عصبی با ۴۰۹۶ نورون را شاهدیم و در انتهای لايه چگال<sup>۱۶</sup> آخر را داریم که به تعداد کلاس‌ها نورون داشته و از نوع بیشینه‌ی نرم<sup>۱۷</sup> است. اين معماری، ساختار مرسوم شبکه‌های پیچشی عمیق است که در آدامه آنرا مدل «پیچشی-چگال» خواهی نامید<sup>۱۸</sup>.

هدف اصلی آن است که بخش طبقبند این معماری با یک لایه‌ی پیچشی به نحوی تعریض 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 به یک مدل تمام پیچشی<sup>۱۹</sup> خورانده شود. طراحی و آموزش مدل پیچشی – چگال ساده است و قابلیت کار بر روی بسیاری از مجموعه دادگان 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.

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

مدل پیچشی- چگال شکل ۲ را در نظر بگیرید. مدل مورد نظر از ۶ لایه‌ی پیچشی (Conv1,...,Conv6)، یک لایه کاملاً متصل ۵۱۲ تایی و یک لایه‌ی آخر چگال با یک نورون تشکیل شده است. که قابلیت استفاده برای مسائل طبقه‌بندی دو کلاسه را دارد. تابع فعالیت هر لایه پیچشی، واحد خطی اصلاح شده  $z^*$  در نظر گرفته شده که با رنگ نارنجی در هر بلوک پیچشی مشخص شده است. لایه‌های نمونه‌برداری (یا انتخاب بیشینه یا ادغام) با رنگ قرمز و لایه‌های تمام متصل (FC) و چگال با رنگ بنفش مشخص شده‌اند. تابع فعالیت آخرین لایه یک تابع سیگموئید است که با رنگ بنفش پر رنگ نمایش داده شده است.

ورودی مدل یک تصویر مربعی  $150 \times 150$  است که بعد از گذراز اولین لایه پیچشی با گام یک و اندازه صافی (فیلتر)  $3 \times 3$  به اندازه  $148 \times 148$  می‌رسد. اعداد بالا و پایین لایه‌های پیچشی بیانگر اندازه‌های

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

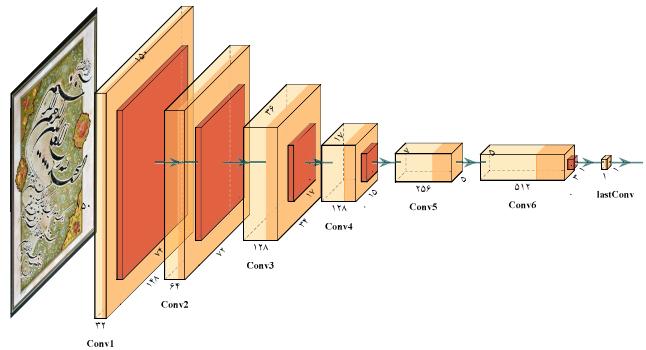
Layer (type)	Output Shape	Param #
Conv1: Conv2D(32, (3,3))	(None, 148, 148, 32)	896
MaxPool1: MaxPooling2D((2,2))	(None, 74, 74, 32)	0
Conv2: Conv2D(64,(3,3))	(None, 72, 72, 64)	18496
MaxPool2: MaxPooling2D((2,2))	(None, 36, 36, 64)	0
Conv3: Conv2D(128,(3,3))	(None, 34, 34, 128)	73856
MaxPool3: MaxPooling2D((2,2))	(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
lastConv: Conv2D(1,(1,1))	(None, 1, 1, 1)	513
Total params:	1,716,673	

خواهد بود که پارامترهای آن در جدول ۲ نمایش داده شده‌اند. اندازه تصویر ورودی با تصویر  $150 \times 150$  که برای آموزش شبکه استفاده شده تفاوت دارد، مدل هم مقداری تفاوت دارد اما پارامترهای مدل فرقی نکرده است. دلیل این موضوع آن است که مدل نمایش داده شده در شکل ۲ 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 بر حسب اندازه تصویر جدید مشخص شود؛ که در این حالت ظاهر مدل به صورت شکل ۴ در خواهد آمد. در آخرین لایه که پیش از این یک لایه پیچشی  $1 \times 1$  داشتیم، اکنون یک لایه پیچشی  $5 \times 5$  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 لایه پیچشی ۲ تا کم خواهد شد، سپس با گذر از لایه‌ی نمونه‌بردار تقسیم بر ۲ خواهد شد (تقسیم صحیح) و الی آخر. اگر  $w$  پنهانی آخرین لایه در راستای همین محور باشد، خواهیم داشت:

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

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  $(150 \times 150)$  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  $(3 \times 3)$  باشد که نتیجه نمونه‌برداری  $(1 \times 1)$  شود. به همین دلیل بین دو لایه پیچشی ۵ و ۶ لایه نمونه‌بردار قرار داده نشد.

با صرفنظر کردن از جزء صحیح، رابطه (۱) به صورت زیر بازنویسی و



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

وزن‌ها به لحاظ مفهومی و ساختاری قابل تبدیل باشند. ابزارهای مشهور تنسورفلو و پای تورج توابع لازم برای کپی کردن وزنها از یک مدل به دیگری را دارا می‌باشند. پیاده‌سازی انجام شده برای این مقاله با تنسورفلو<sup>۱</sup> است که از گیت‌هاب مرتبط با این نوشتار<sup>۲</sup> قابل دانلود و اجراست. در تابع set\_conv\_weights از فایل FC2FC.py نحوه‌ی کپی وزن‌های لایه‌ی چگال آخر مدل پیچشی-چگال به لایه‌ی پیچشی مدل تمام پیچشی ذکر شده است. تابع convert\_model در فایل FC2FC.py برای کپی وزن‌های لایه‌ی همه‌ی لایه‌های مشابه با هم در جدول‌های ۱ و ۲ نوشته شده است؛ در این تابع، تابع set\_conv\_weights فراخوانی شده و وزن‌های لایه‌ی چگال مدل ۲ را به لایه‌ی پیچشی مدل ۳ کپی می‌کند. نکته‌ی اصلی آن است که خواص این دو لایه باید به نحوی باشد که وزن‌های مربوطه متناظر با هم باشند. ساختارهای دو مدل فوق الذکر به نحوی تعریف شده بودند که این موضوع صادق باشد.

مثال اخیر برای حالت دو کلاسه کارایی داشت. در بخش آزمایشات مثال شناسایی ارقام را هم خواهیم داشت که نمایانگر نحوه‌ی عمل برای حالت چند کلاسه خواهد بود.

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

در بخش قبل نحوه‌ی تبدیل یک شبکه پیچشی-چگال به شبکه تمام پیچشی<sup>۳</sup> بیان شد. اندازه تصویر ورودی در معماری ذکر شده در شکل ۲ و جدول ۱  $150 \times 150$  فرض شده بود. برداشت اولیه آن است که این شبکه فقط با تصاویری با همین اندازه کار می‌کند؛ در وضعیت فعلی شبکه به دلیل لایه‌های تمام متصل آخر، این تلقی درست است. اگر در معماری مدل، اندازه تصویر ورودی تغییر باید، ۶ لایه اول پیچشی قادر به کار هستند، اما اندازه‌های خروجی‌های آنها تغییر خواهد کرد. این تغییر باعث خواهد شد که در لایه‌ی Flatten 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 هر اندازه‌ی بزرگتر یا مساوی با  $150 \times 150$  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  $268 \times 160$  است. اگر اندازه ورودی شبکه بر اساس این اندازه‌ها تغییر کند، مدل جدید به صورت شکل ۴

جدول ۳: خلاصه مدل و پارامترهای مدل پیچشی-چگال نمایش داده شده در شکل ۴ برای تصویر با اندازه ورودی  $160 \times 268$ .

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	

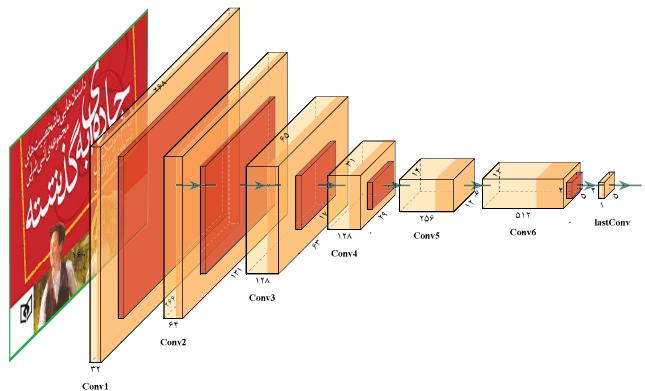
### ۳- نتایج آزمایشات

در این بخش سه کاربرد با روش پیشنهادی مدلسازی شده‌اند، کاربرد اول، مسئله‌ی طبقه‌بندی و شناسایی دو رسم الخط مشهور **شکته‌تعیق** و **ثلث** در تابلو نگاره‌ها و کاربرد دوم تفکیک برگ‌های سالم از مریض درخت سبب 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 پیاده‌سازی شده است که از گیت‌هاب نگارنده به آدرس <https://github.com/mamintoosi/FC2FC> دسترس است.

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 اندازه ورودی متغیر را داشته باشد؛ لذا موضوع مقایسه با سایر روش‌های طبقه‌بندی مطرح نیست.

### ۳-۱- طبقه‌بندی تابلونگاره‌های خوشنویسی

مسئله‌ی پردازش خط و زبان فارسی از دیرباز مطرح بوده و کارهای متنوعی انجام شده است.<sup>۲۳</sup> اما تا آنجا که نگارنده می‌داند هنوز کاری درخصوص طبقه‌بندی تابلونگاره‌های خوشنویسی انجام نشده است. شناسایی نوع قلم بکار رفته در متن، به عنوان یک مرحله پیش‌پردازشی 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 صحفه یا یک سیستم حاشیه‌نویسی خودکار تصویر <sup>۲۴</sup> باشد. رسم الخط‌های **تعیق**، **شکته‌تعیق** و **ثلث** از جمله مشهورترین‌ها در خوشنویسی فارسی هستند. رسم الخط شکسته نستعلیق



شکل ۴: معماری مدل مورد استفاده تمام پیچشی روی تصویر مستطیلی با اندازه  $160 \times 268$ .

خلاصه خواهد شد:

$$\begin{aligned} w &= \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{x-2}{2} - 2 \right) - 2 \right) - 2 \right) - 4 \\ &= \frac{x-94}{32} \end{aligned} \quad (2)$$

لذا برای تصویر ورودی با طول ضلع  $x = 268$ ،  $w$  برابر  $5/44$  خواهد شد که جزو صحیح آن برابر ۵ خواهد بود. برای ارتفاع تصویر هم مقدار  $h$  برابر با ۲ خواهد بود:

$$h = \lfloor \frac{y-94}{32} \rfloor = \lfloor \frac{160-94}{32} \rfloor = \lfloor 231 \rfloor = 2$$

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

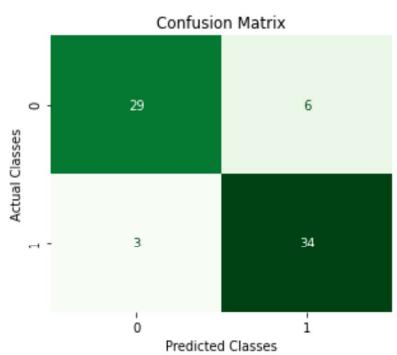
رابطه فوق برای محاسبه  $w$  بر حسب  $x$  در صورت تغییر معماری مدل، عوض خواهد شد. به عنوان مثال اگر ساختار شبکه شامل سه لایه‌ی پیچشی+نمونه‌بردار باشد که صافی‌های لایه‌های پیچشی  $3 \times 3$  با گام یک و صافی‌های لایه‌های نمونه‌بردار  $2 \times 2$  باشند،  $w$  به صورت زیر محاسبه خواهد شد:

$$\begin{aligned} w &= \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} (x-2) - 2 \right) - 2 \right) \\ &= \frac{x-24-2}{2^3} = \frac{x-14}{8} \end{aligned} \quad (3)$$

درخصوص پیاده‌سازی، نیازی به آموزش مدل جدید نیست. فقط کافیست اندازه ورودی اولین لایه متناظر با اندازه تصویر ورودی بروزسانی شده و پارامترهای مدل پیچشی-چگال یا مدل تمام پیچشی 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 نورون‌های SoftMax ( $1 \times 1$ ) به تعداد کلاس‌ها درنظر گرفته شده و ازتابع فعالیت  $k$  کلاس داشته باشیم، اندازه‌ی آخرین لایه‌ی پیچشی استفاده کرد. اگر  $k$  کلاس داشته باشیم، اندازه‌ی  $h \times w \times k$  خواهد بود. در ادامه با بیان کاربردهایی از این شیوه با جزئیات روش کار بیشتر آشنا خواهیم شد.



شکل ۵: نمونه تصاویر آموزشی خط شکسته و ثلث که با یک خزنه از گوگل استخراج و پالایش شده‌اند.

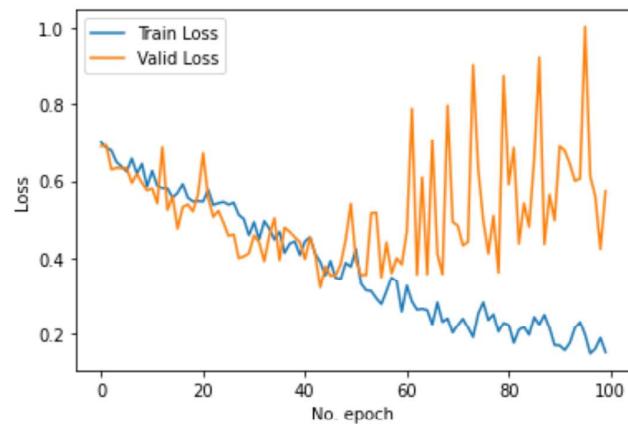


شکل ۷: ماتریس درهم‌ریختگی در شناسایی تابلونگاره‌ها

صورت نپذیرفته است. حداکثر زاویه ۴۵ درجه و میزان کمی تغییر مقیاس و جایجایی در راستای محورها به عنوان پارامترهای افزایش نمونه در نظر گرفته شده بودند. برای تعیین تعداد اپک<sup>۲۸</sup> بهینه، با اندازه دسته‌ی<sup>۲۹</sup> برابر ۳۲، صد تکرار روی داده‌ها انجام شده است. شکل ۶ نمودار تغییرات خط<sup>۳۰</sup> برای داده‌های آموزشی و اعتبارسنجی را نشان می‌دهد.

مدلی که روی داده‌های اعتبارسنجی کمترین خطرا را داشته است به عنوان مدل بهینه انتخاب می‌شود. میزان دقت<sup>۳۱</sup> روی داده‌های آزمون ۸۸ درصد بوده است. شکل ۷ ماتریس درهم‌ریختگی 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.

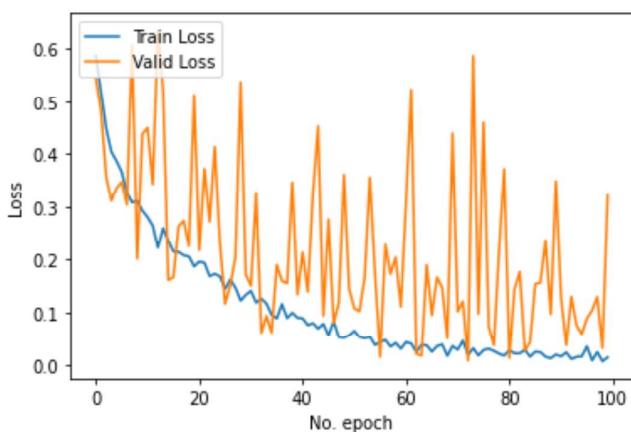
شکل ۸ نتیجه‌ی اجرای برنامه‌ی پایتون نوشته شده روی ده تصویر آزمون را نشان می‌دهد. سطر اول نمونه‌های شکسته نستعلق (شکسته) هستند که همگی به جز آخری به درستی تشخیص داده شده‌اند. سطر دوم نمونه‌های متعلق به رسم الخط ثلث هستند که اولی به اشتباه (شکسته) برچسب خورده است. مقادیر اعشاری میزان اطمینان هر مورد را نشان می‌دهد که همان خروجی آخرین لایه پیچشی است. از آنجا که فقط یک نورون در آخرین لایه پیچشی داشته‌ایم، خروجی شبکه یک عدد 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 خوراندن پارامترهای مدل پیچشی-چگال به مدل تمام پیچشی شکل ۹، مدل قابلیت عمل بر روی ورودی‌های با اندازه‌هایی فراتر از اندازه



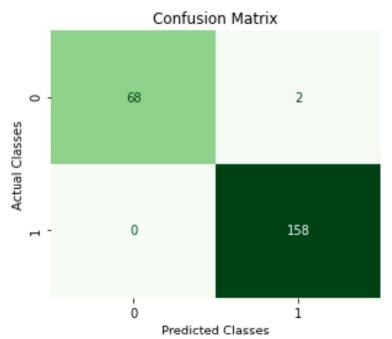
شکل ۶: نمودار خط (زیان) در ۱۰۰ تکرار روی داده‌ها

در طرح‌ها و تابلونگاره‌های خوشنویسی محبوبیت ویژه‌ای دارد. خط ثلث نیز در متون و کتبیه‌های اماکن مذهبی جایگاه خاصی داشته و دارد. در این کاربرد با یک خزنه وب، با جستجوی عبارات «شکسته‌نستعلق» و «خطاطی‌ثلث» در گوگل، حدود هزار تصویر برداشت شد. با ملاحظه بصری، تعداد زیادی از نتایج نامطلوب حذف شدند<sup>۲۵</sup> و برای هر دسته ۳۵۵ تصویر لحاظ شدند. البته حذف تصاویر با کمی اغماض انجام شد. مثلاً تعداد محدودی از تصاویر شکسته نستعلق به واقع جزو این خط محسوب نمی‌شدند اما از دیدگاه کلی شبیه به این خط بودند. برخی تابلوها دارای تصاویر پس زمینه یا انسان هم بودند که آگر نوشته مربوطه بخش اصلی تصویر را تشکیل می‌داد حذف نشدند. نمونه‌هایی از تصاویر باقیمانده در شکل ۵ 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. میانگین اندازه تصاویر ورودی  $151 \times 152$  پیکسل است. از مدل پیچشی-چگال شکل ۲ در این مثال استفاده شده است. روش کلی اجرا، به صورت انتها به انتها<sup>۲۶</sup> بوده و به جز تبدیل از بازه‌ی [۰, ۲۵۵] به بازه‌ی [۱, ۰] و داده‌افزایی<sup>۲۷</sup> پردازش دیگری روی داده‌ها



شکل ۱۰: نمودار خطا (زیان) در ۱۰۰ تکرار روی داده‌ها

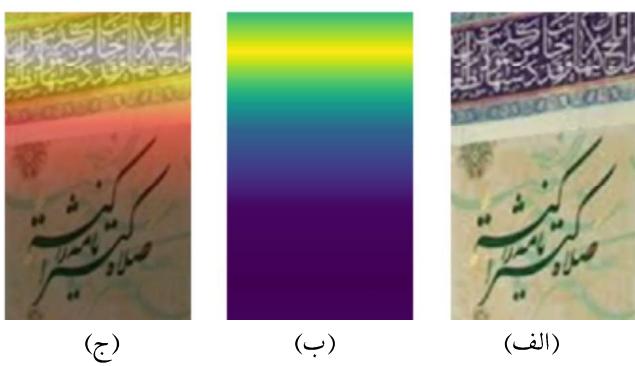


شکل ۱۱: ماتریس درهم‌ریختگی در شناسایی برگ‌های مریض

است. عموم این شیوه‌ها مبتنی بر یک مدل پیچشی-چگال هستند که بر روی تصاویر برگ‌های مجلزا آموزش دیده و قابلیت کارکرد و شناسایی بر روی تصویر یک برگ را دارند. با شیوه‌ی پیشنهادی در این نوشتر می‌توان یک مدل اینچنینی را به یک مدل تمام پیچشی تبدیل کرد که قابلیت عمل بر روی تصاویر شامل مجموعه‌ای از برگ‌ها را داشته باشد. در اینجا هدف ایجاد یک مدل دقیق‌تر یا سریع‌تر از مدل‌های موجود نیست، لذا بحث مقایسه‌ی دقت یا سرعت مطرح نیست. نکته‌ی اصلی تغییر عماری 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 مقایله در دسترس است. مجموعه دادگان مختلفی بر شناسایی امراض گیاهان موجود است. برای این کاربرد از یکی از این مجموعه دادگان عمومی<sup>۳۴</sup> 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 خواهیم کرد. به منظور عدم نیاز به دانلود کل این مجموعه داده، این زیرمجموعه در بخش دادگان گیت‌هاب این نوشتر قرار داده شده است. داده‌ها به سه گروه آموزش (۱۸۱۲ تا)، اعتبارسنجی (۲۲۶ تا) و آزمون (۲۲۸ تا) 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 درهم‌ریختگی و خروجی مدل نهایی را نشان می‌دهند.



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 ۹: خروجی بر روی تصویر شامل هر دو نوع رسم الخط

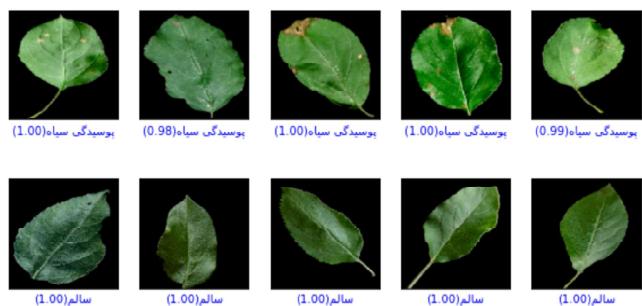
اولیه  $150 \times 150$  را پیدا می‌کند. شکل ۹ (الف) یک کتیبه‌ی دیواری که بخش پایینی آن عبارت «صلاءةً كثيراً تامةً زاكيةً» با خط شکسته علیت و بخش بالای آن کتیبه‌ای با خط ثلث است را نشان می‌دهد. شکل ۹ (ب) خروجی شبکه‌ی تمام پیچشی حاصل از تبدیل مدل پیچشی-چگال به تمام پیچشی را در قالب یک نقشه‌ی رنگی نشان می‌دهد که نقاط روش به منزله‌ی میزان تعليق بيشتر به کلاس ثلث است. همان‌گونه که دیده می‌شود با حرکت به سمت پایین، میزان تعليق به کلاس ثلث کمتر می‌شود. شکل ۹ (ج) نگاشت نقشه‌ی رنگی بر روی تصویر اولیه را نشان می‌دهد. در این بخش یک شبکه‌ی پیچشی-چگال با دقت نزدیک به ۹۰ درصد برای طبقه‌بندی تابلونگاره‌های خوشنویسی فارسی به دو کلاس شکسته نستعلیق و ثلث طراحی و آموزش داده شد. با خوراندن وزنه‌های این شبکه 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 کرد. کاربرد این تبدیل در شکل ۹ ملاحظه گردید.

### ۳-۲-شناസایی برگ‌های سالم و بیمار با مدل تمام پیچشی

کاربرد دیگری از روش پیشنهادی که در این بخش مورد بررسی قرار می‌گیرد، شناناسی برگ‌های سالم از برگ‌های با بیماری «پوسیدگی سیاه» درخت سیب است. تحقیقات متعددی در این زمینه انجام شده که در پیماری از موارد اخیر از یادگیری عمیق برای شناناسی استفاده شده است [۲۰، ۲۱، ۲۲]. در این تحقیقات عموماً با استفاده از روش انتقال یادگیری<sup>۳۵</sup> یک مدل معروف از قبل آموزش دیده انتخاب شده و بر روی یک مجموعه‌ی آموزشی از برگ‌های سالم و مریض آموزش داده شده

جدول ۴: خلاصه مدل و پارامترهای مدل پیچشی-چگال برای تصویر مجموعه داده ارقام فارسی هدی، با اندازه ورودی  $26 \times 26$ .

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
lastConv: Conv2D(10,(1,1))	(None, 1, 1, 10)	1290
Total params: 93,962		

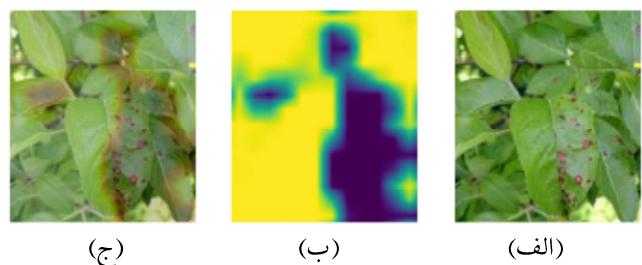


شکل ۱۲: خروجی روی ده تصویر آزمون. مقادیر اعشاری میزان اطمینان هر مورد را نشان می‌دهد.

### ۳-۳- طبقه‌بندی ارقام فارسی با مدل تمام پیچشی

مدل مورد استفاده در دو مثال قبل با آخرین لایه پیچشی با اندازه  $1 \times 1$  فقط مناسب طبقه‌بند دو کلاسه است. در این بخش نحوه کار روی مجموعه دادگان ارقام فارسی هدی [۲۳] را خواهیم دید که دارای ده طبقه می‌باشد. هدف در اینجا نمایش برتری از نظر دقت بر سایر روش‌ها نیست؛ هدف آن است که نشان داده شود چگونه یک شبکه عمیق مرسم تصاویر ورودی با اندازه‌های مختلف را تبدیل شده و تصاویر ورودی را پذیرد.

مشابه بخش پیش، ابتدا یک شبکه پیچشی عمیق معمول با لایه آخر طبقه‌بند ایجاد نموده و آموزش داده می‌شود. 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 مسطح ۱۲۸ تایی، در انتهای یک لایه چگال ده تایی عمل طبقه‌بندی را انجام می‌دهد. ۵۰۰۰ نمونه برای آموزش و اعتبارسنجی و ۶۰۰۰ نمونه برای تست انتخاب شدند. مدل ده اپک آموزش داده شد. دقت مدل بر روی داده‌های آزمون ۹۷٪ بوده است. نمودار خطای روال آموزش و جزییات کار در برنامه‌ی مربوطه در گیت‌هاب این نوشتار موجود است که از درج آنها صرفنظر می‌شود. نکته اصلی تبدیل مدل پیچشی-چگال به تمام پیچشی است. مطابق شیوه پیشنهادی خلاصه مدل پیچشی-چگال و پارامترهای آن در جدول ۴ آمده است. مدل تمام پیچشی جدیدی بر اساس وزنهای مدل آموزش دیده فرق و روش مذکور در بخش‌های قبل ایجاد می‌شود. مدل پیچشی-چگال آموزش دیده با این مدل تمام پیچشی یکی هستند، با این تفاوت که دومی با تغییر اندازه ورودی و بدون آموزش مجدد، قابلیت پذیرش ورودی‌های با اندازه‌های متفاوت را دارا می‌باشد. مدل جدید تمام پیچشی متناسب با تصویر ورودی با اندازه  $26 \times 26$  در جدول ۵ آمده است. آخرین لایه پیچشی بُعد سوم برایر با ۱۰ دارد که متناظر با تعداد کلاس‌های مورد طبقه‌بندی است. همان‌گونه که ملاحظه می‌شود برای مسائل چند کلاسه نیز شیوه‌ی مدنظر در این نوشتار قابلیت استفاده دارد.



شکل ۱۳: خروجی بر روی تصویر شامل هر دو نوع رسم الخط

دقت مدل بر روی داده‌های آزمون 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 خروجی مدل حاصله بر روی یک تصویر از برگ‌های یک درخت را نشان می‌دهد.<sup>۲۵</sup> 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. در بررسی این تصویر، هدف مشخص کردن سالم یا مريض بودن هر برگ یا هر بخش در تصویر است. درصورت استفاده از مدل‌های معمول پیچشی-چگال باید برگ‌های تصویر استخراج شده و به مدل داده شود. اما مدل تمام پیچشی می‌تواند روی هر تصویری با هر اندازه‌ای کار کند؛ با این فرض که اندازه‌ی برگ‌ها در تصویر در حدود اندازه‌ی آنها در تصاویر آموزشی باشد، مدل جدید می‌تواند نتایج خوبی را به دست بدهد. در این مثال برگ‌های مريض گروه با برچسب صفر و برگ‌های سالم گروه یک بوده‌اند. تصویر ۱۳ (ب) خروجی مدل تمام پیچشی را نشان می‌دهد. نواحی روشن تر تعلق بیشتری به گروه یک (برگ‌های سالم) و نواحی تیره‌تر تعلق بیشتری به گروه صفر (برگ‌های بیمار) دارند. نگاشت این نقشه بر روی تصویر اصلی در شکل ۱۳ (ج) نمایش داده شده است. همان‌گونه که ملاحظه می‌شود تا حدود زیادی نواحی تیره‌تر مشخص کننده برگ‌های بیمار هستند. از آنجا که در این تصویر برگ‌ها با زوایای مختلف بوده‌اند و تصاویر آموزشی همه از یک منظر خاص تهیه شده‌اند، در روال داده‌افزایی برای آموزش شبکه، تبدیل‌های خاصی (همچون دوران و انعکاس) بر روی تصاویر آموزشی اعمال شده است.

## ۴- جمع‌بندی

Layer (type)	Output Shape	Param #
Conv1: Conv2D(32, (3,3))	(None, 158, 266, 32)	320
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
lastConv: Conv2D(10, (1,1))	(None, 18, 31, 10)	1290
Total params:	93,962	

مدل پیچشی-چگال مناسب، با هر دقتی که وجود داشته باشد، با روش ذکر شده در این نوشتار، می‌تواند برای کاربردهای با اندازه‌ی ورودی متغیر به یک مدل تمام پیچشی تبدیل شود.

## مراجع

- [1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
- [2] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” *IEEE Access*, vol.7, pp.128837–128868, 2019.
- [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 R-CNN,” *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.

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 لایه‌ی (های) طبقه‌بند هستند. در این مقاله این مدل‌های مرسوم با عنوان پیچشی-چگال نام برد شدند. در این نوشتار نحوه‌ی تبدیل یک شبکه‌ی پیچشی-چگال به شبکه‌ی تمام 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 طبقه‌بند لایه‌ی آخر با یک لایه‌ی پیچشی عوض شده است جایگزین می‌شود. مدل جدید تمام پیچشی با وزن‌های قبلی آموزش دیده کار می‌کند. نحوه‌ی استفاده از وزن‌های لایه‌ی چگال در مدل جدید در نوشتار توضیح داده شده و در برنامه‌ی پایتون مرتبط با این نوشتار به صورت منبع باز 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 برد شود. مجموعه داده‌های برچسب زده‌ی بسیاری موجود هستند که قابلیت استفاده از شبکه‌های جدید یادگیری عمیق را دارند، اما همچون روش‌های قلیمی‌تر حوزه‌ی شناسایی الگو، مدل آماده شده فقط امکان کار با ورودی‌های با اندازه‌ی ثابت (هم‌اندازه با نمونه‌های آموزشی) را دارد. از طرفی برخی از مدل‌های جدید حوزه‌ی یادگیری عمیق نیازمند گذاری شده باشند. 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 ورودی‌های با هر اندازه‌ای قابلیت عملکرد را داشته باشد و بتواند اشیاء هدف را در تصویر ورودی مشخص کند. در واقع این شیوه امکان استفاده از ابزارهای جدیدتر با داده‌ها و مدل‌های مرسوم گذشته را فراهم می‌کند. دست‌آوردهای 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
- عدم نیاز به علامت‌گذاری اشیاء هدف در تصاویر
- استفاده از شبکه‌های عصبی پیچشی برای طبقه‌بندی و شناسایی دو رسم الخط فارسی
- استفاده از روش پیشنهادی برای تفکیک برگ‌های بیمار از سالم در تصاویر دارای چندین برگ.
- نمایش امکان تبدیل به مدل تمام پیچشی بر روی مسئله‌ی طبقه‌بندی و شناسایی ارقام فارسی به عنوان یک مسئله‌ی دارای چندین دسته.
- روش پیشنهادی برای ایجاد یک مدل یادگیری عمیق از پایه، مناسب است و بسته به کاربرد، مدل نهایی می‌تواند بسیار کم حجم بوده و زمان آموزش زیادی لازم نداشته باشد. از آنجا که هدف این نوشتار افزایش دقت طبقه‌بند نبوده است، در خصوص کاربردهای ذکر شده، مقایسه‌ای با روش‌های قبلی صورت نگرفته است؛ هر روش یادگیری عمیق مبتنی بر یک

- [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.

## پاورقی‌ها

- <sup>۱</sup>Convolutional Neural Networks (CNNs/ConvNets)
- <sup>۲</sup>Annotated
- <sup>۳</sup>[https://cs.nyu.edu/~silberman/datasets/nyu\\_depth\\_v2.html](https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html)
- <sup>۴</sup><https://github.com/shelhamer/fcn.berkeleyvision.org>
- <sup>۵</sup><http://www.image-net.org/>
- <sup>۶</sup>Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL)
- <sup>۷</sup>Visual Object Classes (VOC) <http://host.robots.ox.ac.uk/pascal/VOC/index.html>
- <sup>۸</sup><https://cs.stanford.edu/~roozbeh/pascal-context/>
- <sup>۹</sup><https://pjreddie.com/darknet/>
- <sup>۱۰</sup>Weakly Supervised Object Detection (WSOD)
- <sup>۱۱</sup>Build from scratch
- <sup>۱۲</sup>Fully Connected or Dense
- <sup>۱۳</sup>این کار در مسابقه شناسایی درختان ۲۷مین کنفرانس مهندسی برق ایران توسط نگارنده انجام شده است: <https://github.com/mamintoosi/tree-detection>
- <sup>۱۴</sup>MaxPooling
- <sup>۱۵</sup>Flatten
- <sup>۱۶</sup>Dense
- <sup>۱۷</sup>SoftMax
- <sup>۱۸</sup>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 تاچار از عبارت «پیچشی-چگال» استفاده شد.
- <sup>۱۹</sup>Fully Convolution
- <sup>۲۰</sup>ReLU
- <sup>۲۱</sup>Keras, Tensorflow
- <sup>۲۲</sup><https://github.com/mamintoosi/FC2FC>
- <sup>۲۳</sup>به عنوان نمونه به تحقیقات احسان‌الله کبیر مراجعه شود: <https://www.modares.ac.ir/~kabir>
- <sup>۲۴</sup>Automatic image annotation
- <sup>۲۵</sup>نگارنده دارای مدرک خوش از انجمن خوشنویسان ایران می‌باشد.
- <sup>۲۶</sup>end-to-end
- <sup>۲۷</sup>Data Augmentation
- <sup>۲۸</sup>Epoch
- <sup>۲۹</sup>Batch Size
- <sup>۳۰</sup>Loss
- <sup>۳۱</sup>Accuracy
- <sup>۳۲</sup>Apple Black rot
- <sup>۳۳</sup>Transfer Learning
- <sup>۳۴</sup>[https://github.com/digitalepidemiologylab/plantvillage\\_deeplearning\\_paper\\_dataset](https://github.com/digitalepidemiologylab/plantvillage_deeplearning_paper_dataset)
- <sup>۳۵</sup>تصویر برگ‌های درخت از دانشگاه ایالتی میشیگان اخذ شده است: [https://www.canr.msu.edu/ipm/diseases/black\\_rot\\_blossom\\_end\\_rot\\_frogeye\\_leaf\\_spot](https://www.canr.msu.edu/ipm/diseases/black_rot_blossom_end_rot_frogeye_leaf_spot)

- [10] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image 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. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [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.
- [۱۷] محمد صبری، محمد شهرام معین، و فرید رزا زادی، ”ارائه یک روش ترتیبی پویا بر اساس یادگیری عمیق به منظور بهبود کارایی سیستم‌های تطبیق بیومتریکی 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. Sheykhanvand, S. Meshgini, and Z. Mousavi, “Automatic detection of various epileptic seizures from eeg signal using deep learning 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.