



دانشکده مهندسی کامپیوتر  
گروه هوش مصنوعی و رباتیکز  
درس شبکه‌های عصبی  
تیر ۱۴۰۱

# شبکه SqueezeNet

(کار مطالعاتی درس شبکه‌های عصبی)

سبحان مرادیان دقیق<sup>۱</sup>

<sup>۱</sup>دانشجو، گروه هوش مصنوعی و رباتیکز، دانشگاه صنعتی امیرکبیر، [s.mdaghigh@aut.ac.ir](mailto:s.mdaghigh@aut.ac.ir)

۱۴۰۱/۴/۱۴

## چکیده

شبکه SqueezeNet، یک شبکه عصبی پیچشی<sup>۱</sup> است که از راهبردهای طراحی، برای کاهش تعداد پارامترها استفاده می‌کند. به طور کلی، با فرض وجود صحت<sup>۲</sup> یکسان، CNN ها با معماری‌های کوچکتر یا به عبارتی دیگر تعداد پارامترهای کمتر، دارای حداقل سه مزیت هستند: [۱]

- CNN های کوچکتر به ارتباط کمتری در بین سرورها در طول آموزش توزیع شده نیاز دارند.
- CNN های کوچکتر به پهنای باند کمتری نیاز دارند تا یک مدل جدید را از فضای ابری به یک ماشین خودران صادر کنند.
- CNN های کوچکتر برای قرارگیری روی FPGA و سایر سخت افزارها با حافظه محدود امکان پذیرتر هستند.

بنابراین سعی شده است تا ضمن حفظ صحت، پارامترهای شبکه‌های بزرگ را کاهش داد. در این گزارش، ابتدا شبکه SqueezeNet را به طور جامع معرفی می‌کنیم و سپس به بررسی مقالاتی می‌پردازیم که در سال‌های اخیر، توانسته‌اند این شبکه را بهبود دهند.

واژه‌های کلیدی: شبکه SqueezeNet، شبکه عصبی پیچشی

---

<sup>۱</sup> Convolutional Neural Network (CNN)

<sup>۲</sup> Accuracy

## فهرست عناوین

۱	فصل اول: مقدمه .....
۳	فصل دوم: بررسی شبکه SqueezeNet .....
۳	۱-۲ راهبردهای طراحی معماری .....
۴	۲-۲ ماژول آتش .....
۵	۳-۲ معماری SqueezeNet .....
۶	۴-۲ ارزیابی شبکه SqueezeNet .....
۷	فصل سوم: بهبود معماری SqueezeNet با هزینه محاسباتی ناچیز .....
۷	۱-۳ راهکار ارائه شده .....
۷	۱-۱-۳ مرحله اول .....
۷	۱-۱-۱-۳ افزایش داده .....
۸	۲-۱-۱-۳ کاهش نرخ یادگیری .....
۸	۲-۱-۳ مرحله دوم .....
۸	۱-۲-۱-۳ استفاده از PReLU .....
۸	۲-۲-۱-۳ نرمال سازی دسته‌ای .....
۹	۲-۳ ارزیابی مدل بهبود یافته .....
۱۰	فصل چهارم: بهبود معماری SqueezeNet برای سیستم‌های پیشنهادگر .....
۱۱	۱-۴ بیان مشکل .....
۱۱	۲-۴ بهبود مدل SqueezeNet .....
۱۲	۱-۲-۴ تنظیم هسته‌های پیچشی ماژول آتش .....
۱۲	۲-۲-۴ تنظیم ساختار ادغام .....
۱۳	۳-۴ ارزیابی مدل بهبود یافته SqueezeNet .....
۱۵	فصل پنجم: شبکه Faster SqueezeNet .....
۱۵	۱-۵ معرفی شبکه FS .....
۱۶	۱-۱-۵ تقلید از اتصالات شبکه DenseNet .....
۱۶	۲-۱-۵ تقلید از بلوک‌های ساختمانی شبکه ResNet .....

۱۷	..... ۲-۵ ارزیابی مدل بهبود یافته FS
۱۸	..... فصل ششم: شبکه <b>Weight-Quantized SqueezeNet</b>
۱۸	..... ۱-۶ فن چندی سازی وزن
۱۸	..... ۲-۶ دادگان جمع آوری شده ی مقاله
۱۹	..... ۳-۶ مقایسه و ارزیابی مدل های مختلف و مدل پیشنهاد شده
۲۱	..... فصل هفتم: نتیجه گیری و پیشنهادها
۲۲	..... منابع و مراجع

## فهرست اشکال

- شکل ۱-۲ سازماندهی پالایه‌های پیچشی در ماژول آتش ..... ۴
- شکل ۲-۲ معماری شبکه عصبی پیچشی SqueezeNet ..... ۵
- شکل ۱-۴ چند نمونه از تصاویر شلوار در دادگان مقاله ..... ۱۲
- شکل ۲-۴ شکل سمت چپ، ماژول آتش اصلی، سمت راست، ماژول آتش بعد از اعمال تغییرات ..... ۱۳
- شکل ۱-۵ تصاویری از دادگان معرفی شده، به ترتیب از سمت راست، مقاومت، سلف، خازن، برد الکتریکی ... ۱۲
- شکل ۲-۵ معماری شبکه FS ..... ۱۶
- شکل ۱-۶ تعدادی از تصاویر دادگان ..... ۱۹

## فهرست جداول

جدول ۱-۱	مقالاتی که در فصول مختلف مورد بررسی قرار می‌گیرند	۲
جدول ۱-۲	مقایسه SqueezeNet با راهکارهای فشرده‌سازی مدل	۶
جدول ۱-۳	صحت مدل اصلی و مدل بعد از اعمال راهبردهای افزایش داده و کاهش نرخ یادگیری	۹
جدول ۲-۳	صحت مدل بهبود یافته در جدول ۱-۳ و صحت بعد از اعمال نرمال‌سازی دسته‌ای	۹
جدول ۳-۳	صحت مدل بهبود یافته در جدول ۲-۳ و صحت بعد از اعمال PReLU	۱۰
جدول ۱-۴	نتایج حاصل از اعمال پالایه‌هایی با اندازه‌های مختلف در مازول آتش	۱۳
جدول ۲-۴	نتایج حاصل از اعمال لایه‌های ادغام	۱۴
جدول ۱-۵	حساسیت و زمان اجرا مدل‌های مختلف روی دادگان معرفی شده‌ی مقاله	۱۷
جدول ۱-۶	اندازه و صحت مدل‌های مختلف روی دادگان تهیه شده قبل از چندی‌سازی وزن‌ها	۱۹
جدول ۲-۶	اندازه و صحت مدل‌های مختلف روی دادگان تهیه شده بعد از چندی‌سازی وزن‌ها	۲۰

# فصل اول

## مقدمه

در حدود سال‌های ۲۰۱۲ تا ۲۰۱۶، بیشتر تمرکز تحقیقات روی CNN ها به افزایش صحت در مجموعه دادگان بینایی رایانه مانند دادگان ImageNet متمرکز شده بود. برای یک سطح صحت مشخص، معمولاً چندین معماری CNN وجود دارد که به آن سطح صحت دست می‌یابد. اما یکی از تفاوت‌هایی که بین این معماری‌ها وجود دارد، اندازه آن معماری است.<sup>۳</sup> از بین این معماری‌ها، یک معماری CNN با پارامترهای کمتر مزیت‌هایی به همراه دارد که به اختصار به آن‌ها اشاره می‌کنیم: [۱]

### - آموزش توزیع شده‌ی کارآمدتر

ارتباط بین سرورها عاملی است که مقیاس پذیری آموزش توزیع شده CNN را محدود می‌کند. برای آموزش موازی داده‌های توزیع شده، سربار ارتباط، مستقیماً با تعداد پارامترهای مدل متناسب است. [۲] به عبارتی دیگر، مدل‌های کوچک به دلیل نیاز به ارتباط کمتر، سریع‌تر آموزش می‌بینند.

### - به‌روزرسانی راحت‌تر مدل‌های ارائه شده به مشتریان

برای ماشین‌ها خودران، شرکت‌های خودروسازی به طور دوره‌ای مدل‌های جدید را از سرورهای خود به خودروهای مشتریان کپی می‌کنند. مدل‌های کوچک‌تر به ارتباطات کمتری نیاز دارند و در نتیجه به‌روزرسانی‌های مکرر را امکان‌پذیرتر می‌کند.

### - امکان ذخیره‌سازی بر روی تراشه‌های FPGA و تراشه‌هایی با حافظه محدود

تراشه‌های FPGA اغلب دارای حافظه‌ای کمتر از ۱۰ مگابایت هستند و هیچ حافظه خارجی ندارند. بنابراین برای اینکه بتوان مدل مورد نظر را بر روی این تراشه‌ها و به طور کلی تراشه‌هایی با میزان حافظه محدود ذخیره‌سازی کرد، باید مدل به اندازه کافی کوچک باشد.

همانطور که دیدیم، یک معماری CNN کوچک‌تر کارایی بهتری دارد. در نتیجه شبکه SqueezeNet معرفی شد و هدف آن کم کردن پارامترهای مدل، ضمن حفظ صحت اولیه بود. البته نکته قابل توجه آن که قبل از معرفی

---

<sup>۳</sup> منظور از اندازه معماری، میزان فضای ذخیره‌سازی برای پارامترهای آن معماری است. به طور مثال، معماری AlexNet که دارای ۶۰ میلیون پارامتر است، به ۲۴۰ مگابایت فضای ذخیره‌سازی نیاز دارد.

شبکه SqueezeNet، راهکارهای دیگری معرفی شده بودند که فشرده‌ساز نام داشتند؛ یعنی سعی می‌کردند به روش‌های مختلفی، مدل را فشرده‌تر کنند. در فصل دوم، مقایسه‌ای از عملکرد این راهکارها با راهکار شبکه SqueezeNet خواهیم دید.

در ادامه طی چند فصل به بررسی پیشرفت‌ها و بهبودهایی که در سال‌های اخیر بر روی معماری SqueezeNet بوده است، می‌پردازیم.

مقالات جدول ۱-۱ به ترتیب در ۶ فصل بررسی می‌شوند:

صفحات	سال	دسترسی	مقاله	شماره
۱۳	۲۰۱۶	<a href="#">پیوند</a>	SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size	۱
۶	۲۰۲۰	<a href="#">پیوند</a>	Improving the accuracy of SqueezeNet with negligible extra computational cost	۲
۷	۲۰۲۰	<a href="#">پیوند</a>	E-commerce Item Identification Based on Improved SqueezeNet	۳
۱۱	۲۰۲۰	<a href="#">پیوند</a>	An Electronic Component Recognition Algorithm Based on Deep Learning with a Faster SqueezeNet	۴
۱۴	۲۰۲۲	<a href="#">پیوند</a>	Weight-Quantized SqueezeNet for Resource-Constrained Robot Vacuums for Indoor Obstacle Classification	۵
۵۱	-	-	-	جمع

جدول ۱-۱ مقالاتی که در فصول مختلف مورد بررسی قرار می‌گیرند



## فصل دوم

### بررسی شبکه SqueezeNet

در این فصل، معماری شبکه SqueezeNet را از روی مقاله اصلی آن بررسی می‌کنیم.

#### ۱-۲: راهبردهای طراحی معماری

راهبردهای اول و دوم به منظور کاهش منطقی پارامترها در یک CNN به کار می‌روند و در ضمن تلاش می‌کنند که صحت را حفظ کنند. راهبرد سوم به منظور به حداکثر رساندن صحت در تعداد محدود پارامترهای باقی‌مانده است.

- **راهبرد اول:** جایگذاری پالایه  $3 \times 3 \times 4$  با پالایه  $1 \times 1$   
اکثر پالایه‌ها را  $1 \times 1$  می‌گذاریم، زیرا یک پالایه  $1 \times 1$ ، به میزان ۹ برابر پارامترهای کمتری نسبت به پالایه  $3 \times 3$  دارد.
- **راهبرد دوم:** کاهش تعداد کانال‌های ورودی به پالایه‌های  $3 \times 3$   
در یک لایه پیچشی که از پالایه‌های  $3 \times 3$  تشکیل شده، تعداد کل پارامترها از رابطه زیر بدست می‌آید:  
$$(1-2) \quad (\text{تعداد کانال‌های ورودی}) * (\text{تعداد پالایه‌ها}) * (3 \times 3)$$

بنابراین، برای کم کردن پارامترها در یک CNN، نه تنها راهبرد اول یعنی کاهش تعداد پالایه‌های  $3 \times 3$  مهم است، بلکه کاهش تعداد کانال‌های ورودی به پالایه‌های  $3 \times 3$  نیز مهم است. با استفاده از لایه‌های فشرده‌ساز (Squeeze layers) تعداد کانال‌های ورودی را به پالایه‌های  $3 \times 3$  کاهش می‌دهیم.

---

<sup>4</sup> Filter

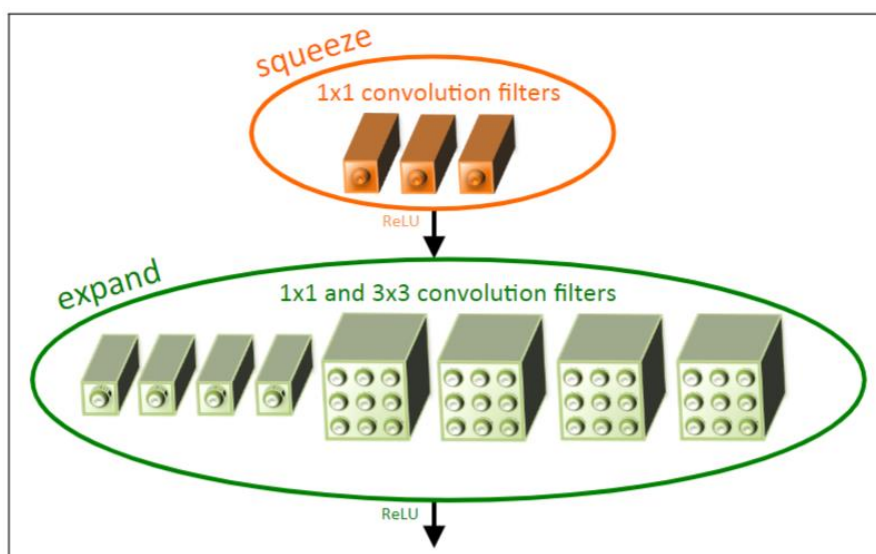
- راهبرد سوم: به تاخیر انداختن عملیات فرونمونه‌برداری<sup>۵</sup>

اثبات می‌شود که با به تاخیر انداختن فرونمونه‌برداری، اندازه نگاشت‌های فعال‌سازی<sup>۶</sup> بزرگ‌تر شده و در نتیجه می‌توانند به صحت طبقه‌بندی بالاتری رسید. در این راهبرد سعی شده است عملیات فرونمونه‌برداری در ابتدای معماری استفاده نشود و در لایه‌های آخر استفاده شود. زیرا اگر در ابتدای معماری این عمل رخ دهد، حجم زیادی از داده‌ها خیلی سریع حذف می‌شوند.

## ۲-۲: ماژول آتش<sup>۷</sup>

در این بخش، ماژول آتش را توصیف می‌کنیم که واحدهای ساختاری برای معماری CNN است. این ماژول امکان استفاده از راهبردهای اول و دوم و سوم را می‌دهد. یک ماژول آتش شامل دو لایه زیر است:

- یک لایه پیچشی فشرده که فقط پالایه‌های  $1 \times 1$  دارد (بخش نارنجی رنگ در شکل ۱-۲)
- خروجی بخش قبل به یک لایه گسترش<sup>۸</sup> وارد می‌شود که ترکیبی از پالایه‌های پیچشی  $1 \times 1$  و  $3 \times 3$  است. (بخش سبز رنگ در شکل ۱-۲)



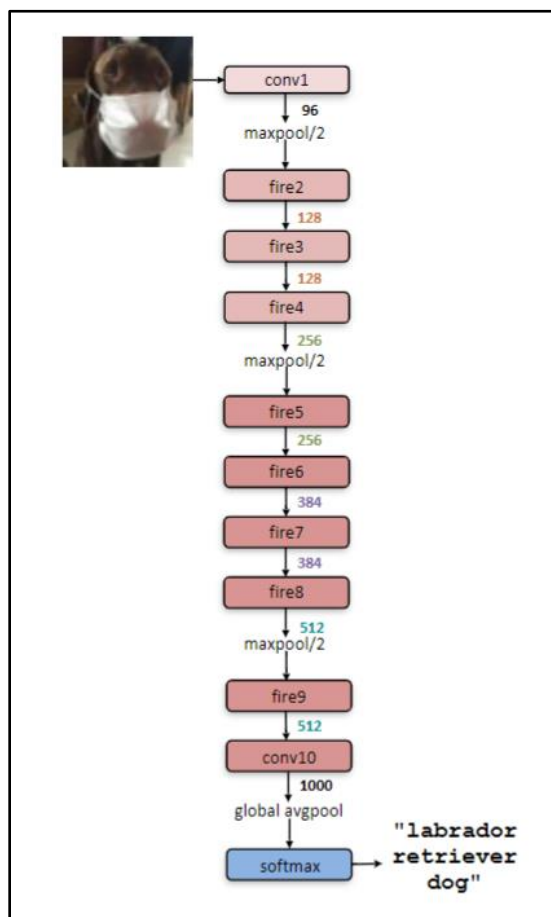
شکل ۱-۲ سازماندهی پالایه‌های پیچشی در ماژول آتش

<sup>5</sup> Downsampling  
<sup>6</sup> Activation Maps  
<sup>7</sup> Fire  
<sup>8</sup> Expand layer

دلیل اینکه در ماژول آتش از پالایه‌های  $1 \times 1$  استفاده شده است، کاربرد راهبرد اول از بخش ۱-۲ است. ابرپارامترهای ماژول آتش به صورت روبه‌رو تعریف می‌کنیم:  $S_{1 \times 1}$ ،  $e_{1 \times 1}$  و  $e_{3 \times 3}$ . که به ترتیب، اولی تعداد پالایه‌های لایه فشرده، دومی تعداد پالایه‌های  $1 \times 1$  در لایه گسترش و سومی تعداد پالایه‌های  $3 \times 3$  در لایه گسترش است. حال اگر هنگام تنظیم این ابرپارامترها، پارامتر  $S_{1 \times 1}$  را کمتر از  $(e_{3 \times 3} + e_{1 \times 1})$  تنظیم کنیم، طبق راهبرد دوم از بخش ۱-۲، لایه فشرده به محدود کردن تعداد کانال‌های ورودی به پالایه‌های  $3 \times 3$  کمک می‌کند.

## ۳-۲: معماری SqueezeNet

پس از معرفی راهبردهای طراحی معماری و ماژول آتش در دو بخش قبل، حال در این بخش معماری شبکه عصبی پیچشی SqueezeNet را شرح می‌دهیم. همانطور که در شکل ۲-۲ می‌بینید، این معماری با یک لایه پیچشی مستقل (conv1) شروع می‌شود، به دنبال آن ۸ ماژول آتش (fire2-9) قرار دارد و در نهایت یک لایه پیچشی (conv10) قرار می‌گیرد. نکته قابل توجه این است که به تدریج تعداد پالایه‌های هر ماژول آتش از ابتدا تا انتهای شبکه افزایش داده می‌شود.



شکل ۲-۲ معماری شبکه عصبی پیچشی SqueezeNet

## ۴-۲: ارزیابی شبکه SqueezeNet

شبکه طراحی شده، مبتنی بر شبکه AlexNet است. به بیانی دیگر، شبکه SqueezeNet سعی کرده است که پارامترهای شبکه AlexNet را کاهش دهد. بنابراین در این بخش به مقایسه بهبود حاصل شده برای اندازه مدل توسط شبکه SqueezeNet و راهکارهای دیگری که با عنوان فشرده‌ساز مدل قبل از این شبکه معرفی شده بودند، می‌پردازیم. جدول ۱-۲ این مقایسه را به خوبی نشان می‌دهد.

معماری CNN	راهکار فشرده‌سازی	اندازه مدل جدید	کاهش اندازه مدل جدید در مقایسه با AlexNet	صحت Top-1 بر روی ImageNet	صحت Top-5 بر روی ImageNet
AlexNet	بدون فشرده‌سازی	۲۴۰ مگابایت	۱ برابر	۵۷/۲٪	۸۰/۳٪
AlexNet	SVD [3]	۴۸ مگابایت	۵ برابر	۵۶/۰٪	۷۹/۴٪
AlexNet	Network Pruning [4]	۲۷ مگابایت	۹ برابر	۵۷/۲٪	۸۰/۳٪
AlexNet	Deep Compression [5]	۶/۹ مگابایت	۳۵ برابر	۵۷/۲٪	۸۰/۳٪
SqueezeNet	بدون فشرده‌سازی	۴/۸ مگابایت	۵۰ برابر	۵۷/۵٪	۸۰/۳٪

جدول ۱-۲ مقایسه SqueezeNet با راهکارهای فشرده‌سازی مدل

## فصل سوم

### بهبود معماری SqueezeNet با هزینه محاسباتی ناچیز [۶]

در این فصل، به مقاله‌ای می‌پردازیم که در سال ۲۰۲۰ منتشر شد و تلاش کرد تا بهبودهایی هم در روش آموزش مدل و هم در ریزمعماری شبکه SqueezeNet ایجاد کند و در نتیجه صحت این شبکه را برای صحت Top-1 و Top-5 به ترتیب به میزان ۷/۰۵٪ و ۴/۷۹٪ افزایش دهد. البته این میزان افزایش صحت، بدون هزینه نبوده و باعث شده است که تعداد پارامترهای این شبکه ۰/۴۸٪ افزایش پیدا کند. این مقدار افزایش تعداد پارامترها، در مقایسه با مقدار بهبودی که در صحت مدل داده شده، قابل قبول است.

#### ۱-۳: راهکار ارائه شده

در این مقاله، بهبود انجام شده طی دو مرحله اتفاق می‌افتد که به شرح زیر است:

#### ۱-۱-۳: مرحله اول

صحت مدل پایه (SqueezeNet) را با استفاده از راهبردهای زیر در بخش آموزش مدل بهبود داده است.

#### ۱-۱-۱-۳: افزایش داده<sup>۹</sup>

یادگیری عمیق یک فرایند داده محور است. از تقویت یا افزایش داده‌ها می‌توان برای گسترش موثر داده‌ها و انجام برخی عملیات عادی سازی داده‌ها استفاده کرد، به طوری که مجموعه داده‌ها را گسترش داد، تنوع داده‌ها را غنی کرد، ویژگی‌های عمیق‌تر و گسترده‌تر را یاد گرفت و از بیش‌برازش و کم‌برازش مدل‌ها جلوگیری کرد. در این بخش، از دو نوع روش افزایش داده استفاده کرده است، برش<sup>۱۰</sup> با تغییر اندازه تصادفی و چرخش افقی تصادفی که به بهبود صحت کمک می‌کنند.

---

<sup>۹</sup> Data Augmentation

<sup>۱۰</sup> Crop

### ۳-۱-۱-۲: کاهش نرخ یادگیری

هنگام آموزش یک مدل، اغلب توصیه می‌شود که میزان یادگیری را در طول پیشرفت آموزش کاهش بدهیم. بنابراین یک تابع کاهنده کسینوسی<sup>۱۱</sup> برای نرخ یادگیری اولیه ارائه شده، اعمال شده است.

### ۳-۱-۲: مرحله دوم

در این مرحله هم دو راهبرد زیر را ارائه داده است. با تغییر تابع فعال‌سازی از واحد خطی اصلاح شده<sup>۱۲</sup> به یکی از دو حالت واحد خطی اصلاح‌شده پارامتری<sup>۱۳</sup> یا فعال‌سازی خطی و همچنین افزودن لایه‌های نرمال‌سازی دسته‌ای<sup>۱۴</sup> قبل از لایه‌های فعال‌سازی، در برابر از دست رفتن اطلاعات به دلیل فشردگی مدل و پدیده نوروپ‌های مرده که به دلیل استفاده از ReLU به وجود می‌آید، جلوگیری کرده است.

### ۳-۱-۲-۱: استفاده از PReLU

به دلیل پارامترهای قابل آموزش، PReLU می‌تواند در برابر از دست دادن اطلاعات به دلیل فشردگی پس از پیچش ۱\*۱ تا حد زیادی جلوگیری کند و پدیده نوروپ‌های مرده یا ReLU مرده را کاهش دهد. در نتیجه این تغییر، صحت SqueezeNet به طور قابل توجهی بهبود یافت که در بخش ارزیابی خواهیم دید.

### ۳-۲-۱-۲: نرمال‌سازی دسته‌ای

نرمال‌سازی دسته‌ای یک فرآیند جدید برای تسریع همگرایی شبکه‌های عمیق است. نرمال‌سازی دسته‌ای فقط دو پارامتر به ازای هر فعال‌سازی اضافه می‌کند. فرمول نرمال‌سازی دسته‌ای به صورت زیر است:

$$\hat{x} = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \quad (۱-۳)$$

که  $\epsilon$  از تقسیم بر صفر شدن، اجتناب می‌کند. نرمال‌سازی دسته‌ای نرخ یادگیری بالاتری را امکان‌پذیر می‌کند و همچنین مدل را منظم می‌کند.

---

<sup>11</sup> Cosine Decay

<sup>12</sup> Rectified Linear Unit (ReLU)

<sup>13</sup> Parametric Rectified Linear Unit (PReLU)

<sup>14</sup> Batch Normalization

نرمال سازی دسته‌ای، که می‌تواند به عنوان نوعی منظم‌سازی<sup>۱۵</sup> مشابه روش حذف تصادفی<sup>۱۶</sup> در نظر گرفته شود، از بیش‌برازش<sup>۱۷</sup> جلوگیری می‌کند و تعمیم‌پذیری<sup>۱۸</sup> مدل را افزایش می‌دهد. استفاده از لایه‌های نرمال سازی دسته‌ای به ما امکان می‌دهد از تعداد دوره<sup>۱۹</sup> کمتری برای دستیابی به صحت یکسان و در نهایت نتایج بهتر استفاده کنیم.

### ۲-۳: ارزیابی مدل بهبود یافته

در جدول ۱-۳ مقایسه‌ای بین صحت مدل اصلی، صحت مدل بعد از اعمال راهبرد افزایش داده (بخش ۱-۱-۳-۱) و در نهایت صحت مدل بعد از اعمال هر دو راهبرد معرفی شده در بخش ۱-۱-۳ می‌باشد. این مقایسه روی نسخه‌ی SqueezeNet v1.1 می‌باشد. در جدول ۲-۳ مقایسه‌ای بین صحت مدل بهبود یافته در جدول ۱-۳ و صحت مدل بعد از اعمال نرمال سازی دسته‌ای، می‌باشد.

معماری CNN	راهبرد	صحت Top-1	صحت Top-5
SqueezeNet v1.1	اصلی	% ۵۷/۱۰	% ۸۰/۳۰
	افزایش داده	% ۵۷/۹۳	% ۸۰/۳۲
	افزایش داده و استفاده از تابع کاهنده نرخ یادگیری کسینوسی	% ۶۰/۰۷	% ۸۲/۱۵

جدول ۱-۳ صحت مدل اصلی و مدل بعد از اعمال راهبردهای افزایش داده و کاهش نرخ یادگیری

معماری CNN	راهبرد نرمال سازی دسته‌ای	صحت Top-1	صحت Top-5	تعداد پارامترها
SqueezeNet v1.1	بدون نرمال سازی دسته‌ای	% ۶۰/۰۷	% ۸۲/۱۵	۱۲۳۵۴۹۶
	با نرمال سازی دسته‌ای	% ۶۱/۳۷	% ۸۳/۴۹	۱۲۴۱۳۸۴

جدول ۲-۳ صحت مدل بهبود یافته در جدول ۱-۳ و صحت مدل بعد از اعمال نرمال سازی دسته‌ای

<sup>15</sup> Regularization

<sup>16</sup> Dropout

<sup>17</sup> Overfitting

<sup>18</sup> Generalization

<sup>19</sup> Epoch

تعداد پارامترها	صحت Top-5	صحت Top-1	راهبرد	معماری CNN
۱۲۴۱۳۸۴	% ۸۳/۴۹	% ۶۱/۳۷	ReLU	SqueezeNet v1.1
۱۲۴۱۴۰۹	% ۸۴/۹۸	% ۶۴/۰۸	PReLU	

جدول ۳-۳ صحت مدل بهبود یافته در جدول ۲-۳ و صحت بعد از اعمال PReLU

مقاله‌ی بررسی شده، روشی را برای بهبود صحت SqueezeNet با یک هزینه محاسباتی اضافی قابل قبول ارائه کرد. آزمایش‌های انجام شده بر روی مجموعه داده‌های ImageNet نشان داده است که روش پیشنهادی، صحت Top-1 و صحت Top-5 را برای SqueezeNet، بدون نیاز به هیچ زمان بیشتری، افزایش داده است. از نظر تئوری، همچنین نشان داده است که نرمال‌سازی دسته‌ای می‌تواند از بیش‌برازش جلوگیری کند و فعال‌سازی خطی یا فعال‌سازی PReLU، از دست دادن اطلاعات را کاهش می‌دهد. این دو روش به طور قابل توجهی قابلیت استفاده از شبکه‌های سبک وزن را بهبود بخشیدند.



## فصل چهارم

### بهبود معماری SqueezeNet برای تشخیص تصاویر محصولات

#### در سیستم‌های پیشنهادگر [۷]

در این فصل، به مقاله‌ای می‌پردازیم که در سال ۲۰۲۰ منتشر شد و تلاش کرد تا به منظور بهبود نرخ تشخیص تصاویر محصول در سیستم‌های پیشنهادگر، یک شبکه عصبی پیچشی SqueezeNet بهبود یافته با کارایی بالا پیشنهاد کند. در واقع با ایجاد تغییراتی منطقی در شبکه عصبی پیچشی SqueezeNet، توانست صحت تشخیص تصاویر در این سیستم‌ها را به میزان یک درصد افزایش دهد.

#### ۴-۱: بیان مشکل

شبکه SqueezeNet نرخ تشخیص کمی برای بخش‌های تصویر دارد. دلیل آن این است که به دلیل تفاوت‌های زیاد محصولات در طبقه‌بندی یکسان، ساختار ماژول آتش در SqueezeNet و تنظیم جایگاه لایه‌های ادغام<sup>۲۰</sup> نمی‌توانند به طور موثر ویژگی‌های یکپارچه‌ای را استخراج کنند، و در نتیجه نتایج پیشنهاد شده، ضعیف هستند.

بنابراین در این مقاله سعی شده است که ابرپارامترهایی مانند تعداد ماژول‌های آتش و اندازه هسته پیچشی را بهینه کند و پس از هر ماژول آتش یک لایه ادغام اضافه کند تا شبکه بتواند نرخ تشخیص را افزایش دهد و سرعت همگرایی سریع‌تری برای شناسایی داشته باشد.

#### ۴-۲: بهبود مدل SqueezeNet

مدل بهبود یافته در این مقاله همانطور که توضیح داده شد، با اندازه هسته پیچشی در ساختار ماژول آتش و تنظیم لایه ادغام در شبکه شروع می‌شود. ساختار معقول ماژول آتش می‌تواند به طور کامل ویژگی‌های تصویر را استخراج کند. روش ادغام چند لایه<sup>۲۱</sup> به پالایش بیشتر ویژگی‌های کلیدی کمک می‌کند تا به طور موثر با ویژگی‌های داده متنی مطابقت داده شود و صحت نتایج توصیه‌های سیستم بهبود یابد.

---

<sup>20</sup> Pooling Layer

<sup>21</sup> Multi-layer pooling

ماژول آتش شبکه SqueezeNet از هسته‌های پیچشی  $1 \times 1$  و  $3 \times 3$  تشکیل شده است. این اندازه‌ها کوچک هستند و استخراج ویژگی‌های ریز، به خوبی انجام می‌شود. بنابراین برای داده‌ها با تفاوت‌های کم مناسب است. در حالی که تصاویر محصولات سیستم‌های تجارت الکترونیکی<sup>۲۲</sup> معمولاً دارای تفاوت‌های بیشتری در هر کلاس مشخص هستند، یعنی استخراج ویژگی‌های ریز، خیلی کارگشا نخواهد بود. به طور مثال به شکل ۱-۴ توجه کنید. همانطور که مشخص است، انواع مختلفی از شلوارها دیده می‌شوند که همگی در کلاس شلوار قرار دارند و ویژگی‌های کلی‌تری باید برای تشخیص درست استخراج شود.



شکل ۱-۴ چند نمونه از تصاویر شلوار در دادگان مقاله

حال با توجه به توضیحات داده شده، اصلاحات موردنیاز به شرح زیر می‌باشند.

#### ۴-۲-۱: تنظیم هسته‌های پیچشی ماژول آتش

تنظیم اندازه هسته پیچشی در ماژول آتش باید دو نکته زیر را تضمین کند:

- تعداد نقشه‌های ویژگی خروجی دو بخش ساختار بلاک گسترش، باید یکسان باشد. هدف این است که اطمینان حاصل شود که تعداد نقشه‌های ویژگی که پس از گسترش ایجاد می‌شود، یکسان باشد.
- از آنجایی که هسته‌های پیچشی دو قسمت ساختار گسترش از نظر اندازه متفاوت هستند، لازم است که صفر به نقشه ویژگی خروجی کوچکتر اضافه شود تا خروجی نقشه‌های ویژگی دو قسمت به طور کامل دوخته شود، که برای عملیات ادغام مناسب است.

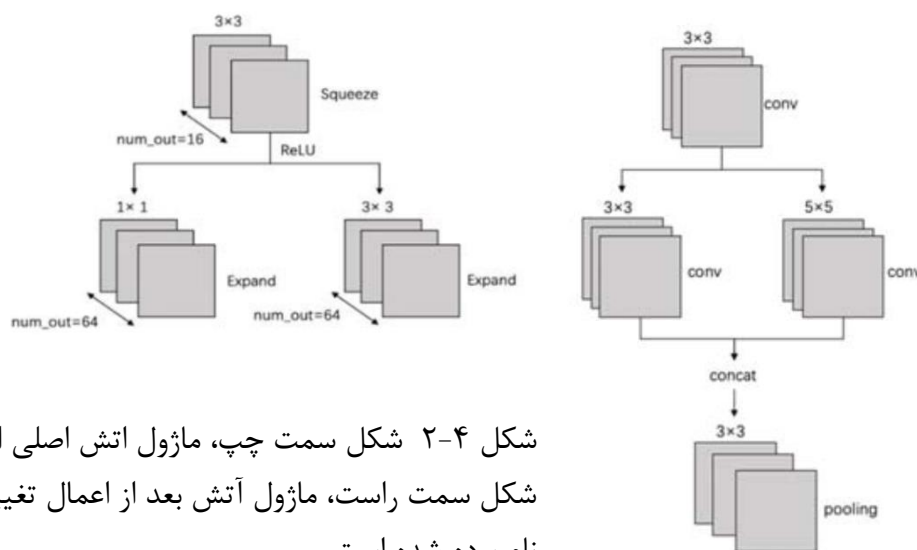
#### ۴-۲-۲: تنظیم ساختار ادغام

علاوه بر تنظیم پارامترهای هسته پیچشی، پس از اینکه ماژول آتش ویژگی‌های تصویر را استخراج کرد، نحوه تنظیم ساختار ادغام برای پالایش مناسب ویژگی‌ها نیز یک عامل کلیدی است که بر صحت تشخیص تأثیر می‌گذارد.

<sup>22</sup> E-commerce

بنابراین، بر اساس آزمایش‌های کامل، این مقاله دو بهبود در SqueezeNet برای این مجموعه داده‌های مورد استفاده در سیستم‌های تجارت الکترونیک ایجاد می‌کند:

- ۱- ماژول آتش اصلی شامل پالایه‌های  $1 \times 1$  و  $3 \times 3$  است که به پالایه‌های  $3 \times 3$  و  $5 \times 5$  تغییر می‌کنند.
- ۲- تعداد ماژول‌های آتش را در شبکه اصلی SqueezeNet تنظیم کرده و بعد از هر ماژول آتش یک لایه ادغام اضافه کرده است. هدف از این کار کاهش تعداد پارامترها در حین استخراج ویژگی‌های کلیدی است. ساختار یک ماژول آتش، قبل و بعد از تغییرات نام برده شده، در شکل ۴-۲ نشان داده شده است.



شکل ۴-۲ شکل سمت چپ، ماژول آتش اصلی است.  
شکل سمت راست، ماژول آتش بعد از اعمال تغییرات نام برده شده است.

### ۳-۴: ارزیابی مدل بهبود یافته SqueezeNet

در جدول ۴-۱ مقایسه‌ای بین صحت برای آزمایش‌هایی روی اندازه هسته پیچشی بلاک گسترش در ماژول آتش است.

شماره آزمایش	اندازه پالایه اول از بخش گسترش	اندازه پالایه دوم از بخش گسترش	صحت Top-5
۱	$1 \times 1$	$3 \times 3$	۹۷٪
۲	$1 \times 1$	$5 \times 5$	۹۷٪
۳	$3 \times 3$	$5 \times 5$	۹۸٪
۴	$5 \times 5$	$7 \times 7$	۸۷٪

جدول ۴-۱ نتایج حاصل از اعمال پالایه‌هایی با اندازه‌های مختلف در ماژول آتش

در جدول ۲-۴ نتایج حاصل از اعمال لایه‌های ادغام به ازای هر ماژول آتش را می‌بینیم. نکته قابل توجه این است که در SqueezeNet اصلی، لایه‌های ادغام فقط بعد از ماژول آتش سه و پنج حضور دارند. ولی در روشی که این مقاله معرفی کرده است، بعد از هر ماژول آتش یک لایه ادغام گذاشته شده است.

صحت Top-5	معماری
٪ ۹۷	SqueezeNet
٪ ۹۸	SqueezeNet بهبود یافته

جدول ۲-۴ نتایج حاصل از اعمال لایه‌های ادغام

نکته قابل توجه دیگر این است که نتایج این بخش، روی دیتاستی است که خود مقاله معرفی کرده است و مربوط به تصاویر محصولاتی است که در سیستم‌های تجارت الکترونیک استفاده می‌شوند.

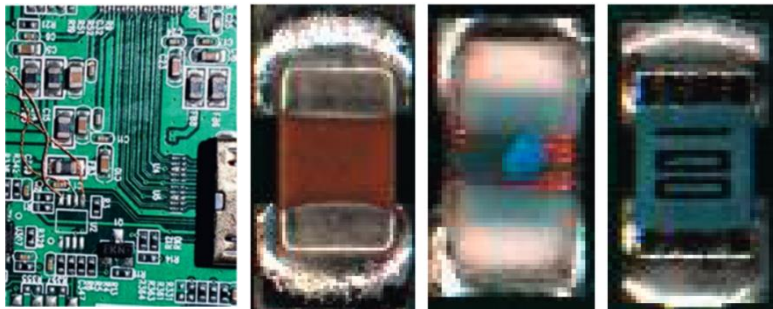
## فصل پنجم

### بهبود اندازه پارامترها و پیچیدگی محاسباتی شبکه SqueezeNet

#### با پیشنهاد شبکه Faster SqueezeNet [۸]

در این فصل، به مقاله‌ای می‌پردازیم که در سال ۲۰۲۰ منتشر شد و تلاش کرد تا یک سیستم شناسایی اجزای الکترونیکی بر اساس الگوریتم بهبود یافته‌ی SqueezeNet پیشنهاد کند که آن را Faster SqueezeNet نامیده است. ما در این فصل، از این به بعد آن را به اختصار FS می‌نامیم. این ساختار می‌تواند اندازه پارامترهای شبکه و پیچیدگی محاسباتی را بدون کاهش عملکرد شبکه کم کند. نتایج تجربی نشان داده است که الگوریتم FS پیشنهادی از نظر پارامترهای مدل، زمان استنتاج و قابلیت انتقال، عملکرد عالی دارد و برای کاربردهای مهندسی تشخیص اجزای الکترونیکی در تولید الکترونیکی مناسب است.

دادگان استفاده شده در این مقاله، بردهای مدار چاپی تلفن‌های همراه، رایانه‌ها، دستگاه‌های تهویه مطبوع، خودروها، و ... هستند. چند نمونه از تصاویر دادگان را در شکل ۵-۱ مشاهده می‌کنید.



شکل ۵-۱ تصاویری از دادگان معرفی شده، به ترتیب از سمت راست، مقاومت، سلف، خازن، برد الکترونیکی

#### ۵-۱: معرفی شبکه بهبود یافته FS

به منظور بهبود صحت و عملکرد بلادرنگ طبقه‌بندی قطعات الکترونیکی، FS پیشنهاد شد. در این شبکه به منظور جلوگیری از بیش‌برازش، از نرمال‌سازی دسته‌ای و ساختارهای باقیمانده‌ای استفاده شده است. همچنین مانند شبکه DenseNet، از عملگر اتصال<sup>۲۳</sup> برای اتصال لایه‌های مختلف استفاده کرده است. شبکه FS از یک

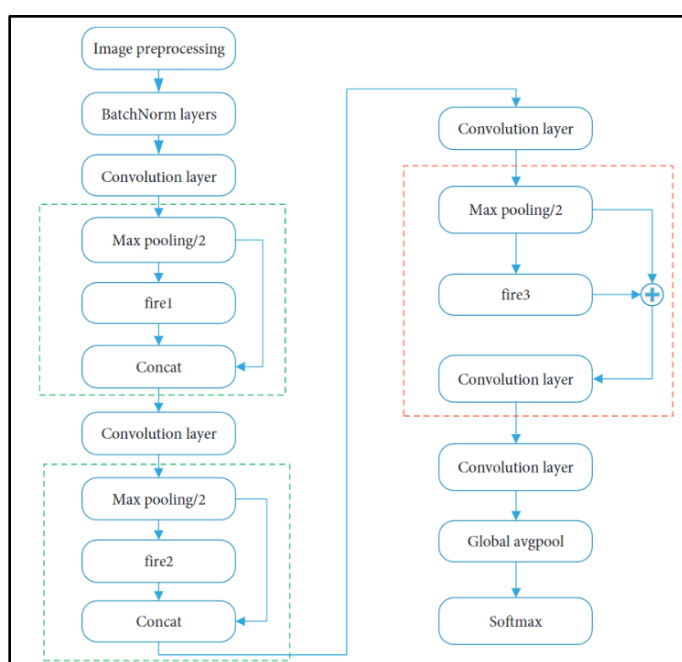
<sup>23</sup> Concat

لایه نرمال سازی دسته‌ای، سه لایه بلوکی، چهار لایه پیچشی و یک لایه ادغام میانگین جهانی<sup>۲۴</sup> تشکیل شده است.

شبکه FS در دو مرحله زیر بهبود پیدا کرده است:

### ۵-۱-۱: تقلید از اتصالات شبکه DenseNet

به منظور بهبود بیشتر جریان اطلاعات بین لایه‌ها، ساختار DenseNet را تقلید کرده است و حالت اتصال متفاوتی را پیشنهاد کرده است. این اتصال شامل یک لایه ادغام و یک ماژول آتش است و در نهایت دو لایه اتصالی که به لایه پیچشی بعدی متصل می‌شوند. ساختار این سبک اتصالات را در کادر سبز رنگ در شکل ۵-۲ مشاهده می‌کنید.



شکل ۵-۲ معماری شبکه FS

بنابراین، بدون افزایش بیش از حد تعداد پارامترهای شبکه، عملکرد شبکه در مراحل اولیه افزایش می‌یابد و در عین حال، هر شبکه دو لایه می‌تواند مستقیماً ارتباط برقرار کند.

### ۵-۱-۲: تقلید از بلوک‌های ساختمانی شبکه ResNet

در این بخش، به منظور اطمینان از همگرایی بهتر شبکه، اینبار از ساختار ResNet استفاده کرده است و بلوک‌های ساختمانی مختلفی را پیشنهاد کرده است که از یک لایه ادغام و یک ماژول آتش

<sup>24</sup> Global Average Pooling

تشکیل شده است. در نهایت، پس از جمع شدن دو لایه، به لایه پیچشی بعدی متصل شده است که ساختار آن در کادر قرمز رنگ در شکل ۵-۲ قابل مشاهده است.

## ۵-۲: ارزیابی مدل بهبود یافته FS

در این مقاله، در نهایت نتایج تجربی نشان داد که شبکه FS پیشنهادی می‌تواند اجزای الکترونیکی روی برد مدار را شناسایی کند و این مدل بهبود یافته قدرت و پتانسیل کاربردی خوبی دارد. در مقایسه با روش‌های سنتی یا سایر مدل‌های یادگیری عمیق، FS پیشنهادی در این مقاله دارای عملکرد عالی در رابطه با پارامترها، زمان استدلال و قابلیت انتقال مدل است که برای کاربردهای مهندسی شناسایی اجزای الکترونیکی در تولید محصولات الکترونیکی مناسب است. در جدول ۵-۱ مقایسه‌ای بین مدل‌های مختلف استفاده شده در این حوزه و مدل SqueezeNet و FS مشاهده می‌کنید. نکته قابل توجه این است که ستون TPR موجود در جدول، مقدار نسبت مثبت واقعی نام دارد و مخفف True Positive Rate است.

معماری CNN	اندازه مدل	TPR	میانگین زمان استنتاج
PCA SVM	۰/۸۷۵ مگابایت	۹۳/۲۹٪	۰/۲۹ میلی ثانیه
ResNetV2	۹/۱۰ مگابایت	۹۶/۹۹۵٪	۴/۶۵ میلی ثانیه
DenseNet	۲/۹۵ مگابایت	۹۶/۹۹۵٪	۳/۳۹ میلی ثانیه
EfficientNet	۱۳/۱ مگابایت	۹۷/۹۹۶٪	۴/۷۲ میلی ثانیه
MobileNetV2	۸/۷۵ مگابایت	۹۹/۹۹۸٪	۴/۸۳ میلی ثانیه
SqueezeNe	۰/۴۷ مگابایت	۹۶/۳۹۱٪	۳/۵۳ میلی ثانیه
Faster SqueezeNe	۰/۲۵۵ مگابایت	۹۹/۹۹۹٪	۲/۶۷ میلی ثانیه

جدول ۵-۱ حساسیت و زمان اجرا مدل‌های مختلف روی دادگان معرفی شده مقاله

## فصل ششم

### کاهش اندازه و پیچیدگی محاسباتی SqueezeNet با پیشنهاد شبکه Weight-Quantized SqueezeNet [۹]

در این فصل، به مقاله‌ای می‌پردازیم که در سال ۲۰۲۲ منتشر شد و تلاش کرد تا یک مدل بهبود یافته از SqueezeNet با وزن‌های چندی‌سازی<sup>۲۵</sup> شده (کوانتیزه شده) برای به حداقل رساندن مصرف حافظه و پیچیدگی محاسباتی ارائه دهد. تفاوتی که این مقاله با مقالات فصل‌های قبل دارد این است که در این مقاله، مستقیماً در معماری SqueezeNet چیزی تغییر نمی‌کند و فقط یک مدل فشرده‌سازی جدید به آن اضافه می‌شود که مقاله آن را Weight-Quantized SqueezeNet نامیده است.

#### ۶-۱: فن چندی‌سازی وزن

به منظور اطمینان از صحت بالای مدل‌های یادگیری عمیق که بر روی بسترهای سخت‌افزاری با محدودیت منابع اجرا می‌شوند، محققان شروع به مطالعه فشرده‌سازی مدل هوش مصنوعی کردند. یکی از روش‌های فشرده‌سازی، چندی‌سازی وزن‌ها است. ایده‌ی چندی‌سازی وزن‌ها این است که وزن‌های با دقت بالا را با وزن‌های کم دقت بدون تغییر در معماری شبکه جایگزین کنیم.

چندی‌سازی وزن، پارامترهای وزن را در قالب اعداد اعشاری ۳۲ بیتی به اعداد صحیح ۸ بیتی تبدیل می‌کند. به این ترتیب، سربار حافظه و ذخیره‌سازی را کاهش می‌دهد. همچنین این کار ممکن است کارایی پیش‌بینی را بهبود بخشد. چالش چندی‌سازی وزن این است که تعادلی بین فشرده‌سازی مدل و از میزان صحت از دست رفته ایجاد کند.

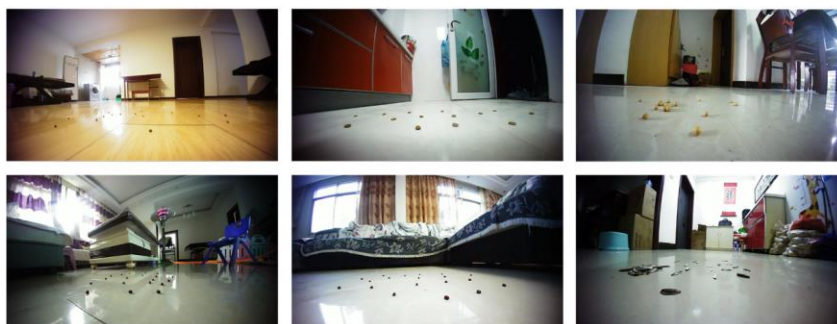
چندی‌سازی وزن‌های مدل علاوه بر کاهش مقدار داده‌هایی که باید ذخیره شوند، به کاهش قابل توجه زمان استنتاج و منابع محاسباتی مورد نیاز کمک می‌کند.

#### ۶-۲: دادگان جمع‌آوری شده‌ی مقاله

یک دادگان از فیلم‌ها یا تصاویر گرفته شده توسط دوربین‌های داخلی ربات جاروبرقی ایجاد کرده است. در شکل ۶-۱ تعدادی از این تصاویر را مشاهده می‌کنیم.

<sup>25</sup> Quantization





شکل ۱-۶ تعدادی از تصاویر دادگان

### ۳-۶: مقایسه و ارزیابی مدل‌های مختلف و مدل پیشنهاد شده

در این مقاله، به جای پیاده‌سازی مدل پیشنهاد شده در یک جاروبرقی واقعی، از پردازنده‌های گرافیکی NVIDIA برای آموزش و آزمایش عملکرد مدل خود استفاده کرده است. در آزمایش‌های انجام شده، تمام وزن‌های مدل‌های هوش مصنوعی توسط بهینه‌ساز SGD<sup>۲۶</sup> آموزش داده شده‌اند، که با به‌روزرسانی پارامترهای وزن در الگوریتم پس‌انتشار، به حداقل رساندن خطای آموزش کمک می‌کند. نرخ یادگیری به عنوان یک تابع کاهنده نمایی پیکربندی می‌شود، جایی که سرعت یادگیری به تدریج از یک مقدار اولیه کاهش می‌یابد. در جدول ۱-۶ مقایسه‌ای بین مدل‌های مختلف و مدل پیشنهاد شده، قبل از چندی‌سازی وزن‌ها مشاهده می‌کنید. همچنین در جدول ۲-۶ همین مقایسه را بعد از اعمال چندی‌سازی وزن‌ها مشاهده می‌کنید.

معماری CNN	اندازه مدل	صحت طبقه‌بندی
4-layer Deep CNN [10]	۲/۱ گیگابایت	٪ ۹۰/۲
VGG-16	۱/۳ گیگابایت	٪ ۹۴/۴
ResNet-34	۱۷۲ مگابایت	٪ ۹۳/۱
16-layer Deep CNN [11]	۱۲۸/۵ مگابایت	٪ ۹۳/۷
MobileNetV2	۱۸/۸ مگابایت	٪ ۹۳/۹
Weight-Quantized SqueezeNet	۶/۱ مگابایت	٪ ۹۳/۲

جدول ۱-۶ اندازه و صحت مدل‌های مختلف روی دادگان تهیه شده قبل از چندی‌سازی وزن‌ها

<sup>26</sup> Stochastic Gradient Descent

معماری CNN	اندازه مدل	میزان کاهش اندازه	صحت طبقه‌بندی
4-layer Deep CNN [10]	۲۶۸/۵ مگابایت	٪ ۸۷	٪ ۹۰/۵
VGG-16	۱۶۵/۹ مگابایت	٪ ۸۷	٪ ۹۴/۱
ResNet-34	۲۱/۸ مگابایت	٪ ۸۷	٪ ۹۳/۱
16-layer Deep CNN [11]	۱۶/۱ مگابایت	٪ ۸۷	٪ ۹۳/۹
MobileNetV2	۲/۶ مگابایت	٪ ۸۶	٪ ۹۳/۹
Weight-Quantized SqueezeNet	۰/۸ مگابایت	٪ ۸۷	٪ ۹۳/۲

جدول ۶-۲ اندازه و صحت مدل‌های مختلف روی دادگان تهیه شده بعد از چندی‌سازی وزن‌ها

## فصل هفتم

### نتیجه‌گیری و پیشنهادها

در شش فصل قبلی مقالات مختلفی را بررسی کردیم که هر کدام از آن‌ها، از روش و جنبه متفاوتی تلاش بر بهبود شبکه SqueezeNet کرده بود. به طور کلی در بین شبکه‌های عصبی پیچشی، شبکه SqueezeNet جزو شبکه‌های کم‌حجم و اصطلاحاً سبک وزن است. در عین حال این شبکه صحتی در سطح شبکه‌ی AlexNet دارد و در نتیجه داشتن صحت خوب و حجم کم آن باعث شده است که بسیار پرکاربرد باشد.

در مقالاتی که بررسی شد، دیدیم که می‌شود با فن‌های مختلفی از جمله افزایش داده، نرمال‌سازی دسته‌ای و ...، حتی نتایج و صحت بهتری از شبکه AlexNet گرفت. اما این نتایج خوب را، روی دادگان ImageNet داریم. و گاهی اوقات وقتی از SqueezeNet روی دادگان دیگری استفاده می‌کنیم، دیگر نتایج قابل قبولی دریافت نمی‌کنیم. این چالش دیگری بود که در چند مقاله به آن پرداخته شده بود. در واقع بسته به اینکه چه استفاده‌ای از شبکه عصبی پیچشی می‌خواهیم بکنیم، می‌توانیم شبکه پایه مورد نظر را با اعمال تغییراتی روی معماری آن، یا افزودن الگوریتم‌های بهینه‌سازی به آن، بهبود دهیم.

نکته مهمی که مطرح است، اینکه برای ایجاد اینگونه بهبودها، نیازمند یک دید جامع و دانش خوب از شبکه‌های عصبی پیچشی است. به طوری که دیدیم چقدر منطقی و آگاهانه، در یکی از مقالات، برای بهبود در راستای نیاز مطرح شده، از سبک اتصالات و بلوک‌های شبکه‌های DenseNet و ResNet الگو گرفته شده بود. پیشنهادی که می‌توان بعد از بررسی این مقالات و کارهایی که انجام شده داد، این است که شاید بتوان دو روشی که در دو فصل سه و پنج مطرح شد را تلفیق کرد. به عبارتی دیگر، با استفاده از روش مطرح شده در فصل پنج، تغییراتی در معماری شبکه SqueezeNet ایجاد کنیم و سپس با استفاده از روش‌های مطرح شده در فصل سه، به سمت گرفتن صحت زیاد، با قبول هزینه‌ای ناچیز برویم.

## منابع و مراجع

- [1] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size." arXiv, 2016. doi: 10.48550/ARXIV.1602.07360.
- [2] F. N. Iandola, K. Ashraf, M. W. Moskewicz, and K. Keutzer, "FireCaffe: near-linear acceleration of deep neural network training on compute clusters." arXiv, 2015. doi: 10.48550/ARXIV.1511.00175.
- [3] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation." arXiv, 2014. doi: 10.48550/ARXIV.1404.0736.
- [4] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both Weights and Connections for Efficient Neural Networks." arXiv, 2015. doi: 10.48550/ARXIV.1506.02626.
- [5] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding." arXiv, 2015. doi: 10.48550/ARXIV.1510.00149.
- [6] Y. Liu, Z. Li, X. Chen, G. Gong, and H. Lu, "Improving the accuracy of SqueezeNet with negligible extra computational cost," 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS). IEEE, May 2020. doi: 10.1109/hpbdis49115.2020.9130577.
- [7] K. Fan, L. Niu, and S. Zhang, "E-commerce Item Identification Based on Improved SqueezeNet," Journal of Physics: Conference Series, vol. 1626, no. 1. IOP Publishing, p. 012002, Oct. 01, 2020. doi: 10.1088/1742-6596/1626/1/012002.
- [8] Y. Xu, G. Yang, J. Luo, and J. He, "An Electronic Component Recognition Algorithm Based on Deep Learning with a Faster SqueezeNet," Mathematical Problems in Engineering, vol. 2020. Hindawi Limited, pp. 1–11, Oct. 28, 2020. doi: 10.1155/2020/2940286.
- [9] Q. Huang, "Weight-Quantized SqueezeNet for Resource-Constrained Robot Vacuums for Indoor Obstacle Classification," AI, vol. 3, no. 1. MDPI AG, pp. 180–193, Mar. 09, 2022. doi: 10.3390/ai3010011.

- [10] Hong Yu, Ruxia Hong, XiaoLei Huang, and Zhengyou Wang, "Obstacle Detection with Deep Convolutional Neural Network," 2013 Sixth International Symposium on Computational Intelligence and Design. IEEE, Oct. 2013. doi: 10.1109/iscid.2013.73.
- [11] J. Yin, K. G. S. Apuroop, Y. K. Tamilselvam, R. E. Mohan, B. Ramalingam, and A. V. Le, "Table Cleaning Task by Human Support Robot Using Deep Learning Technique," Sensors, vol. 20, no. 6. MDPI AG, p. 1698, Mar. 18, 2020. doi: 10.3390/s20061698.