



دانشکده مهندسی برق و کامپیوتر
پایان نامه دوره کارشناسی مهندسی کامپیوتر

طراحی و پیاده‌سازی مدل کانولوشنی جهت دسته‌بندی چندکلاسه تصاویر غذای غذاخوری با روش‌های مقابله با محدودیت در تعداد داده

استاد راهنما:

دکتر محمدرضا فیضی درخشی

پژوهشگر:

پارسا یوسفی نژاد

بهمن ماه ۱۴۰۳

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

چکیده

مدل‌های کانولوشنال شبکه عصبی^۱ به دلیل توانمندی‌های برجسته خود در پردازش تصویر، از جمله استخراج ویژگی‌ها و کاهش نیاز به مهندسی ویژگی، کاهش پیچیدگی محاسباتی، و بهبود سرعت پردازش، یکی از اصلی‌ترین گزینه‌های روی میز برای کاربردهای متنوع در حوزه بینایی ماشین به شمار می‌روند. این مدل‌ها به‌ویژه در مقایسه با شبکه‌های عصبی کاملاً متصل^۲ از بهینگی بسیار بالاتری برخوردار هستند و نتایج دقیق‌تری را ارائه می‌دهند. با این حال، مدل‌های پیچیده‌تر و عمیق‌تر CNN معمولاً برای دستیابی به دقت‌های بالاتر نیازمند مجموعه داده‌های حجیم، متنوع و از توزیع‌های یکنواخت هستند. این امر به‌خصوص در پروژه‌هایی که با محدودیت داده تصویری مواجه‌اند، به یک چالش کلیدی تبدیل می‌شود.

پروژه کلاس‌بندی تصاویر غذاهای سلف دانشگاه تبریز نمونه‌ای از این چالش است. در این پروژه، حجم محدود داده‌های موجود و تنوع کم از هر کلاس و نامتقارن بودن توزیع داده‌ها در هر کلاس از موانع اصلی برای رسیدن به عملکرد بهینه مدل ما است. با توجه به این محدودیت‌ها و اهمیت طراحی مدلی مطمئن و قدرتمند برای دسته‌بندی دقیق تصاویر غذاها، از روش‌های مدرن و پیشرفته‌ای همچون بالانس‌سازی اوزان کلاس‌ها، یادگیری انتقالی^۳ و افزایش داده‌ها^۴ و در نهایت از شیوه ابتکاری و پله‌ای جهت آموزش مدل استفاده شده است.

علاوه بر موارد گفته شده، بهینه‌سازی ساختار مدل شبکه عصبی و استفاده از تکنیک‌های Regularization نیز به منظور جلوگیری از بیش‌برازش^۵ در مدل مدنظر قرار گرفته است. همچنین در نهایت از معیارهای ارزیابی گوناگون مانند دقت^۶، یادآوری^۷ و میانگین وزنی^۸ برای تحلیل عملکرد مدل استفاده شده است. هدف نهایی این پروژه، طراحی و پیاده‌سازی مدلی کارآمد و دقیق است که با وجود محدودیت‌های داده، توانایی ارائه عملکرد مطلوب در طبقه‌بندی تصاویر غذاهای موجود دانشگاه را داشته باشد.

در نهایت، به مدلی کارآمد و دقیق با دقت ۹۲ درصد در طبقه‌بندی تصاویر غذای سلف دست یافتیم که به دلیل سبکی و سرعت مناسب، قابلیت پیاده‌سازی در دستگاه‌های همراه یا سیستم‌های با توان پردازشی محدود را دارد.

^۱ Convolutional Neural Network (CNN)

^۲ Fully Connected Neural Network (CNN)

^۳ Transfer Learning

^۴ Data Augmentation

^۵ Overfitting

^۶ Accuracy

^۷ Recall

^۸ Weighted Average

فهرست مطالب

5.....	فصل اول: مقدمه
6.....	۱.۱ مقدمه
7.....	۲.۱ ساختار پایان نامه
8.....	فصل دوم: روش پیشنهادی
9.....	۱.۲ مقدمه
10.....	۲.۲ یادگیری انتقالی (Transfer Learning) و ساختار مدل
13.....	۳.۲ پیش پردازش داده و افزودن داده مصنوعی
15.....	۳.۲ بالانس کردن وزن کلاس ها
16.....	۴.۲ شیوه آموزش ابتکاری و پله ای مدل
17.....	۱.۴.۲ الگوریتم Adam
18.....	۲.۴.۲ تابع هزینه Categorical Cross-Entropy
19.....	۳.۴.۲ روش آموزش پله ای مدل
21.....	۵.۲ جمع بندی
22.....	فصل سوم: ارزیابی مدل پیشنهادی
23.....	۱.۳ مقدمه
23.....	۲.۳ تحلیل و ارزیابی عملکرد در حین آموزش
25.....	۳.۳ تحلیل و ارزیابی عملکرد نهایی مدل
28.....	۴.۳ جمع بندی
29.....	فصل چهارم: نتیجه گیری و کارهای آینده
30.....	۱.۴ نتیجه گیری
31.....	۲.۴ کارهای آینده
32.....	فصل پنجم: مراجع

فصل اول

مقدمه

۱.۱ مقدمه

پردازش تصویر یکی از حوزه‌های پراهمیت و پرکاربرد در هوش مصنوعی است که نقش اساسی در بسیاری از کاربردهای صنعتی، پزشکی، و پژوهشی ایفا می‌کند. از جمله کاربردهای مهم این حوزه می‌توان به تشخیص اشیا^۱، طبقه‌بندی تصویر^۲، تشخیص چهره^۳، و طبقه‌بندی تصاویر^۴ اشاره کرد. در میان تکنیک‌های موجود، استفاده مدل‌های CNN جایگاه ویژه‌ای دارند و به عنوان یکی از ابزارهای کلیدی در پیشرفت‌های اخیر هوش مصنوعی شناخته می‌شوند. این مدل‌ها نه تنها در پردازش تصاویر، بلکه در دیگر حوزه‌ها مثل تحلیل داده‌های سری زمانی، داده‌های صوتی، و داده‌های پزشکی نیز کاربردهای گسترده‌ای دارند.

یکی از ویژگی‌های بارز CNN‌ها توانایی آن‌ها در استخراج خودکار الگوها و ویژگی‌های پیچیده از داده‌های خام بدون نیاز به پیش‌پردازش دستی و مهندسی ویژگی^۵ گسترده می‌باشد. این قابلیت باعث شده تا CNN‌ها در پروژه‌های مرتبط با طبقه‌بندی و دسته‌بندی تصاویر به عنوان یکی از اصلی‌ترین گزینه‌ها مورد توجه قرار گیرند.

یکی از چالش‌های اصلی در پروژه‌های مرتبط با پردازش تصویر، کمبود داده‌های آموزشی است. محدودیت در تعداد و تنوع داده‌ها می‌تواند به کاهش دقت و توانایی تعمیم مدل منجر شود و اغلب استفاده از مدل‌های پیچیده‌تر را محدود می‌کند. در این پژوهش به منظور غلبه بر این چالش، از روش‌هایی همچون یادگیری انتقالی^۶ و افزایش داده‌ها^۷ استفاده شده است که امکان بهره‌گیری از دانش مدل‌های از پیش آموزش‌دیده^۸ [1] و ایجاد تنوع بیشتر در داده‌ها را فراهم می‌آورند.

در این پروژه، با استفاده از مجموعه‌ای شامل ۱۹۰+۶۴ تصویر از غذاهای سلف دانشگاه تبریز که توسط اینجانب جمع‌آوری شده‌اند، مدلی طراحی شده است که می‌تواند این تصاویر را با دقت ۹۲ درصد به ۹ کلاس مختلف دسته‌بندی کند. این مجموعه داده شامل تصاویری از غذاهای مختلف با تنوع محدود است و همین امر این پروژه را به نمونه‌ای واقعی از چالش‌های مرتبط با کمبود داده تبدیل کرده است.

هدف اصلی این پروژه، طراحی مدلی مقاوم و کارآمد است که بتواند با استفاده از تکنیک‌های پیشرفته از جمله یادگیری انتقالی، تنظیم دقیق مدل^۹، و افزایش داده‌ها، بر محدودیت‌های موجود در داده‌های آموزشی غلبه کند. در کنار این موارد، روش‌هایی همچون تنظیم وزن کلاس‌ها برای کاهش عدم توازن داده‌ها و استفاده از الگوریتم‌های بهینه‌سازی پیشرفته برای افزایش دقت و کاهش خطا نیز به کار گرفته شده‌اند.

^۱ Object Detection
^۲ Image Segmentation
^۳ Face Recognition
^۴ Image Classification
^۵ Feature Engineering

^۶ Transfer Learning
^۷ Data Augmentation
^۸ Pre-trained Models
^۹ Fine-Tuning

۲.۱ ساختار پایان نامه

این پایان نامه در ادامه از سه فصل اصلی به شرح زیر سازماندهی شده است:

- فصل دوم: روش پیشنهادی

در این فصل، انتخاب MobileNetV2 [2] به عنوان مدل پایه به دلیل کارایی و سبکی محاسباتی توجیه شده است. لایه‌های پایه مدل فریز شده‌اند تا ویژگی‌های عمومی حفظ بشود و لایه‌های جدید به انتهای مدل اضافه شده است. پیش‌پردازش داده‌ها شامل تغییر ابعاد تصاویر به 224×224 پیکسل، نرمال‌سازی مقادیر پیکسل، و تولید 5700 تصویر جدید از طریق ۹ تکنیک Data Augmentation انجام گرفته است. وزن‌دهی معکوس به کلاس‌ها برای جبران توزیع داده‌ها از کلاس‌ها اعمال شده، و آموزش مدل در ۵ مرحله با ترکیب داده‌های اصلی و افزوده‌شده، تنظیم اندازه دسته، و استفاده از Early Stopping پیاده‌سازی گردیده شده است.

- فصل سوم: ارزیابی و مدل پیشنهادی

در این فصل، عملکرد مدل با تحلیل نمودارهای دقت آموزش و اعتبارسنجی بررسی شده است. تفاوت بین دقت آموزش و اعتبارسنجی ناشی از تنوع داده‌های Augment شده تفسیر شده است. ارزیابی نهایی روی 64 تصویر تست با معیارهای دقت (100% در Validation Generator و 92% در Test)، یادآوری، و دقت انجام شده است.

- فصل چهارم: نتیجه‌گیری و کارهای آینده

در این فصل، دستاوردهای اصلی شامل طراحی مدلی سبک‌وزن با دقت بالا در شرایط داده‌ای محدود و نامتوازن جمع‌بندی شده و کاربردهای عملی مانند استقرار مدل روی گوشی‌های همراه یا سیستم‌های مدیریت سلف دانشگاه معرفی شده است. در نهایت پیشنهادهاى آینده شامل توسعه مدل برای تشخیص غذاهای ترکیبی، بهبود تعمیم‌پذیری با دادگان^۱ بزرگتر، و ادغام با فناوری‌های مکمل ارائه شده است.

^۱ Dataset

فصل دوم

روش پیشنهادی

۱.۲ مقدمه

در حوزه یادگیری عمیق^۱ و به ویژه در مسائل طبقه‌بندی تصاویر، چالش‌های متعددی در مواجهه با داده‌های محدود، عدم توازن کلاس‌ها، و محدودیت منابع پردازشی وجود دارد. این فصل به بررسی راهکارهای نوین و ترکیبی برای غلبه بر این چالش‌ها در پروژه طبقه‌بندی غذاهای سلف دانشگاه می‌پردازد. با توجه به حجم اندک داده‌های آموزشی (۱۶۱ تصویر اولیه)، استفاده از مدل‌های پیچیده و سنگین از پیش آموزش‌نندیده، نه تنها به دقت مطلوب منجر نمی‌شود، بلکه خطر بیش‌برازش^۲ و ناکارآمدی محاسباتی را افزایش می‌دهد. راهکار اصلی این پژوهش، بهره‌گیری از یادگیری انتقالی^۳ است که با استفاده از دانش آموخته‌شده مدل‌های پیش‌آموزش‌دیده روی داده‌گان بزرگ (مانند ImageNet [3])، امکان استخراج ویژگی‌های عمومی و تطبیق آن‌ها با مسئله خاص را فراهم می‌کند. انتخاب مدل [2] MobileNetV2 به عنوان پایه، به دلیل معماری سبک، سرعت پردازش بالا، و سازگاری با سیستم‌های کم‌منبع، هسته اصلی این راهکار را تشکیل می‌دهد.

علاوه بر این، به منظور جبران کمبود داده، از تکنیک‌های افزایش داده استفاده شده است که با ایجاد تنوع در تصاویر موجود (از طریق تغییرات هندسی، رنگی، و نوری)، حجم مؤثر داده‌های آموزشی را به صورت مصنوعی افزایش می‌دهد. همچنین، مسئله عدم توازن کلاس‌ها با اعمال وزن‌دهی هوشمند به هر کلاس بر اساس فراوانی نمونه‌ها حل شده است تا از بایاس مدل به سمت کلاس‌های پرتعداد جلوگیری شود.

در نهایت، طراحی یک روش آموزش پلکانی و ابتکاری با ترکیب تنظیمات گوناگون (مانند تغییر اندازه دسته، استفاده از داده‌های افزوده‌شده و اصلی، و اعمال تکنیک توقف زودهنگام) به بهینه‌سازی فرآیند یادگیری و بهبود قابلیت تعمیم مدل انجامیده است. این فصل، مراحل پیاده‌سازی، مبانی نظری، و منطق پشت هر یک از این انتخاب‌ها را به تفصیل شرح می‌دهد.

^۱ Deep Learning
^۲ Overfitting

^۳ Transfer Learning

۲.۲ یادگیری انتقالی (Transfer Learning) و ساختار مدل

زمانی که حجم داده‌های موجود برای مسئله محدود باشد، آموزش مدل‌های پیچیده و بزرگ مانند CNN به نتایج مطلوبی منجر نمی‌شود. دلیل اصلی این موضوع، نیاز این مدل‌ها به داده‌های حجیم برای بهینه‌سازی پارامترهای فراوان قابل یادگیری آن‌ها است. برای حل این چالش، از رویکرد یادگیری انتقالی استفاده می‌شود. در این روش، از مدل‌هایی که قبلاً بر روی داده‌گان بزرگ و استاندارد (مانند ImageNet [3]) آموزش دیده‌اند، برای مسائل خاص یا عمومی بهره‌برداری می‌شود.

در یادگیری انتقالی، لایه‌های اولیه مدل که وظیفه استخراج ویژگی‌های کلی از تصاویر را دارند، فریز می‌شوند تا در هنگام پردازش^۱ ورودی‌ها تغییر نکنند. سپس، لایه‌های بالایی مدل (که معمولاً مختص مسئله‌ای خاص طراحی شده‌اند) تغییر داده می‌شوند تا مدل با نیازهای مسئله جدید تنظیم شود.

یکی از چالش‌های اصلی در این فرآیند، انتخاب مدل پایه مناسب برای مسئله بود. مدل‌های متنوعی مانند [4] ResNet، [5] VGG و [6] EfficientNet را مورد بررسی قرار داده شد، اما در نهایت مدل MobileNetV2 به دلیل حجم کم و سرعت پردازش بالا و امکان گسترش در کارهای آتی انتخاب شد. این ویژگی‌ها به مدل نهایی فاین-تیون^۲ شده اجازه می‌دهد حتی در سیستم‌های با محدودیت منابع پردازشی نیز به خوبی عمل کند.

[2] MobileNetV2 یک معماری پیشرفته از CNN است که به‌طور ویژه برای کاربردهای مبتنی بر دستگاه‌های همراه و سیستم‌های کم‌منبع طراحی شده است. این مدل با بهره‌گیری از دو ویژگی کلیدی: بلوک‌های Bottleneck و کانولوشن‌های عمقی-جداپذیر^۳ موفق به کاهش چشمگیر حجم محاسبات و مصرف حافظه شده است، بدون آنکه دقت مدل به‌طور محسوسی افت کند.

ساختار MobileNetV2 با کاهش تدریجی ابعاد فضایی تصویر ورودی و افزایش همزمان تعداد کانال‌های ویژگی، امکان استخراج الگوهای پیچیده را فراهم می‌کند. در انتها، با استفاده از لایه‌های جمع‌بندی هوشمند و تبدیل ویژگی‌ها به بردارهای طبقه‌بندی، مدل را برای انجام وظایف تشخیصی آماده می‌سازد. این تعادل بین کارایی و دقت، MobileNetV2 را به انتخابی ایده‌آل برای پروژه‌های واقع‌گرا با محدودیت منابع تبدیل می‌کند.

^۱ Inference

^۲ Fine-tuning

^۳ Depthwise Separable Convolution

در ادامه معماری توضیح داده شده به صورت خلاصه در جدول ۱.۲ آورده شده است:

جدول ۱.۲: ساختار اصلی معماری MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

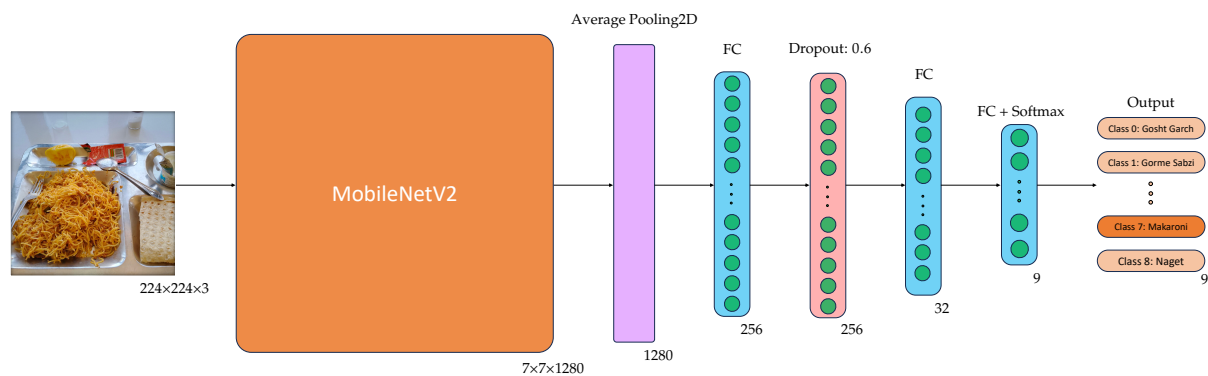
در این پژوهش، که از مدل MobileNetV2 استفاده شده است، لایه‌های نهایی که شامل یک لایه Fully Connected با ۱۲۸۰ نورون بود، با لایه‌های جدید جایگزین شدند. ساختار جدید لایه بالایی مدل پیشنهادی این شکل طراحی شد:

- لایه Global Average Pooling 2D: به جای استفاده از لایه Flatten سنتی که منجر به افزایش چشمگیر پارامترها می‌شود، از لایه Global Average Pooling 2D استفاده شده است. این لایه با محاسبه میانگین مقادیر هر کانال در نقشه‌های ویژگی خروجی، ابعاد و حجم داده‌ها را به صورت هوشمندانه کاهش می‌دهد و اطلاعات ساختاری را حفظ می‌کند.
- لایه Dropout: برای افزایش مقاومت مدل در برابر نویز و کاهش وابستگی به نورون‌های خاص، لایه Dropout با نرخ غیرفعال‌سازی تصادفی ۰.۶ در معماری پس از لایه تمام متصل با ۲۵۶ نورون گنجانده شد. این لایه در هر گام آموزشی، به صورت تصادفی بخشی از نورون‌ها را حذف می‌کند تا مدل نتواند به الگوهای سطحی یا نویزهای موجود در داده‌های آموزشی وابسته شود و در نتیجه، توانایی تعمیم‌پذیری آن بهبود یابد.
- لایه‌های Fully Connected (FC): پس از استخراج ویژگی‌های کلی، دو لایه Fully Connected پیاپی با کاهش تدریجی تعداد نورون‌ها (به ترتیب ۲۵۶، و ۳۲ نورون) طراحی شد. این کاهش گام‌به‌گام، امکان یادگیری ویژگی‌های

انتزاعی و ارتباطات غیرخطی به کمک تابع فعالسازی^۱ ReLU^۲ بین لایه‌ها را فراهم می‌کند و از افت ناگهانی ابعاد داده‌ها که می‌تواند به از دست رفتن اطلاعات حیاتی منجر شود، جلوگیری می‌نماید.

- لایه دسته بندی نهایی: در آخرین لایه، یک لایه Dense با ۹ نورون (مطابق با تعداد کلاس‌های هدف) و تابع فعال‌ساز Softmax جایگذاری شد. این لایه، بردار ویژگی‌های خروجی از لایه‌های پیشین را به توزیع احتمال روی کلاس‌ها تبدیل می‌کند و کلاس نهایی را بر اساس بالاترین احتمال پیش‌بینی شده تعیین می‌نماید. استفاده از Softmax تضمین می‌کند که مجموع احتمالات خروجی برابر با ۱ است و تفسیر نتایج را برای مسئله چندکلاسه آسان می‌سازد.

در ادامه خلاصه‌ای از معماری کلی مدل پیشنهادی طرح شده در فریم ورک TensorFlow در تصویر ۲.۲ آورده شده است.



تصویر ۲.۲: ساختار مدل پیشنهادی

به صورت خلاصه روند پیاده‌سازی به ترتیب زیر است:

۱. فراخوانی مدل پایه: مدل از پیش آموزش دیده MobileNetV2 با استفاده از کتابخانه‌های TensorFlow و Keras به محیط پروژه وارد شد. این مدل به دلیل دارا بودن وزن‌های آموزش دیده روی دادگان ImageNet، به عنوان نقطه شروع انتخاب گردید.
۲. انجماد لایه‌های پایه: به منظور حفظ دانش آموخته شده در لایه‌های ابتدایی مدل (مسئول استخراج ویژگی‌های عمومی مانند لبه‌ها و بافت‌ها)، تمامی لایه‌های اصلی MobileNetV2 غیرفعال^۳ شدند. این اقدام از بازآموزش پارامترهای این لایه‌ها در طول فرآیند Fine-Tuning جلوگیری کرد.

^۱ Activation Function
^۲ Rectified Linear Unit

^۳ Freeze

۳. طراحی معماری سفارشی: لایه‌های جدیدی شامل دو لایه تمام متصل با کاهش تدریجی نورون‌ها (۲۵۶، ۳۲) و یک لایه Dropout با نرخ ۶۰٪ بعد از لایه تمام متصل ۲۵۶ نورونه به مدل پایه افزوده شد. این معماری با هدف کاهش ابعاد داده، جلوگیری از بیش‌برازش و یادگیری ویژگی‌های اختصاصی مرتبط با طبقه‌بندی غذاها طراحی گردید.

۴. تنظیم دقیق^۱ و آموزش: مدل نهایی با استفاده از ۱۶۱ تصویر آموزشی (به همراه داده‌های افزوده‌شده) و با تمرکز بر به‌روزرسانی وزن‌های لایه‌های اضافه‌شده، آموزش داده شد. این فرآیند به مدل اجازه داد تا ضمن حفظ دانش عمومی استخراج ویژگی، خود را با ویژگی‌های منحصر به فرد تصاویر غذاها سلف دانشگاه بهینه‌سازی کند.

۳.۲ پیش پردازش داده و افزودن داده مصنوعی

ورودی مدل MobileNetV2 باید به صورت تصاویر با ابعاد $3 \times 224 \times 224$ باشد؛ بنابراین، تمامی تصاویر، صرف نظر از نسبت ابعاد^۲، به این اندازه تغییر داده شدند. در این مرحله، تنظیم هایپر پارامترهایی نظیر اندازه دسته^۳ و نسبت تقسیم داده‌ها میان داده‌های آموزشی^۴ و اعتبارسنجی^۵ انجام شد. پس از انجام آزمایش‌های متعدد، مقدار اندازه دسته برابر ۳۲ و ۱۵ درصد از داده‌ها به validation اختصاص یافت.

در گام بعدی، از تکنیک [7] Data Augmentation به عنوان یک روش رایج Regularization برای تولید داده‌های اضافی-مصنوعی و متنوع جهت بهبود عملکرد مدل استفاده شد. در این پژوهش، با دستکاری ۹ ویژگی مختلف هر تصویر از ۱۶۱ تصویر موجود در دادگان آموزشی، حدود ۵۷۰۰ تصویر جدید تولید گردید.

ویژگی‌های دستکاری شده تصاویر به شرح زیر است:

- چرخش^۶: ایجاد چرخش در تصاویر به صورت تصادفی در محدوده مشخص برای افزایش تنوع داده.
- برش عرضی^۷: جابه‌جایی تصویر به صورت افقی برای شبیه‌سازی تغییرات دیدگاه.
- برش ارتفاعی^۸: جابه‌جایی تصویر به صورت عمودی برای ایجاد تغییر در موقعیت تصویر.

^۱ Fine-tuning
^۲ Aspect Ratio
^۳ Batch Size
^۴ Train Set

^۵ Validation Set
^۶ Rotation Range
^۷ Width Shift Range
^۸ Height Shift Range

- دامنه برش کانال^۱: تغییرات رنگی در کانال‌های تصویر برای شبیه‌سازی تفاوت در نورپردازی.
- برش افقی^۲: وارونه کردن تصویر در جهت افقی برای تنوع در داده‌ها.
- برش عمودی^۳: وارونه کردن تصویر در جهت عمودی برای تنوع بیشتر.
- دامنه روشنایی^۴: تغییر روشنایی تصاویر برای شبیه‌سازی شرایط نوری مختلف.
- دامنه زوم^۵: زوم کردن به داخل یا خارج از تصویر برای ایجاد تغییرات در مقیاس.
- دامنه برش^۶: اعمال برش و تغییر شکل هندسی تصویر برای افزایش انعطاف در داده‌ها.

همچنین تمامی تصاویر، پیش از ورود به مدل، نرمال^۷ شده و مقادیر هر پیکسل به بازه ۰ تا ۱ مقیاس‌بندی شدند، به این دلیل که نرمال‌سازی تصاویر به بازه ۰ تا ۱ باعث می‌شود که مقادیر پیکسل‌ها در یک محدوده یکسان قرار بگیرند، این امر تسهیل‌کننده فرآیند یادگیری مدل است و از بروز مشکلاتی مانند انفجار گرادیان^۸ یا ناپدید شدن گرادیان^۹ جلوگیری می‌کند. نرمال‌سازی مقادیر پیکسل‌ها باعث می‌شود که گرادیان‌ها در محدوده‌ای متناسب باقی بمانند و وزن‌ها در طی فرآیند backpropagation به‌طور مؤثری به‌روز شوند، که این امر به افزایش پایداری و سرعت همگرایی^{۱۰} مدل کمک می‌کند. همچنین، این تغییرات تأثیر ویژگی‌های مختلف (مانند رنگ یا روشنایی) را در فرآیند آموزش مدل کاهش می‌دهد و باعث می‌شود مدل به‌طور مؤثرتری ویژگی‌های اصلی تصویر را یاد بگیرد.

همچنین برای آموزش مدل، از یک دیتا لودر برای مجموعه آموزشی استفاده شد که در هر دسته، ۳۲ تصویر به‌صورت تصادفی شامل ترکیبی از تصاویر اصلی و Augment شده را تولید می‌کرد. لازم به ذکر است که داده‌های اعتبارسنجی به دلیل ارزیابی درست مدل، تحت عملیات Augmentation قرار نگرفتند و تنها نرمال‌سازی و مقیاس‌بندی شدند. همچنین، هر دو Data Loader آموزش و اعتبارسنجی دارای اندازه دسته برابر ۳۲ بودند که انتخابی متداول و مناسب در کاربردهای عمومی محسوب می‌شود.

^۱ Channel Shift Range

^۲ Horizontal Flip

^۳ Vertical Flip

^۴ Brightness Range

^۵ Zoom Range

^۶ Shear Range

^۷ Normalization

^۸ Gradient Explosion

^۹ Gradient Vanishing

^{۱۰} Convergence

۳.۲ بالانس کردن وزن کلاس‌ها

در پروژه‌های طبقه‌بندی تصاویر، به‌ویژه زمانی که تعداد نمونه‌ها در هر کلاس به‌طور یکنواخت توزیع نشده باشد، یکی از چالش‌های مهم مقابله با بایاس^۱ مدل نسبت به کلاس‌ها است. زمانی که تعداد نمونه‌های یک کلاس بیش از حد زیاد باشد، مدل تمایل دارد که پیش‌بینی‌های خود را به سمت آن کلاس غالب ببرد، حتی اگر این پیش‌بینی‌ها برای کلاس‌های دیگر به اندازه کافی دقیق نباشند. این پدیده به عنوان بایاس در مدل شناخته می‌شود که موجب کاهش دقت مدل برای کلاس‌های نادرتر می‌شود.

برای حل این مشکل، بالانس کردن وزن کلاس‌ها [8] یکی از تکنیک‌های مؤثر است. این روش تلاش می‌کند تا در هنگام آموزش مدل، تأثیر کلاس‌های نادر و غالب را به‌طور برابر تنظیم کند. به عبارت دیگر، مدل به‌طور ویژه بر روی کلاس‌هایی که تعداد نمونه‌های کمتری دارند، تمرکز بیشتری خواهد کرد و از پیش‌بینی‌های نادرست برای کلاس‌های کم‌نمونه جلوگیری می‌شود.

روش معمول برای بالانس کردن وزن کلاس‌ها استفاده از وزن‌دهی به هر کلاس است. در این روش، هر کلاس بر اساس تعداد نمونه‌های موجود در آن، وزن خاصی دریافت می‌کند. کلاس‌هایی که نمونه‌های کمتری دارند، وزن بیشتری دریافت می‌کنند و در نتیجه، مدل باید به آن‌ها توجه بیشتری داشته باشد. این وزن‌ها در طی فرآیند آموزش برای اصلاح خطاها در هر کلاس و بهبود دقت در پیش‌بینی کلاس‌های نادر استفاده می‌شوند.

برای محاسبه وزن کلاس‌ها، ابتدا تعداد نمونه‌های هر کلاس به‌طور جداگانه شمارش می‌شود، سپس وزن هر کلاس به‌طور معکوس با تعداد نمونه‌های آن کلاس محاسبه می‌شود. این فرمول به‌طور دقیق به شکل زیر است:

$$weight_i = \frac{N_{total}}{C_i \times N_{samples}} \quad \text{رابطه ۱.۲}$$

که در آن :

- $weight_i$ وزن کلاس i است.
- N_{total} مجموع کل نمونه‌ها در تمام کلاس‌ها است.
- C_i تعداد نمونه‌ها در کلاس i است.
- $N_{samples}$ تعداد کل کلاس‌ها است.

^۱ Bias

بالانس کردن وزن کلاس‌ها به مدل کمک می‌کند تا از بایاس کلاس‌های غالب جلوگیری کند و به‌طور برابر به هر کلاس توجه کند. این امر به‌ویژه در مسائل با داده‌های نامتوازن اهمیت دارد، چرا که مدل قادر به یادگیری صحیح ویژگی‌های کلاس‌های نادر می‌شود و دقت آن‌ها بهبود می‌یابد. به‌عنوان مثال، در پروژه‌ای که تعداد تصاویر کلاس چلوکباب چندین برابر تصاویر کلاس ناگت مرغ است، با اعمال وزن‌دهی مناسب به کلاس‌ها، مدل می‌تواند از تمایل خود برای پیش‌بینی کلاس غالب (چلوکباب) جلوگیری کند و دقت برای کلاس نادرتر (ناگت مرغ) را افزایش دهد.

به صورت کلی در این روش، وزن‌دهی به کلاس‌ها به مدل این امکان را می‌دهد که در زمان آموزش، اشتباهات مربوط به کلاس‌های اقلیتی را مهم‌تر از اشتباهات کلاس‌های اکثریتی در نظر بگیرد. به عبارت دیگر، اشتباهاتی که در پیش‌بینی کلاس‌های نادر رخ می‌دهند، با افزایش وزن‌های کلاس‌ها بزرگتر می‌شوند، که باعث می‌شود مدل در فرآیند backpropagation به این اشتباهات توجه بیشتری داشته باشد و پارامترهای خود را برای بهبود پیش‌بینی‌های کلاس‌های اقلیتی به‌طور مؤثرتری به‌روز کند. این عمل باعث می‌شود که در هنگام محاسبه تابع خطا^۱، خطای مربوط به کلاس‌های اقلیتی نسبت به کلاس‌های اکثریتی بزرگ‌تر محاسبه شود. در نتیجه، مدل تمرکز بیشتری بر روی یادگیری از داده‌های کلاس‌های نادر پیدا می‌کند و به‌طور مؤثری به تقویت دقت خود در طبقه‌بندی آن‌ها می‌پردازد.

۴.۲ شیوه آموزش ابتکاری و پله‌ای مدل

پس از انجام تمامی مراحل پیشین، آموزش مدل به شیوه‌ای ابتکاری طراحی شد تا مدل نسبت به تصاویر مختلف، حتی از توزیع‌های متفاوت، قابل اتکا و دقیق شود؛ الگوریتم [9] Adam به‌عنوان بهینه‌ساز^۲ انتخاب شد، زیرا با ترکیب ویژگی‌های دو روش مومنتوم^۳ و RMSProp، نرخ همگرایی^۴ را افزایش داده، به مشکلات تنظیم نرخ یادگیری^۵ حساس نیست و در مسائل پیچیده با داده‌های متغیر عملکردی پایدار ارائه می‌دهد، و تابع هزینه^۶ [7] categorical cross-entropy نیز به کار گرفته شد، چراکه با مدل‌سازی احتمالاتی توزیع کلاس‌ها، اختلاف بین توزیع پیش‌بینی‌شده مدل و توزیع واقعی داده‌ها را به‌طور دقیق اندازه‌گیری کرده و امکان یادگیری بهتر در مسائل طبقه‌بندی چندکلاسی را فراهم می‌کند.

در ادامه به بررسی الگوریتم Adam و سپس تابع هزینه Categorical cross-entropy استفاده شده در پژوهش می‌پردازیم.

^۱ Loss Function

^۲ Optimizer

^۳ Momentum

^۴ Convergence Rate

^۵ Learning Rate

^۶ Cost Function

۱.۴.۲ الگوریتم Adam

الگوریتم بهینه‌سازی Adam [9] از دو مفهوم اصلی استفاده می‌کند:

۱. مومنتوم: که به آن کمک می‌کند تا در مسیر بهینه‌سازی از تغییرات بزرگ جلوگیری شود و به سمت بهینه‌سازی با سرعت ثابت حرکت کند.
۲. تنظیم گام یادگیری برای هر پارامتر^۱: که با تغییر مقدار گام یادگیری برای هر پارامتر بر اساس مقیاس گرادیان‌های آن پارامتر، باعث بهبود سرعت همگرایی می‌شود.

مراحل الگوریتم Adam:

۱. محاسبه میانگین اول و دوم گرادیان‌ها:
در ابتدا برای هر پارامتر، میانگین اول (گرادیان‌ها) و میانگین دوم (مربع گرادیان‌ها) محاسبه می‌شود:
- m_t : میانگین وزنی از گرادیان‌ها تا گام t
 - v_t : میانگین وزنی از مربع گرادیان‌ها تا گام t

فرمول‌های محاسبه این دو به صورت زیر هستند:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} J(\theta) \quad \text{رابطه ۲.۲}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} J(\theta))^2 \quad \text{رابطه ۳.۲}$$

در این فرمول‌ها:

- $\nabla_{\theta} J(\theta)$ گرادیان تابع هزینه نسبت به پارامتر θ است.
- β_1 و β_2 پارامترهای تنظیمی هستند که معمولاً به ترتیب به ۰.۹ و ۰.۹۹۹ تنظیم می‌شوند.

۲. اصلاح پیش‌نمایش‌ها برای جلوگیری از تعصب اولیه^۲:

در ابتدای آموزش، میانگین‌های m_t و v_t از صفر شروع می‌شوند که باعث بایاس اولیه در محاسبات می‌شود. برای رفع این بایاس‌ها، اصلاحاتی انجام می‌شود:

^۱ RMSprop

^۲ Bias Correction

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad \text{رابطه ۴.۲}$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad \text{رابطه ۵.۲}$$

۳. به روزرسانی پارامترها:

در نهایت، پارامترها با استفاده از میانگین‌های تصحیح‌شده \widehat{m}_t و \widehat{v}_t به روزرسانی می‌شوند. گام به روزرسانی به صورت زیر محاسبه می‌شود:

$$\theta_t = \theta_{t-1} - \frac{\alpha \widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \quad \text{رابطه ۶.۲}$$

که در اینجا:

- θ_t : پارامتر مدل در گام t است.
- α : نرخ یادگیری است.
- ϵ : یک مقدار کوچک است (معمولاً 10^{-8}) که برای جلوگیری از تقسیم بر صفر در زمان به روزرسانی استفاده می‌شود.

۲.۴.۲ تابع هزینه Categorical Cross-Entropy

فرمول ریاضی برای محاسبه تابع هزینه Categorical Cross-Entropy [7] به شکل زیر می‌باشد:

$$\text{Loss} = - \sum_{i=1}^C y_i \log(p_i) \quad \text{رابطه ۷.۲}$$

که در آن:

- C تعداد کلاس‌ها است.
- y_i مقدار برجسب واقعی (که برای طبقه‌بندی One-Hot، تنها برای کلاس صحیح برابر 1 و برای بقیه کلاس‌ها برابر 0 است)
- p_i احتمال پیش‌بینی‌شده برای کلاس i است.
- \log لگاریتم طبیعی است.

در این تابع هزینه، برای هر نمونه داده‌ای، احتمال پیش‌بینی شده برای هر کلاس p_i محاسبه می‌شود. سپس این احتمال‌ها با برچسب‌های واقعی (که به صورت One-Hot کدگذاری شده‌اند) مقایسه می‌شوند.

در نهایت، هزینه برای همه کلاس‌ها جمع می‌شود. هدف آموزش مدل این است که مقدار این تابع هزینه را کمینه کند.

این ترکیب از Adam و Categorical Cross-Entropy باعث می‌شود که مدل هم به‌طور خودکار نرخ یادگیری خود را تنظیم کند و هم با دقت بالا کلاس‌ها را حتی در داده‌های پیچیده و با توزیع متنوع از هم تفکیک کند.

۳.۴.۲ روش آموزش پله‌ای مدل

ابتکار این بخش در طراحی ۵ مرحله آموزش مختلف است که در هر مرحله از تنظیمات و استراتژی‌های متفاوتی استفاده شد. وجه تمایز این مراحل، استفاده از ترکیبات مختلف data generator و اعمال تکنیک Early Stopping برای جلوگیری از بیش‌برازش بود. در مراحل اولیه از اندازه دسته کوچک‌تر ۳۲ تایی استفاده شد و در مراحل آخر هم از اندازه دسته با اندازه‌های ۶۴ تایی استفاده شد. این به این دلیل است که در مراحل اولیه آموزش، استفاده از اندازه دسته کوچک‌تر (۳۲) کمک می‌کند تا مدل با جزئیات بیشتری از داده‌ها آشنا شود و پارامترها با دقت بیشتری تنظیم شوند. اندازه دسته کوچک‌تر، به دلیل ایجاد نویز بیشتر در محاسبات گرادیان، مدل را از افتادن در کمینه‌های محلی ضعیف باز می‌دارد و به کشف مسیرهای بهتر در فضای جستجو کمک می‌کند. این امر به مدل امکان می‌دهد تا در شروع، ویژگی‌های پیچیده‌تر داده‌ها را با دقت بیشتری استخراج و یاد بگیرد.

در مراحل پایانی، استفاده از اندازه دسته بزرگ‌تر (۶۴) به هموارسازی مسیر یادگیری مدل کمک می‌کند. با افزایش اندازه دسته، میانگین‌گیری روی تعداد بیشتری از نمونه‌ها انجام می‌شود و گرادیان‌ها پایدارتر و دقیق‌تر می‌شوند. این امر باعث می‌شود که مدل به سمت کمینه‌های بهینه‌تر همگرا شود و همچنین فرآیند یادگیری سریع‌تر انجام گیرد. ترکیب این دو استراتژی باعث می‌شود که مدل ابتدا به‌صورت عمیق و دقیق ویژگی‌ها را یاد بگیرد و سپس در مراحل پایانی، با ثبات بیشتری همگرا شود و از بیش‌برازش جلوگیری شود.

Early Stopping نیز تضمین می‌کند که آموزش متوقف شود پیش از آنکه مدل بیش از حد روی داده‌های آموزشی تنظیم شود و عملکرد آن روی داده‌های آزمایشی کاهش یابد. این تکنیک با رصد عملکرد مدل در داده‌های اعتبارسنجی^۱ تعیین می‌کند که بهترین نقطه توقف کجاست.

در نهایت جزئیات مراحل آموزش مدل پیشنهادی به شرح زیر است:

- مرحله اول: مدل برای ۳۰ اپیاک روی train generator شامل تصاویر augment شده همراه با داده‌های اعتبارسنجی آموزش داده شد. نسبت تقسیم داده‌ها در این generator، ۱۵ درصد برای اعتبارسنجی و ۸۵ درصد برای آموزش در نظر گرفته شد.
 - مرحله دوم: اندازه دسته به ۶۴ افزایش یافت و مدل برای ۳۵ اپیاک دیگر با داده‌های augment شده آموزش داده شد. این تغییر باعث بهبود همگرایی مدل با افزایش حجم داده در هر به‌روزرسانی شد.
 - مرحله سوم: برای تمرکز بیشتر مدل روی داده‌های اصلی^۲، یک train generator بدون augmentation ایجاد شد. مدل با این generator و اندازه دسته برابر با ۳۲ برای ۱۶ اپیاک دیگر آموزش داده شد. این کار منجر به بهبود دقت مدل شد، زیرا داده‌های اورجینال ساده‌تر و مستقیم‌تر به مدل ارائه شدند.
 - مرحله چهارم: مشابه مرحله اول، مدل برای ۲۰ اپیاک دیگر با داده‌های augment شده آموزش داده شد تا تنوع بیشتری در داده‌ها ایجاد شود و توانایی تعمیم مدل افزایش یابد.
 - مرحله پنجم: در آخرین مرحله، اندازه دسته و نسبت داده‌های آموزش به اعتبارسنجی تغییر داده شد و تکنیک Early Stopping از کتابخانه Keras اعمال شد. این تکنیک به گونه‌ای تنظیم شد که در صورت افت قابل توجه در هزینه^۳ مربوط به مجموعه داده آموزشی، آموزش مدل به صورت خودکار متوقف شود و از بیش‌برازش جلوگیری شود.
- این روش آموزش چندمرحله‌ای با تنظیمات خاص، مدل را بهینه کرده و توانسته است دقت و قابلیت تعمیم آن را در مواجهه با داده‌های واقعی و متنوع به میزان چشمگیری افزایش دهد.

^۱ Validation

^۲ Original Data

^۳ Loss

۵.۲ جمع‌بندی

در این فصل، چارچوب جامعی برای آموزش یک مدل طبقه‌بندی تصاویر با داده‌های محدود و منابع پردازشی کم‌ظرفیت ارائه شد. هسته مرکزی این چارچوب، مدل MobileNetV2 بود که پس از انجماد لایه‌های بدنه و بازطراحی لایه‌های بالایی، به صورت اختصاصی مدل برای مسئله تشخیص غذاهای سلف دانشگاه تنظیم شد. استفاده از لایه‌های Global Average Pooling و Dropout در معماری جدید، نه تنها ابعاد مدل را کاهش داد، بلکه از بیش‌برازش با حذف نورون‌های تصادفی در طول آموزش جلوگیری کرد.

تکنیک Data Augmentation با ایجاد حدود ۵۷۰۰ تصویر جدید از ۱۶۱ نمونه اولیه، تنوع داده‌ها را به طور مصنوعی افزایش داد و مدل را در برابر تغییرات محیطی (نور، زاویه دید، و مقیاس) مقاوم ساخت. نرمال‌سازی مقادیر پیکسل‌ها به بازه [۰, ۱] نیز به پایداری فرآیند آموزش و جلوگیری از مشکلات گرادیان کمک زیادی کرد.

برای مقابله با عدم توازن کلاس‌ها، وزن‌دهی معکوس بر اساس فراوانی هر کلاس اعمال شد تا خطای پیش‌بینی کلاس‌های با تعداد کم در تابع هزینه تقویت گردد. این امر مدل را وادار کرد تا به جای تمرکز صرف بر کلاس‌های پرتعداد، ویژگی‌های کلاس‌های نادر را نیز به دقت یاد بگیرد.

در مرحله آموزش، ترکیب هوشمندانه الگوریتم Adam و تابع هزینه Categorical Cross-Entropy همراه با تقسیم آموزش به پنج مرحله (با تغییر اندازه دسته، استفاده متناوب از داده‌های افزوده‌شده و اصلی، و اعمال Early Stopping) به مدل اجازه داد تا ابتدا با داده‌های متنوع آشنا شود و سپس به تدریج بر داده‌های واقعی نیز متمرکز گردد.

فصل سوم

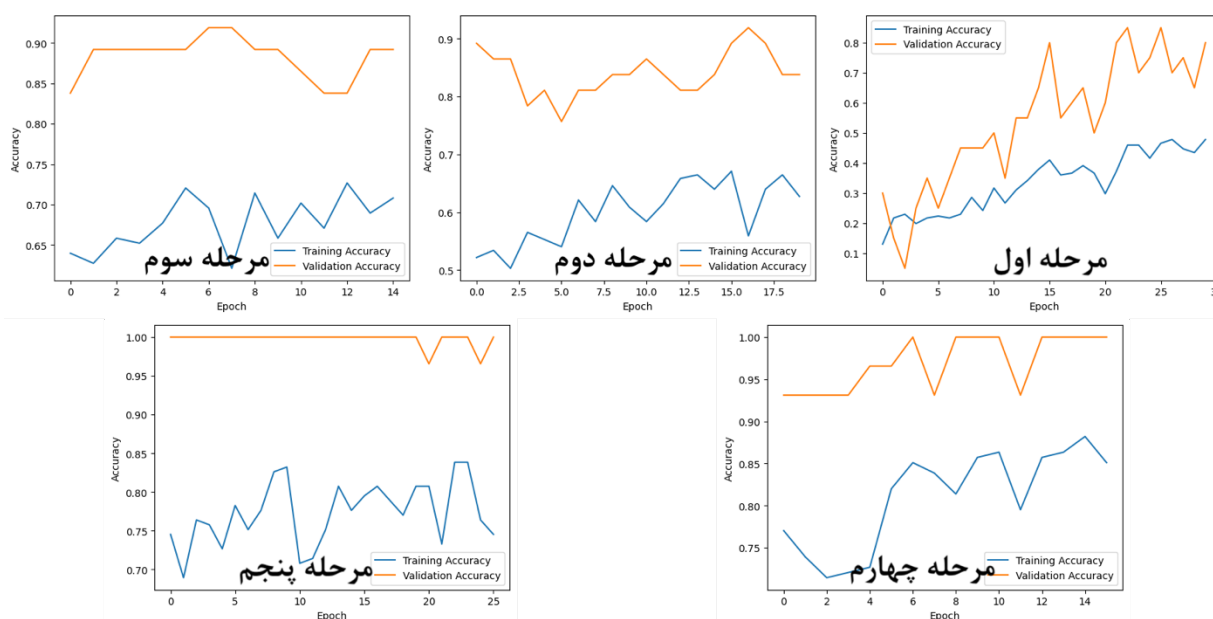
ارزیابی مدل پیشنهادی

۱.۳ مقدمه

ارزیابی مدل‌های یادگیری عمیق^۱، به‌ویژه در مسائل پیچیده، بخشی کلیدی برای سنجش عملکرد و تعمیم‌پذیری^۲ آن‌ها است. در این بخش، پس از آموزش مدل، عملکرد آن روی مجموعه داده‌های تست و اعتبارسنجی^۳ مورد بررسی قرار می‌گیرد. این ارزیابی با استفاده از معیارهای مختلفی از جمله دقت^۴، میانگین یادآوری^۵، میانگین دقت^۶، و F1-Score انجام شده است. همچنین، از ماتریس سردرگمی^۷ برای تحلیل جزئی‌تر رفتار مدل در شناسایی کلاس‌های مختلف استفاده شده است.

۲.۳ تحلیل و ارزیابی عملکرد در حین آموزش

برای درک بهتر فرآیند یادگیری، نمودارهای دقت آموزش و اعتبارسنجی در هر مرحله از آموزش ترسیم شده‌اند. تصویر ۱.۳، روند تغییرات دقت مدل بر اساس تعداد اپاک در مراحل اول تا پنجم آموزش را نمایش می‌دهد. تحلیل این نمودارها نشان می‌دهد که دقت اعتبارسنجی در تمامی مراحل بالاتر از دقت آموزش بوده است. این موضوع به‌طور مستقیم تحت تأثیر تنوع بالای تصاویر ایجادشده از طریق Data Augmentation قرار دارد.



تصویر ۱.۳: نمودار دقت بر حسب اپاک برای مجموعه داده‌های آموزشی و اعتبارسنجی در تمامی مراحل آموزش مدل پیشنهادی (مرحله اول تا پنجم). در این نمودار، خطوط آبی نشان‌دهنده دقت در مجموعه آموزشی و خطوط نارنجی نشان‌دهنده دقت در مجموعه اعتبارسنجی هستند.

^۱ Deep Learning
^۲ Generalization
^۳ Validation
^۴ Accuracy

^۵ Mean Sensitivity
^۶ Mean Precision
^۷ Confusion Matrix

بر اساس تصویر ۱.۳، در مراحل ابتدایی آموزش، مدل پیشنهادی با رشد سریعی در دقت مواجه شده و هم‌زمان، دقت اعتبارسنجی از دقت آموزش پیشی گرفته است. در مراحل بعدی، این فاصله به مرور افزایش پیدا کرده است. دلیل اصلی این روند، تأثیر فرآیند Data Augmentation بر تنوع و پیچیدگی داده‌های آموزشی است. این فرآیند با اعمال تغییرات گوناگون بر روی داده‌ها، ویژگی‌های پیچیده‌تری را به مدل ارائه کرده و آن را قادر ساخته است تا حالت‌های مختلفی از نمونه‌های داده را یاد بگیرد. در نتیجه، مدل طی فرآیند آموزش با داده‌های چالش‌برانگیزتری مواجه شده و توانایی استخراج ویژگی‌های غنی‌تر و تعمیم‌پذیری بیشتری را کسب کرده است. در مقابل، داده‌های مجموعه اعتبارسنجی که از توزیع اصلی تصاویر انتخاب شده‌اند، مدل را در شرایط ساده‌تری مورد ارزیابی قرار داده‌اند. این تفاوت در سطح پیچیدگی داده‌ها، عامل اصلی عملکرد بهتر مدل در مجموعه اعتبارسنجی بوده است، زیرا داده‌های این مجموعه فاقد تغییرات گسترده‌ای هستند که در داده‌های آموزشی اعمال شده‌اند. در نتیجه، مدل در مواجهه با داده‌های اعتبارسنجی که به نمونه‌های واقعی نزدیک‌تر هستند، عملکرد بهتری نشان داده است.

برای ارزیابی روند آموزشی، طبق جدول ۱.۳، در اولین مرحله از آموزش که به تعداد ۳۰ ایپاک اجرا شده است، دقت اولیه مدل در مجموعه داده آموزشی پس از ۶ قدم^۱ برابر با ۱۳٪ بوده و در انتهای مرحله، پس از ۲۹ ایپاک، به ۴۷٫۷۸٪ افزایش یافته است. هم‌زمان، مقدار هزینه از ۱۰٫۸۸ به ۲٫۷۵ کاهش پیدا کرده است. این نتایج نشان‌دهنده پیشرفت چشمگیر مدل در مراحل اولیه آموزش است، به‌طوری که شاهد کاهش قابل توجه هزینه و افزایش دقت، هم در مجموعه داده آموزشی و هم در مجموعه داده اعتبارسنجی هستیم.

برای بهبود بیشتر دقت مدل و کاهش بیشتر هزینه، مراحل چهارم به بعد آموزش انجام شد.

جدول ۱.۳: به ازای هر مرحله آموزشی و به تعداد ایپاک ذکر شده، هزینه و دقت در ابتدا و انتهای هر ایپاک را محاسبه شده‌است.

مراحل آموزش	تعداد ایپاک ^۱	مجموعه داده آموزشی		مجموعه داده اعتبارسنجی	
		هزینه ^۱	دقت ^۱	هزینه	دقت
		اولین ایپاک	آخرین ایپاک	اولین ایپاک	آخرین ایپاک
مرحله اول	۳۰	۱۰٫۸۸	۲٫۷۵	۹٫۷۴	۲٫۴۲
مرحله دوم	۲۰	۲٫۷۰	۲٫۰۸	۲٫۲۶	۱٫۶۶
مرحله سوم	۱۵	۱٫۹۹	۱٫۶۴	۱٫۶۱	۱٫۴۰
مرحله چهارم	۱۶	۱٫۷۲	۱٫۲۹	۱٫۳۳	۰٫۹۲
مرحله پنجم	۲۵	۱٫۳۰	۱٫۲۹۴	۰٫۸۵	۰٫۸۳

^۱ Step

در مراحل چهارم و پنجم، با استفاده از تکنیک‌های ذکرشده در بخش ۳.۴.۲، تلاش شده است مدل را بیشتر بهبود داده شود بدون آنکه دچار بیش‌برازش شود. این اقدامات باعث شدند که مدل از مرحله چهارم به بعد به دقت ۱۰۰٪ و کمترین میزان هزینه (۰٫۸۳) در مجموعه داده اعتبارسنجی دست یابد. با این حال، همان‌طور که مشاهده می‌شود، در مرحله پنجم پیشرفت قابل توجهی در تنظیم-دقیق^۱ کردن مدل رخ نداده است. در این مرحله، مدل به نقطه‌ای رسیده بود که عملکرد آن با توجه به معماری انتخاب‌شده و نوع داده‌ها به حداکثر توان خود رسیده است. به همین دلیل، آموزش مدل آزاد بود که به تعداد بیشتری از اپیک‌ها ادامه یابد و برای جلوگیری از افزایش هزینه یا بروز بیش‌برازش، از Early Stopping استفاده شده است تا در صورت توقف بهبود، فرآیند آموزش متوقف شود. این امر نشان داد که مدل با تنظیمات و داده‌های فعلی به یک وضعیت بهینه دست یافته و بهبود بیشتری امکان‌پذیر نیست.

۳.۳ تحلیل و ارزیابی عملکرد نهایی مدل

برای ارزیابی نهایی مدل، از مجموعه داده تست شامل ۶۴ تصویر استفاده شده است که به‌صورت جداگانه از پوشه تصاویر اصلی استخراج شدند و شامل داده‌های کاملاً جدید بودند که مدل در طول آموزش یا اعتبارسنجی آن‌ها را ندیده بود.

جهت ارزیابی مدل نهایی از معیارهای زیر استفاده شده است:

- دقت پیش‌بینی^۲: درصد پیش‌بینی‌های صحیح از کل نمونه‌ها.
- یادآوری^۳: توانایی مدل در شناسایی صحیح نمونه‌های مثبت از هر کلاس.
- دقت^۴: درصد نمونه‌های پیش‌بینی‌شده صحیح در هر کلاس.
- F1-Score: میانگین هارمونیک دقت و یادآوری، که تعادل بین این دو معیار را اندازه‌گیری می‌کند.
- ماتریس سردرگمی^۵: ماتریسی که تعداد پیش‌بینی‌های صحیح و غلط را برای هر کلاس نشان می‌دهد. این ابزار به تحلیل دقیق عملکرد مدل در شناسایی هر کلاس کمک می‌کند و به‌ویژه مواردی مانند پیش‌بینی اشتباه یک کلاس به‌عنوان کلاس دیگر را برجسته می‌کند.

در جدول ۲.۳ ارزیابی عملکرد مدل نهایی به ازای هر کلاس از مجموعه تست و همچنین میانگین معیارها آورده شده است.

^۱ Fine-tuning

^۲ Accuracy

^۳ Recall

^۴ Precision

^۵ Confusion Matrix

در گزارش عملکرد مدل بر روی داده‌های تست، ابتدا مقادیر یادآوری و دقت مورد بررسی قرار گرفته‌اند. معیار یادآوری بیانگر توانایی مدل در شناسایی صحیح تمام نمونه‌های واقعی از یک کلاس است و به‌عنوان سنجشی از جامعیت مدل در تشخیص نمونه‌های مثبت شناخته می‌شود. به عبارت دیگر، این معیار نشان می‌دهد که چه درصدی از نمونه‌های مثبت توسط مدل به درستی شناسایی شده‌اند. از سوی دیگر، معیار دقت توانایی مدل را در پیش‌بینی صحیح نمونه‌ها برای هر کلاس نشان می‌دهد و مشخص می‌کند که از میان پیش‌بینی‌های انجام‌شده برای یک کلاس خاص، چه درصدی از آن‌ها به‌درستی صورت گرفته‌اند. مقادیر یادآوری و دقت برای هر یک از کلاس‌ها در جدول ۲.۳ به‌طور دقیق آورده شده است.

همچنین، در بخش تعداد نمونه‌های کلاس، تعداد نمونه‌هایی که مدل آن‌ها را به کلاس‌های پیش‌بینی‌شده اختصاص داده است، مشخص شده است. این اطلاعات کمک می‌کنند تا دریابیم مدل چه تعداد نمونه از هر کلاس را پردازش کرده و چگونه پیش‌بینی‌ها را میان آن‌ها توزیع کرده است.

طبق نتایج جدول ۲.۳، مدل بهترین عملکرد خود را برای تفکیک تصاویر کلاس‌های ۰ (گوشت و قارچ)، ۳ (جوجه)، ۶ (ماهی) و ۸ داشته است. یکی از دلایلی که منجر به تفکیک ۱۰۰ درصدی این کلاس‌ها توسط مدل نهایی شده است، تعداد کمتر نمونه‌های موجود در این کلاس‌ها نسبت به کلاس‌های پرجمعیت‌تر است.

جدول ۲.۳: عملکرد مدل در ارزیابی مجموعه داده تست

شماره کلاس مجموعه تست	دقت	یادآوری	F1-Score	تعداد نمونه‌های کلاس ^۱
۰	۱,۰۰	۱,۰۰	۱,۰۰	۶
۱	۰,۸۳	۱,۰۰	۰,۹۱	۱۰
۲	۱,۰۰	۰,۸۹	۰,۸۴	۹
۳	۱,۰۰	۱,۰۰	۱,۰۰	۲
۴	۰,۹۲	۰,۸۵	۰,۸۸	۱۳
۵	۰,۷۵	۱,۰۰	۰,۸۶	۳
۶	۱,۰۰	۱,۰۰	۱,۰۰	۴
۷	۰,۹۳	۰,۸۸	۰,۹۰	۱۶
۸	۱,۰۰	۱,۰۰	۱,۰۰	۱
میانگین ساده معیارها	۰,۹۴	۰,۹۶	۰,۹۴	۶۴
میانگین وزنی معیارها	۰,۹۳	۰,۹۲	۰,۹۲	۶۴
دقت پیش‌بینی کل مدل			۰,۹۲	

برای جلوگیری از وجود بایاس در ارزیابی عملکرد مدل، علاوه بر محاسبه میانگین ساده معیارهای ارزیابی، میانگین وزنی این معیارها نیز با توجه به تعداد نمونه‌های هر کلاس محاسبه شده است تا دقت واقعی مدل به‌دست آید. در عمل، تفاوت چندانی میان میانگین ساده و میانگین وزنی در مقدار F1-Score مشاهده نمی‌شود و این اختلاف تنها ۲٪ است.

^۱ Support

میانگین ساده^۱ F1-Score مدل برابر با ۹۴٪ به دست آمده است که نشان‌دهنده دقت بالای مدل در طبقه‌بندی تصاویر است. همچنین، مقادیر یادآوری و دقت به ترتیب ۹۶٪ و ۹۴٪ به دست آمده‌اند که عملکرد بسیار خوب مدل را در شناسایی و دسته‌بندی تصاویر مختلف غذاها تأیید می‌کنند. این مقادیر به وضوح توانایی مدل را در بازشناسی صحیح نمونه‌های مثبت و منفی نشان می‌دهند. میانگین وزنی^۲ معیارهای F1-Score، یادآوری و دقت به ترتیب برابر با ۹۲٪، ۹۲٪ و ۹۳٪ محاسبه شده‌اند. این نتایج نشان‌دهنده تعمیم‌پذیری بالای مدل و عدم وابستگی آن به توزیع نمونه‌ها در هر کلاس هستند. به عبارت دیگر، این مقادیر بیانگر این واقعیت‌اند که مدل به‌خوبی قادر به شبیه‌سازی و طبقه‌بندی دقیق تصاویر، حتی در مواجهه با داده‌های نامتوازن، است.

در نهایت، میانگین وزنی و هارمونیک معیارهای یادآوری و دقت، به‌عنوان دقت کل مدل گزارش شده است. این نتایج نشان می‌دهند که دقت نهایی مدل بر روی تصاویر تستی که تاکنون ندیده بود، برابر با ۹۲٪ است. این عملکرد به‌وضوح توانایی بالای مدل را در تعمیم‌دهی به داده‌های جدید و طبقه‌بندی صحیح آن‌ها نشان می‌دهد.

در تصویر ۲.۳ ماتریس سردرگمی را برای مجموعه تست بررسی می‌شود:

[6	0	0	0	0	0	0	0	0]
[0	10	0	0	0	0	0	0	0]
[0	0	8	0	0	0	0	1	0]
[0	0	0	2	0	0	0	0	0]
[0	2	0	0	11	0	0	0	0]
[0	0	0	0	0	3	0	0	0]
[0	0	0	0	0	0	4	0	0]
[0	0	0	0	1	1	0	14	0]
[0	0	0	0	0	0	0	0	1]]

تصویر ۲.۳: ماتریس سردرگمی برای مجموعه تست

ماتریس سردرگمی یک ابزار بصری برای ارزیابی عملکرد مدل‌های طبقه‌بندی در یادگیری ماشین است که در آن هر ستون نشان‌دهنده تعداد نمونه‌های واقعی هر یک از کلاس‌های مجموعه تست می‌باشد، برای مثال ستون شماره ۵ام بیانگر نمونه‌هایی هست که واقعا به کلاس ۳ تعلق دارند، همچنین هر سطر در این ماتریس، بیانگر نمونه‌هایی است که مدل پیش‌بینی کرده و به همان سطر اختصاص داده است، برای مثال طبق تصویر ۲.۳، در سطر سوم در مجموعه ۹ تصویر را مدل به کلاس ۲ اختصاص داده در حالی که یک تصویر را به اشتباه به کلاس ۷ طبقه‌بندی کرده است. در این ماتریس تنها تصاویری به درستی طبقه‌بندی شده‌اند که بر روی قطر ماتریس مربعی سردرگمی قرار داشته باشند. مدل به طور کلی توانسته عملکرد

^۱ Macro Average

^۲ Weighted Average

خوبی در طبقه‌بندی تصاویر داشته باشد. با این حال، یک اشتباه جزئی در دسته‌بندی وجود داشته است. به‌طور خاص، در ۱۲ تصویر مربوط به کلاس دوم، مدل ۲ تصویر را به اشتباه به کلاس پنجم طبقه‌بندی کرده است. این نشان‌دهنده یک اشتباه کوچک در فرآیند طبقه‌بندی است که می‌توان با بهینه‌سازی‌های بیشتر آن را کاهش داد.

۴.۳ جمع‌بندی

مدل ارائه‌شده با استفاده از داده‌های Augmented و متنوع، توانسته است ویژگی‌های پیچیده‌تر را از داده‌ها استخراج کرده و عملکرد بسیار مناسبی بر روی داده‌های تست، که پیش‌تر هرگز دیده نشده بودند، ارائه دهد. دقت کلی ۹۲ درصد و مقادیر میانگین وزنی^۱ بالای دقت (۹۳٪) و یادآوری (۹۲٪) نشان‌دهنده توانایی بالا و عمومی‌سازی مناسب مدل است. تحلیل ماتریس سردرگمی و نتایج حاصل از ارزیابی نشان می‌دهند که مدل در اغلب کلاس‌ها عملکردی بسیار قوی داشته و تنها در تعداد محدودی از پیش‌بینی‌ها دچار خطا شده است. این نتایج نشان‌دهنده موفقیت استراتژی‌های آموزشی استفاده‌شده و آماده بودن مدل برای استفاده در دنیای واقعی است.

^۱ Weighted Average

فصل چهارم

نتیجه‌گیری و کارهای آینده

۱.۴ نتیجه گیری

در این پروژه، با استفاده از تکنیک‌های پیشرفته یادگیری ماشین و بینایی ماشین، توانستیم سیستم شناسایی و طبقه‌بندی تصاویر غذاهای سلف دانشگاه را با دقت و کارایی بسیار بالا طراحی و پیاده‌سازی کنیم. هدف اصلی، بهبود عملکرد مدل‌ها در شرایط داده‌های غیر متوازن و افزایش توانایی تعمیم‌پذیری آنها بود. برای رسیدن به این هدف، چندین رویکرد مهم اتخاذ شد که عبارت بودند از:

۱. پیش‌پردازش و افزایش داده‌ها: این مرحله به منظور غلبه بر محدودیت‌های موجود در تنوع داده‌ها و کاهش تاثیر داده‌های غیر متوازن انجام شد. تکنیک‌هایی مانند چرخش، تغییر مقیاس، برش تصادفی، و تغییر روشنایی تصاویر به کار گرفته شد تا تنوع بیشتری در مجموعه داده ایجاد شود.
۲. استفاده از یادگیری انتقالی: مدل MobileNetV2، که به دلیل ساختار سبک و کارایی بالا شناخته شده است، به عنوان مدل پایه انتخاب شد. این رویکرد به ما امکان داد از ویژگی‌های از پیش استخراج‌شده در لایه‌های ابتدایی برای شناسایی ویژگی‌های عمومی و از لایه‌های انتهایی برای طبقه‌بندی دقیق استفاده کنیم.
۳. تکنیک‌های بهینه‌سازی: تنظیم وزن کلاس‌ها برای جلوگیری از بایاس، انتخاب مناسب نرخ یادگیری^۱ و استفاده از Early Stopping برای جلوگیری از بیش‌برازش مدل از جمله مواردی بودند که باعث افزایش دقت و پایداری مدل شدند.
۴. تنظیمات آموزش: تغییرات در اندازه دسته و نسبت تقسیم داده‌ها بین مجموعه آموزش و اعتبارسنجی، تاثیر بسزایی در بهبود عملکرد نهایی مدل داشتند.

در مرحله ارزیابی، نتایج به دست آمده نشان داد که مدل به خوبی توانسته است داده‌های جدید و نادیده گرفته‌شده را شناسایی و طبقه‌بندی کند. معیارهایی مانند دقت پیش‌بینی، یادآوری، دقت و F1-score نشان‌دهنده عملکرد چشمگیر سیستم بودند. دقت ۱۰۰ درصدی بر روی داده‌های اعتبارسنجی و دقت ۹۲ درصدی بر روی ۶۴ داده تست تأیید می‌کند که مدل نه تنها در شناسایی ویژگی‌های پیچیده موفق بوده، بلکه در تعمیم‌پذیری نیز عملکرد قابل قبولی داشته است.

^۱ Learning Rate

این پروژه ثابت کرد که با بهره‌گیری از مدل‌های سبک و کارآمد مانند MobileNetV2 همراه با تکنیک‌های پیش‌پردازش و افزایش داده، می‌توان سیستم‌های شناسایی تصویر را حتی در شرایط چالش‌برانگیز به دقت بالا رساند. علاوه بر این، امکان استفاده از مدل بر روی دستگاه‌های سبک مانند گوشی‌های هوشمند، به دلیل سبک بودن ساختار آن، یکی از دستاوردهای مهم این تحقیق است.

۲.۴ کارهای آینده

در نهایت مجموعه‌ای از کارهایی که پس از این پژوهش می‌توانیم انجام دهیم را آورده‌ایم:

۱. گسترش مجموعه داده‌ها: به‌منظور افزایش تعمیم‌پذیری مدل، پیشنهاد می‌شود که مجموعه داده‌های بیشتری از غذاهای متنوع و محیط‌های مختلف جمع‌آوری شود تا مدل با توزیع‌های مختلف بیشتر آشنا شود.
۲. استفاده از مدل‌های هیبریدی: ترکیب MobileNetV2 با مدل‌های پیشرفته‌تر مانند EfficientNet می‌تواند عملکرد سیستم را در شناسایی دقیق‌تر بهبود بخشد.
۳. پشتیبانی از چندین کلاس همزمان: توسعه مدل به گونه‌ای که بتواند غذاهایی با چندین برچسب^۱ را شناسایی کند، می‌تواند قابلیت‌های آن را ارتقاء دهد.
۴. بهینه‌سازی برای استفاده بلادرنگ^۲: برای کاربردهایی مانند سلف دانشگاه، بهینه‌سازی بیشتر مدل برای پردازش بلادرنگ روی دستگاه‌های کم‌قدرت پیشنهاد می‌شود.
۵. اضافه کردن قابلیت تحلیل مواد مغذی: یک قابلیت مفید دیگر می‌تواند تحلیل مواد مغذی موجود در غذاها بر اساس تصویر باشد که نیازمند توسعه مدل‌های تکمیلی است.
۶. وب اپلیکیشن: طراحی یک وب اپلیکیشن بر بستر ابر و پیاده‌سازی مدل برای استفاده تمام کاربران با سیستم‌های مختلف که می‌تواند به مقبولیت بیشتر سیستم ما و کاربردی‌تر کردن آن کمک کند.
۷. استفاده از تکنیک‌های خودنظارتی: استفاده از روش‌های Self-Supervised Learning برای بهبود عملکرد در شرایط کمبود داده‌های برچسب‌گذاری شده می‌تواند موثر باشد.
۸. ارزیابی در محیط‌های واقعی: بررسی عملکرد مدل در محیط‌های واقعی مانند سلف‌های دانشگاهی مختلف یا سایر محیط‌های عمومی برای ارزیابی دقیق‌تر و شناسایی چالش‌های احتمالی پیشنهاد می‌شود.

^۱ Multi-label Classification

^۲ Real-Time

فصل پنجم

مراجع

- [1] "TensorFlow Hub," TensorFlow, [Online]. Available: <https://www.tensorflow.org/hub>.
- [2] Sandler, Mark Howard, Andrew Zhu, Menglong Zhmoginov, Andrey Chen, Liang-Chieh, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [3] Deng, Jia Dong, Wei Socher, Richard Li, Li-Jia Li, Kai Fei-Fei, Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009.
- [4] He, Kaiming Zhang, Xiangyu Ren, Shaoqing Sun, Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [5] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [6] Tan, Mingxing Le, Quoc, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019.
- [7] devendewc0m, "geeksforgeeks," geeksforgeeks, 17 Sep 2024. [Online]. Available: <https://www.geeksforgeeks.org/categorical-cross-entropy-in-multi-class-classification/>.
- [8] G. Developers, "Classification on imbalanced data," TensorFlow, 20 08 2024. [Online]. Available: https://www.tensorflow.org/tutorials/structured_data/imbalanced_data.
- [9] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [10] LeCun, Yann Bottou, Léon Bengio, Yoshua Haffner, Patrick, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
- [11] G. Developers, "Data augmentation in TensorFlow," TensorFlow, 19 07 2024. [Online]. Available: https://www.tensorflow.org/tutorials/images/data_augmentation.