نشریه علمی علوم و فنون نقشه‌برداری، دوره ۱۵، شماره ۱، صفحه ۳۷ تا ۴۸، شهریور ماه ۱۴۰۴ – مقاله پژوهشی

تشخیص نمادهای زمینی و مقاصد گردشگری با طراحی یک سرویس وب پردازش تصویر مبتنی بر شبکه های عصبی پیچشی خفیف

محمد حسن وحیدنیا*[1]

[1] استادیار مرکز مطالعات سنجش از دور و GIS، دانشکده علوم زمین، دانشگاه شهید بهشتی، تهران، ایران
mh_vahidnia@sbu.ac.ir

(دریافت: بهمن ۱۴۰۳، تصویب: تیر ۱۴۰۴)

چکیده

توسعه گردشگری با رشد فناوری اطلاعات و ارائه خدمات مکان-مبنا به تکامل بیشتری منتهی شده است. یکی از زیر ساخت‌ها در خدمات مکان-مبنا برای ایجاد برنامه‌های کاربردی در سال‌های اخیر، تشخیص مقاصد گردشگری و نمادهای زمینی از روی تصویر می‌باشد تا بر اساس آن سایر اطلاعات به کاربران ارائه شود. در این پژوهش، به عنوان مشارکت اصلی، یک سرویس پردازشی وب برای برخی نمادهای شناخته شده تهران طراحی و آزمایش می‌شود تا امکان شناخت آن‌ها از تصویر به صورت آنلاین امکانپذیر شود. همچنین، از مزایای شبکه‌های عصبی پیچشی خفیف (lightweight) در این راستا بهره‌گیری می‌شود. برای این منظور ابتدا چهارصد تصویر از چندین نماد شهر تهران مورد استفاده قرار گرفت و یک شبکه عصبی پیچشی معمولی و یک شبکه عصبی پیچشی خفیف مورد ارزیابی و مقایسه قرار گرفت. شبکه عصبی پیچشی پیش آموزش دیده خفیف، نسبت به شبکه معمولی، صحت کلی ۹۲٪ در مقابل ۷۱٪ را نتیجه داد و علاوه بر برتری در سایر متریک‌های عملکرد، در مدت زمان بسیار کوتاهتر ۵ دقیقه در مقابل ۹۰ دقیقه، آموزش دید. پس از آن یک سرویس پردازش وب به کمک فناوری‌های tensorflow، flask توسعه یافته و بر روی ارائه‌دهنده سرویس ابری Render راه‌اندازی شد. ارزیابی‌های زمانی و مقیاس‌پذیری نیز نتایج رضایت-بخشی نشان داد و روند افزایش زمانی در حضور کاربران همزمان با شیب بسیار کم ۰٫۰۳۹ همراه شده و در ۹۰٪ آزمایش‌ها، پاسخ موفقی از سمت سرور دریافت شد. این پژوهش نشان داد، رویکرد مذکور می‌تواند زیرساخت مناسبی برای شناخت و اخذ اطلاعات مقاصد گردشگری در برنامه‌های کاربردی باشد.

واژگان کلیدی: پردازش تصویر، یادگیری عمیق، شبکه عصبی پیچشی خفیف، میراث فرهنگی، گردشگری، خدمات وب

* نویسنده رابط

۱- مقدمه

شناخت راحت‌تر شهر توسط افراد و توسعه گردشگری با رشد فناوری اطلاعات و ارائه خدمات مکان-مبنا به بلوغ بیشتری منتهی شده است [۱]. در حال حاضر شاهد هستیم که پلتفرم‌های هوشمندی برای بازیابی اطلاعات، ارائه خدمات و پیشنهاد مکان‌های گردشگری به کاربران و ناوبری آن‌ها ارائه می‌شوند [۲]. یکی از موجودیت‌های مهم در امر شناسایی و ناوبری کاربران و گردشگران، اماکن و ساختمان‌های ویژه برای بازدید و به طور خاص نمادهای زمینی می‌باشند [۳]. نمادهای زمینی یک ویژگی طبیعی یا مصنوعی قابل تشخیص است که برای ناوبری استفاده می‌شود، ویژگی‌ای که از محیط نزدیک خود متمایز است و اغلب از فواصل دور قابل مشاهده است [۴]. به طور مثال برج آزادی یا برج میلاد در تهران از جمله شناخته شده‌ترین نمادهای زمینی هستند. هرچند اماکن و ساختمان‌هایی به عنوان مقاصد گردشگری نیز موجود هستند که ممکن است کمتر شناخته شده یا دست کم برای گردشگران ناآشنا اینچنین باشند. در حال حاضر سرویس‌های پردازشی وب بومی که توانایی تشخیص نمادها از تصویر را داشته باشد در کشور وجود نداشته و ایجاد چنین زیرساختی در توسعه برنامه‌های کاربردی خدمات مکان-مبنا برای توسعه گردشگری بیش از پیش احساس می شود.

طبقه‌بندی تصاویر و تشخیص شیء در تصویر یکی از پرکاربردترین و مهمترین قابلیت‌های بینایی ماشینی یا رایانه‌ای می‌باشد [۵ و ۶]. پس از موفقیت در استفاده از شبکه‌های عصبی پیچشی عمیق[۱] (DCNN) برای طبقه‌بندی تصویر، تشخیص اشیا نیز بر اساس تکنیک‌های یادگیری عمیق پیشرفت قابل‌توجهی داشت [۷]. تشخیص اشیاء مبتنی بر یادگیری عمیق، اشیاء موجود را در یک تصویر یا ویدیو شناسایی می‌کند و نشان می‌دهد که در کجا قرار دارند (یعنی، محلی‌سازی شی) و به کدام دسته تعلق دارند (یعنی طبقه‌بندی شی) [۸]. در این رابطه پژوهش-هایی وجود دارند که تلاش نموده‌اند نمادهای زمینی را از تصویر به کمک روش‌های یادگیری عمیق بازیابی نمایند.

به عنوان مثال، سامانی (۲۰۱۹) روشی را برای استخراج خودکار نمادهای زمینی با خوشه‌بندی تصاویر برچسب‌گذاری‌شده جغرافیایی با استفاده از روش خوشه‌بندی مکانی، بر اساس تشخیص تراکم با الگوریتم DBSCAN و تشخیص شی با استفاده از الگوریتم شبکه عصبی عمیق (شبکه باور عمیق) پیشنهاد دادند [۹]. نتایج نشان داد که این روش می تواند نقش موثری در مسیریابی داشته باشد. چاووانواتی و همکاران (۲۰۲۲) از یادگیری ماشینی برای شناسایی ساختمان‌های شاخص در پوکت استفاده کردند [۱۰]. با جمع آوری تصاویر ۱۵ ساختمان مختلف از وب، سیستم با استفاده از مدل های شبکه عصبی کانولوشن از پیش آموزش دیده مانند DenseNet، EfficientNet، Inception و غیره آموزش داده شد. برای بهینه‌سازی عملکرد مدل، عملکرد سیستم با استفاده از یادگیری گروهی و رای گیری نرم بهبود یافت.

رزعلی و همکاران (۲۰۲۳) یک مدل سبک وزن و قوی برای شناسایی ساختمان های شاخص پیشنهاد می‌کنند که برای استفاده در سیستم‌های راهنمای تور هوشمند در صنعت گردشگری اهمیت می‌یابد [۱۱]. با توجه به چالش‌های شناسایی ساختمان‌ها در مکان‌های عمومی به دلیل پیچیدگی صحنه‌ها، این مطالعه از ترکیب CNN و تحلیل تشخیص خطی LDA استفاده می‌کند. ماداک و همکاران (۲۰۲۴) روش جدیدی را بر اساس معماری انتقال بینایی[۲] (ViT) برای شناسایی دقیق ساختمان‌های شاخص در فضاهای باز از روی تصاویر پیشنهاد می‌کنند [۱۲]. آنها به این نتیجه رسیدند که این مدل از CNN های معمولی در شناسایی ساختمان‌ها در شرایط سخت بیرونی بهتر عمل می‌کند.

طبق نتایج تحقیقات پیشین، الگوریتم‌های جدید مبتنی بر یادگیری عمیق با اختلاف زیادی از الگوریتم‌های تشخیص سنتی بهتر عمل می‌نمایند [۵ و ۱۳]. شبکه عصبی پیچشی عمیق یک ساختار الهام گرفته از بیولوژیک برای محاسبه ویژگی‌های سلسله مراتبی است. برخلاف توصیفگرهای دست ساز که در آشکارسازهای سنتی استفاده می‌شود، شبکه‌های عصبی پیچشی عمیق نمایش‌های ویژگی سلسله مراتبی را از پیکسل‌های خام تا اطلاعات معنایی سطح بالا تولید می‌کنند که به طور خودکار از داده‌های آموزشی آموخته می‌شوند و قابلیت بیان متمایز بیشتری را در زمینه‌های پیچیده نشان می‌دهند. علاوه بر

۱ Deep Convolutional Neural Networks

۲ Visual Transfer

این، با بهره‌مندی از ظرفیت یادگیری قدرتمند، یک شبکه عصبی پیچشی عمیق می‌تواند نمایش ویژگی بهتری را با مجموعه داده بزرگ‌تر به دست آورد، در حالی که ظرفیت یادگیری توصیف‌گرهای بصری سنتی ثابت است و زمانی که داده‌های بیشتری در دسترس قرار می‌گیرد، نمی‌تواند بهبود یابد [۱۴].

اگرچه روش‌های جدید یادگیری عمیق مانند شبکه کانولوشن گراف (GCN)، ترانسفورمرها، شبکه‌های از پیش آموزش دیده و مشابه آن در تشخیص نمادها از تصاویر در سال‌های اخیر مورد توجه زیادی قرار گرفته‌اند، اما حوزه تخصصی آن‌ها عمدتاً تصاویر مربوط به پزشکی یا تشخیص چهره بوده است [۱۵، ۱۶ و ۱۷] و کمتر پژوهش‌هایی میتوان یافت که به تشخیص نمادهای شهری و چالش‌های آن پرداخته باشند.

با نگاهی به پژوهش‌های قبل می‌توان به این نتیجه رسید که به شبکه‌های عصبی پیچشی خفیف[۱] در مقایسه با CNN و مشتقات آن در استخراج نمادهای زمینی شهر به ندرت توجه شده است. همچنین توسعه سرویس وب برای تشخیص آنی نمادهای زمینی و میراث فرهنگی بر اساس مدل‌های یادگیری عمیق تعلیم دیده، به ندرت مورد پژوهش واقع شده است. رفع این شکاف‌های پژوهشی به اهداف اصلی این مقاله تبدیل می‌شوند. بنابراین، به عنوان مشارکت اصلی، یک سرویس پردازشی وب برای برخی نمادهای شناخته شده تهران طراحی و آزمایش می-شود تا امکان شناخت آن‌ها از تصویر به صورت آنلاین امکانپذیر شود. همچنین، از مزایای شبکه‌های عصبی پیچشی خفیف در این راستا بهره‌گیری می‌شود.

۲- مواد و روش‌ها

۲-۱- مدل‌های یادگیری

شبکه‌های عصبی پیچشی (CNNS) به‌ویژه برای تشخیص و طبقه‌بندی تصاویر مؤثر هستند. معماری CNN شامل چندین لایه است که هرکدام وظیفه خاصی در استخراج و پردازش ویژگی‌ها از تصاویر دارند [۱۸]. لایه‌های پیچشی هسته‌ها یا فیلترهایی را اعمال می‌کنند که ویژگی‌های محلی تصویر را استخراج می‌کنند. این فیلترها به صورت ماتریس‌های کوچکی هستند که بر روی تصویر ورودی حرکت می‌کنند و عملیات پیچشی را انجام می‌دهند.

$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m.n) \qquad (۱)$$

که در آن I تصویر ورودی و K هسته پیچشی است.

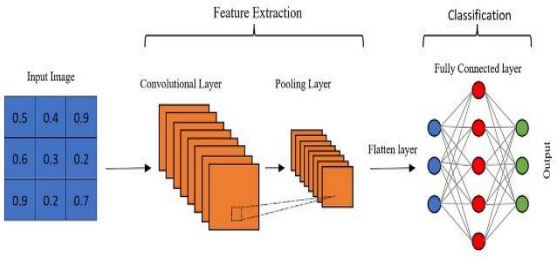
لایه‌های مهم دیگر لایه‌های تجمعی می‌باشند. این لایه‌ها اندازه داده‌ها را کاهش می‌دهند و در عین حال ویژگی‌های مهم را حفظ می‌کنند. این کار باعث کاهش پیچیدگی محاسباتی و جلوگیری از بیش‌برازش می‌شود. نوع رایج لایه‌های تجمعی تجمعی بیشینه نام دارند که طبق رابطه زیر عمل می‌کنند.

$$P(i,j) = \max_{m.n}\big(I(i+m, j+n)\big) \qquad (۲)$$

لایه‌های مهم دیگر، لایه های کاملاً متصل نام دارند. این لایه‌ها مشابه شبکه‌های عصبی کلاسیک عمل می‌کنند و وظیفه طبقه‌بندی نهایی را بر عهده دارند:

$$y = f(Wx + b) \qquad (۳)$$

که در آن W ماتریس وزن، x ورودی و b بایاس است. معماری CNN در شکل ۱ مشاهده می شود.

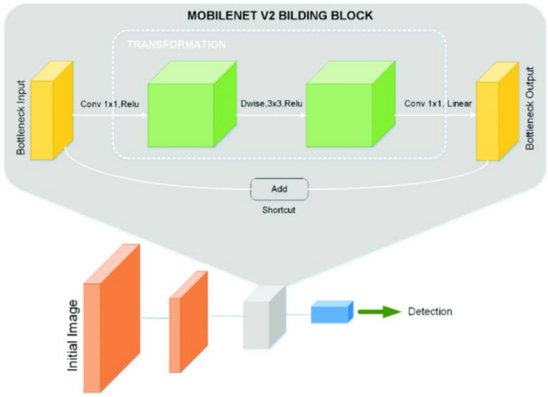


شکل ۱- معماری CNN (https://peerj.com/articles/cs-1395/)

همچنین MobileNetV2 یکی از مدل‌های پیشرفته و سبک برای طبقه‌بندی و شناسایی اشیاء در تصاویر است که توسط گوگل ارائه شده و بر پایه معماری MobileNet توسعه یافته‌است [۱۹]. هدف از طراحی MobileNetV2 ایجاد مدلی کارآمد برای دستگاه‌های دارای منابع محدود، مانند موبایل‌ها و سیستم‌های تعبیه شده است. این مدل از معماری خاصی به نام Inverted Residuals و Linear Bottlenecks استفاده می‌کند که موجب کاهش قابل توجه تعداد پارامترها و محاسبات می‌شود، بدون آن‌که دقت مدل به‌طور چشم‌گیری کاهش یابد. این معماری در شکل ۲

۱ Lightweight

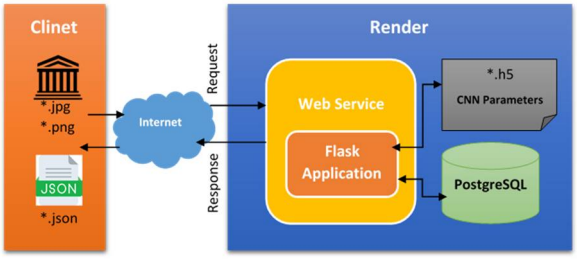
نشان داده شده است. معماری MobileNetV2 شامل بلوک‌هایی است که به جای استفاده از لایه‌های عمیق و سنگین، از عملیات‌های سبک‌تر بهره می‌گیرند. در این مدل از مفاهیم Depthwise Separable Convolutions، Inverted Residuals، و Linear Bottlenecks استفاده می شود.



شکل ۲- معماری MobileNetV2 [۲۰]

۲-۲- طراحی وب سرویس تشخیص نماد

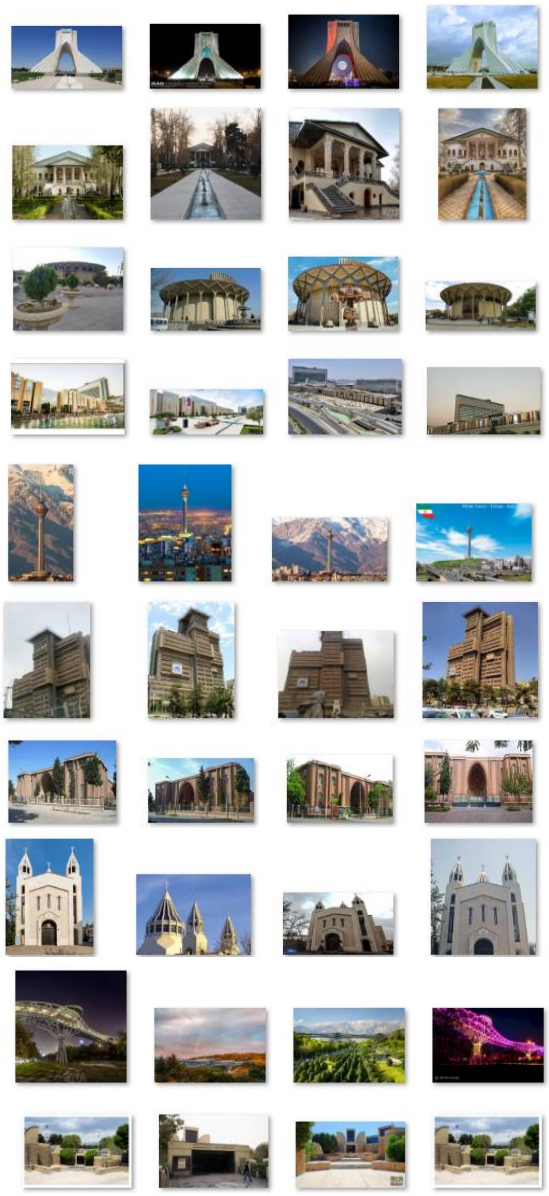
با آموزش شبکه های عصبی پیچشی معمول و خفیف، نهایتاً به یک شبکه تعلیم دیده دست می یابیم که می‌توان آن را در یک سرویس پردازشی تحت وب استفاده نمود. معماری چنین سرویس وبی در شکل ۳ نمایش داده شده است. در این معماری بخش کلاینت از طریق اینترنت به سرویس وب ارائه شده در فضای ابری Render دسترسی داشته و با ارسال درخواست، پاسخ لازم را می‌گیرد. در سمت سرور، یک برنامه کاربردی به زبان پایتون بر اساس فناوری Flask [۲۱] طراحی می شود که برای ذخیره سازی اطلاعات نمادهای زمینی از یک پایگاه داه SQL استفاده میکند. همچنین مدل بهینه تعلیم دیده در یک فایل با ساختار *.h5 درون این معماری استفاده می‌شود. با درخواست کاربر به صورت یک فایل تصویری، اطلاعات نماد تشخیص داده شده در قالب json به وی باز می‌گردد.



شکل ۳- معماری سرویس وب تشخیص نمادهای زمینی

۲-۳- داده‌های تصویری

برای این پژوهش، به صورت نمونه ۱۰ نماد زمینی در شهر تهران انتخاب گردید. برای هر نماد ۴۰ تصویر، که عمده آن از منابع آنلاین و رسانه های اجتماعی تهیه گردید، جمع آوری شده و یک آرشیو تصویر مناسب شامل ۴۰۰ تصویر جهت استفاده در مدل یادگیری عمیق بدست آمد. نمادهای انتخاب شده عبارتند از برج آزادی، باغ فردوس، تئاتر شهر، ایران مال، برج میلاد، وزارت کشور، موزه ملی ایران، پل طبیعت، موزه هنرهای معاصر و کلیسای سرکیس مقدس. نمونه کوچکی از این منابع تصویری در شکل ۳ مشاهده می‌شود.



شکل ۴- نمونه کوچکی از منابع تصویری استفاده شده در فرآیند یادگیری

۳- نتایج و بحث

۳-۱- مقایسه شبکه های عصبی پیچشی

در اجرای یادگیری عمیق، مقادیر پیکسل‌های تصاویر به مقیاس ۰ تا ۱ تبدیل می‌شوند تا نرمالسازی صورت گیرد. در CNN به صورت پیش فرض همه تصاویر به ابعاد ۱۵۰ در ۱۵۰ (بهترین نتیجه ابعاد تصویری با آزمون و خطا) تغییر اندازه یافته و به دسته‌های آموزشی و اعتبارسنجی تقسیم گردید. داده‌ها از پوشه‌ی اصلی خوانده شده و به صورت تصادفی به دو مجموعه ۸۰ درصدی برای آموزش و ۲۰ درصدی برای تست تقسیم می‌شوند. در مرحله دوم، مدل یادگیری عمیق و جزئیات آن تعریف شد. تلاش گردید یک مدل قدرتمند با تعداد لایه های میانی بالا طراحی و اجرا گردد. مدل شامل چندین لایه‌ی پیچشی، لایه‌های مکس‌پولینگ، یک لایه‌ی تخت‌کننده، یک لایه‌ی چگال و یک لایه‌ی دراپ‌اوت است. از تابع فعال‌سازی relu در لایه‌های کانولوشن و از تابع softmax در لایه‌ی خروجی استفاده شده است. همچنین تعداد نمونه‌ها در هر دسته برای آموزش و تست ۳۲ نمونه انتخاب گردید. مدل با استفاده از تولیدکننده داده‌های آموزش، به مدت ۲۰۰ دوره آموزش داده می‌شود. در هر دوره، عملکرد مدل با استفاده از داده‌های تست ارزیابی می‌شود. تعداد چهار لایه کانولوشن با فیلترهای ۳۲، ۶۴، ۱۲۸ و ۱۲۸ که ابعاد هر فیلتر ۳×۳ می باشد، انتخاب گردید. در این پژوهش، تابع فعال‌سازی relu در لایه های کانولوشن استفاده شد. همچنین، تعداد چهار لایه مکس‌پولینگ برای کاهش ابعاد ویژگی‌ها بکار رفت. از لایه تخت برای تبدیل داده‌های دو بعدی به یک بعدی جهت ورود به لایه‌های چگال استفاده شد. در مدل ایجاد شده، یک لایه چگال با ۵۱۲ نرون و تابع فعال‌سازی relu وجود دارد. همچنین، یک لایه خروجی با تعداد نرون‌های برابر با تعداد کلاس‌ها با تابع فعال‌سازی softmax در انتها استفاده می شود. برای کاهش بیش‌برازش یک لایه دراپ‌اوت نیز با نرخ دراپ‌اوت ۰٫۵ در نظر گرفته شد.

در زمان یادگیری نرخ یادگیری اولیه روی ۰٫۰۰۱ روی تنظیم گردید. و نهایتا مدل و ضرایب آن جهت استفاده بعدی در سیستم ذخیره سازی شد. جزئیات مربوط به پارامترهای مدل یادگیری عمیق در جدول ۱ مشاهده می‌شود.

جدول ۱- پارامترهای تنظیم‌شده در مدل CNN

پارامتر	نوع یا مقدار
ابعاد ورودی تصاویر	۱۵۰×۱۵۰
لایه کانولوشن اول	۳۲×۱۴۸×۱۴۸
لایه مکس پولینگ اول	۳۲×۷۴×۷۴
لایه کانولوشن دوم	۶۴×۷۲×۷۲
لایه مکس پولینگ دوم	۶۴×۳۶×۳۶
لایه کانولوشن سوم	۱۲۸×۳۴×۳۴
لایه مکس پولینگ سوم	۱۲۸×۱۷×۱۷
لایه کانولوشن چهارم	۱۲۸×۱۵×۱۵
لایه مکس پولینگ چهارم	۱۲۸×۷×۷
لایه تخت	Flatten با خروجی ۶۲۷۲ مقدار
لایه چگال	Dense با ۵۱۲ نورون و فعال‌ساز ReLU
لایه دراپ‌اوت	Dropout با ۵۱۲ نورون با نرخ ۰.۵
لایه خروجی	Dense با تعداد نورون‌های برابر با تعداد کلاس‌ها و فعال‌ساز Softmax
نرخ یادگیری	۰.۰۰۱
تعداد دوره‌ها (epochs)	۸۰
اندازه دسته (Batch Size)	۳۲
درصد مجموعه آموزش	۸۰٪
درصد مجموعه آزمون	۲۰٪
تعداد کل تصاویر	۴۰۰
معیارهای ارزیابی	Precision Recall Accuracy

فرآیند آموزش با تنظیمات فوق و برای ۳۰۰ دوره تکرار انجام شد که حدوداً به مدت ۹۰ دقیقه به طول انجامید. ۸۰ درصد داده ها برای آموزش و ۲۰٪ برای آزمون استفاده گردید. در شکل های ۵ و ۶ به ترتیب نمودار تغییرات صحت و خطا به ازای فرایند تکرار برای هر دو مجموعه آموزش و آزمون مشاهده می‌شود. همانگونه که ملاحظه می‌شود، مدل یادگیری عمیق به صحت خوبی برای هر دو مجموعه آموزش و آزمون با گذشت تکرارها و اصلاح پارامترهای شبکه دست می یابد. بهترین صحت در حدود ۰.۷۱ و کمترین خطا در حدود ۰٫۸ برآورد گردید.

در جدول ۲ نتایج شاخص های ارزیابی دقت، فراخوانی و امتیاز f1 برای مجموعه تست مشاهده می‌شود. شکل ۷ نیز جدول ابهام حاصل برای داده های آزمون را به صورت گرافیکی نمایش می‌دهد. مقادیر تیره تر در قطر اصلی نشان‌دهنده پیش بینی صحیح تر به ازای هر کلاس می باشند.

جدول ۲- شاخص‌های ارزیابی بدست‌آمده برای مجموعه تست با شبکه CNN

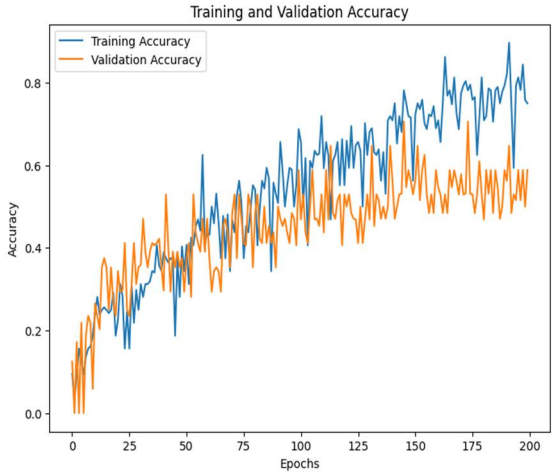
Class Name	Precision	Recall	F1-Score
AzadiTower	0.50	0.50	**0.50**
BaghFerdows	0.62	1.00	**0.76**
CityTheater	0.50	0.62	**0.56**
IranMall	0.73	0.89	**0.80**
MiladTower	0.60	0.75	**0.67**
MinistryOfInterior	1.00	0.75	**0.86**
NationalMuseumOfIran	0.50	0.25	**0.33**
SaintSarkisCathedral	0.50	0.38	**0.43**
TabiatBridge	0.57	0.50	**0.53**
TehranMuseumOfContemporaryArt	0.83	0.62	**0.71**

در الگوریتم MobileNetV2 در ابتدا، تصاویر ورودی با ابعاد (۲۲۴,۲۲۴,۳) به شبکه داده می‌شوند، جایی که ۲۲۴ نشان‌دهنده طول و عرض تصویر و ۳ نشان‌دهنده تعداد کانال‌های رنگی است. اولین گام در پردازش تصاویر، عبور از لایه‌های اولیه MobileNetV2 است که این لایه‌ها با استفاده از فیلترهای کانولوشنی سبک و محاسبات بهینه شده طراحی شده‌اند. نکته قابل توجه راجع به ابعاد ورودی آن است که MobileNetV2 و بسیاری از مدل‌های پیش‌آموزش‌یافته بر اساس ابعاد استاندارد ۲۲۴×۲۲۴ آموزش دیده‌اند. اگر از ابعاد کوچکتر مانند ۱۵۰×۱۵۰ استفاده می‌شد، برخی از ویژگی‌های مهم موجود در تصاویر ممکن بود از بین برود.

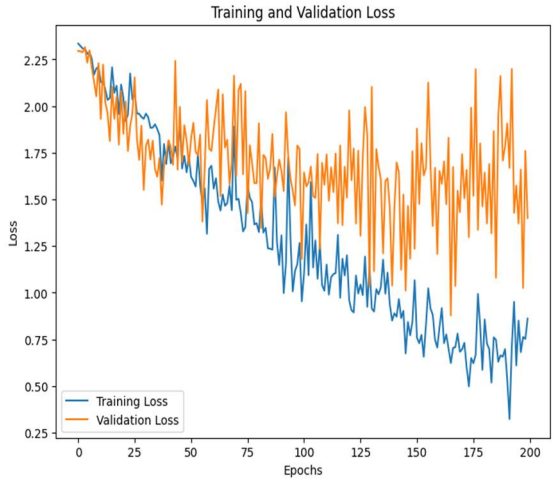
MobileNetV2 از بلوک‌های Inverted Residuals استفاده می‌کند. این بلوک‌ها به این شکل طراحی شده‌اند که ابتدا ویژگی‌های ورودی را به فضای دارای ابعاد بالاتر منتقل می‌کنند و سپس با استفاده از Depthwise Separable Convolutions پردازش می‌شوند. در نهایت، ویژگی‌ها دوباره به فضای دارای ابعاد پایین‌تر فشرده می‌شوند. این طراحی به شبکه اجازه می‌دهد که اطلاعات مفید را بدون افزایش غیرضروری پیچیدگی محاسباتی حفظ کند. برای هر بلوک Inverted Residual، ابعاد ویژگی‌ها به گونه‌ای تنظیم می‌شود که همواره تناسب بین کاهش پیچیدگی و حفظ دقت مدل برقرار باشد.

در این پژوهش، لایه‌های MobileNetV2 به طور کامل ثابت[1] شده‌اند. این بدان معنی است که وزن‌های این لایه‌ها در حین آموزش تغییر نمی‌کنند و فقط از آنها برای استخراج ویژگی‌های عمومی استفاده می‌شود. این ویژگی‌های عمومی شامل لبه‌ها، بافت‌ها، و اشکال ساده است که مدل از طریق آموزش روی مجموعه داده ImageNet فرا گرفته است. به

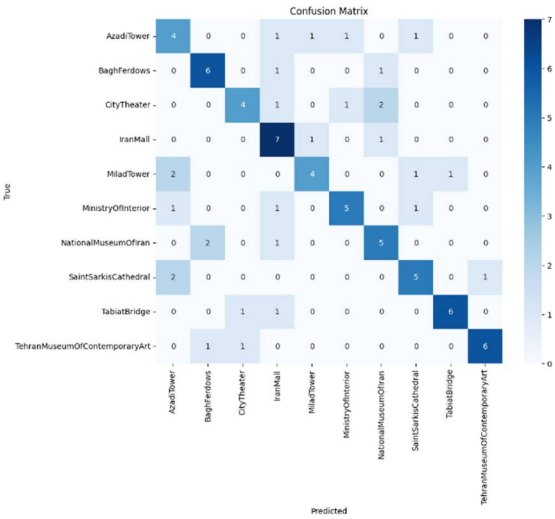
۱ freeze



شکل ۵- نمودار صحت داده های آموزش و آزمون در تکرارهای الگوریتم CNN



شکل ۶- نمودار خطای داده های آموزش و آزمون در تکرارهای الگوریتم CNN



شکل ۷- جدول ابهام برای کلاس‌های مختلف نمادهای زمینی و بر اساس داده های آزمون با CNN

عنوان مثال، یک لایه کانولوشنی اولیه ممکن است ابعاد ویژگی‌هایی با عمق کمتر (مثلاً (۱۱۲,۱۱۲,۳۲)) تولید کند، در حالی که لایه‌های عمیق‌تر ویژگی‌هایی با عمق بیشتر اما ابعاد مکانی کوچکتر تولید می‌کنند.

بعد از عبور از لایه‌های MobileNetV2، ویژگی‌های خروجی به لایه GlobalAveragePooling2D ارسال می‌شوند. این لایه ابعاد مکانی ویژگی‌ها را با میانگین‌گیری کاهش می‌دهد و یک بردار تک‌بعدی تولید می‌کند که اندازه آن برابر با تعداد کانال‌های خروجی است. این بردار فشرده‌شده اطلاعات کلیدی را از تصویر حفظ می‌کند، در حالی که تعداد پارامترها را به طور قابل توجهی کاهش می‌دهد. به عنوان مثال، اگر ویژگی‌های خروجی از لایه‌های MobileNetV2 ابعادی مانند (۷,۷,۱۲۸۰) داشته باشند، لایه GlobalAveragePooling2D آن را به یک بردار با ابعاد (۱۲۸۰) فشرده می‌کند.

در ادامه، لایه Dropout با نرخ ۵۰ درصد قرار داده شده است. این لایه به طور تصادفی نیمی از نورون‌ها را غیرفعال می‌کند تا از بیش‌برازش جلوگیری کند. این عمل باعث می‌شود مدل به سمت یادگیری بهتر و تعمیم بیشتر روی داده‌های دیده نشده حرکت کند. در نهایت، یک لایه Dense با تعداد نورون‌هایی که برابر با تعداد کلاس‌های موردنظر برای طبقه‌بندی است، اضافه شده است. این لایه از تابع فعال‌سازی softmax استفاده می‌کند که احتمال‌های هر کلاس را محاسبه کرده و خروجی نهایی را به شکل توزیع احتمالاتی ارائه می‌دهد.

به طور کلی، این معماری با استفاده از طراحی سبک و بهینه MobileNetV2 همراه با لایه‌های سفارشی، قادر به ارائه عملکرد بالا با هزینه محاسباتی کم است. این رویکرد نه تنها سرعت پردازش را افزایش می‌دهد بلکه امکان استفاده از مدل را در دستگاه‌هایی با توان محاسباتی پایین‌تر نیز فراهم می‌کند. جزئیات بیشتر نیز در جدول ۳ مشاهده می‌شود.

فرآیند آموزش با تنظیمات فوق و برای ۳۰ دوره تکرار انجام شد که حدوداً به مدت ۵ دقیقه به طول انجامید. ۸۰ درصد داده ها برای آموزش و ۲۰ درصد برای آزمون استفاده گردید. در شکل های ۸ و ۹ به ترتیب نمودار تغییرات صحت و خطا به ازای فرایند تکرار برای هر دو مجموعه آموزش و آزمون مشاهده می شود. همانگونه که ملاحظه می شود، مدل یادگیری عمیق MobileNetV2 به صحت بالایی برای هر دو مجموعه آموزش و آزمون با گذشت تکرارها و اصلاح پارامترهای شبکه دست می یابد.

بهترین صحت در حدود ۰/۹۲ و کمترین خطا در حدود ۰/۰۱ برآورد گردید. در جدول ۴ نتایج شاخص های ارزیابی دقت، فراخوانی و امتیاز f1 برای مجموعه تست مشاهده می‌شود. شکل ۱۰ نیز جدول ابهام حاصل برای داده‌های آزمون را به صورت گرافیکی نمایش می‌دهد. مقادیر تیره تر در قطر اصلی نشان‌دهنده پیش‌بینی صحیح‌تر به ازای هر کلاس می باشند.

جدول ۳- پارامترهای تنظیم شده در مدل یادگیری عمیق با MobileNetV2

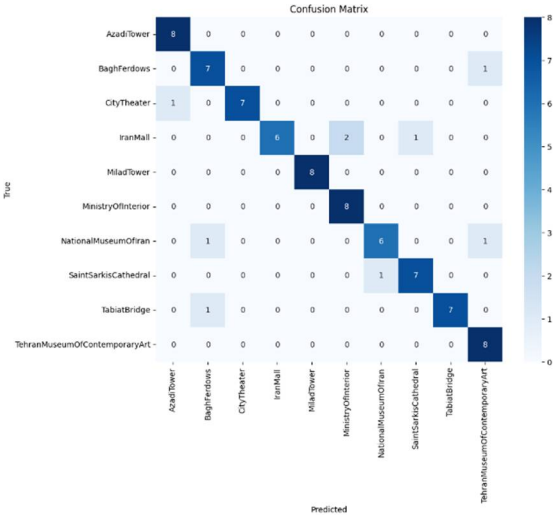
پارامتر	نوع یا مقدار
ابعاد ورودی تصاویر	۲۲۴×۲۲۴
ابعاد لایه‌های پیچشی	۱۲۸۰×۷×۷
لایه‌های پیچشی اولیه	چندین لایه کانولوشن با فیلترهای سبک Depthwise و Pointwise با فعال‌ساز ReLU6، ابعاد خروجی مختلف بسته به لایه (مانند ۱۱۲×۱۱۲×۳۲ یا ۵۶×۵۶×۶۴)
بلوک‌های Inverted Residual	شامل چندین بلوک Inverted Residual که در آن‌ها ابعاد ویژگی‌ها تغییر می‌کند، مانند: ۲۸×۲۸×۹۶، ۱۴×۱۴×۱۶۰، ۷×۷×۳۲۰
لایه Global Average Pooling	کاهش ابعاد به یک بردار با ابعاد ۱×۱×۱۲۸۰
لایه Dropout	Dropout با نرخ ۰.۵ برای جلوگیری از بیش‌برازش
لایه چگال (Dense)	Dense با تعداد نورون‌های برابر با تعداد کلاس‌ها و فعال‌ساز Softmax
فعال‌سازهای لایه‌ها	ReLU6در لایه‌های میانی و Softmax در لایه خروجی
نرخ یادگیری	۰.۰۰۱
تعداد دوره‌ها (epochs)	۳۰
اندازه دسته (Batch Size)	۳۲
درصد مجموعه آموزش	%۸۰
درصد مجموعه آزمون	%۲۰
معیارهای ارزیابی	Precision Recall Accuracy

جدول ۴- شاخص‌های ارزیابی بدست آمده برای مجموعه تست با شبکه MobileNetV2

Class Name	Precision	Recall	F1-Score
AzadiTower	0.89	1.00	**0.94**
BaghFerdows	0.78	0.88	**0.82**
CityTheater	1.00	0.88	**0.92**
IranMall	1.00	0.67	**0.80**
MiladTower	1.00	1.00	**1.00**
MinistryOfInterior	0.80	1.00	**0.89**
NationalMuseumOfIran	0.86	0.75	**0.80**
SaintSarkisCathedral	0.88	0.88	**0.88**
TabiatBridge	1.00	0.88	**0.92**
TehranMuseumOfContemporaryArt	0.80	1.00	**0.89**

شکل ۸- نمودار صحت داده های آموزش و آزمون در تکرارهای الگوریتم موبایل نت

شکل ۹- نمودار خطای داده های آموزش و آزمون در تکرارهای الگوریتم موبایل نت



شکل ۱۰- جدول ابهام برای کلاس های مختلف نمادهای زمینی و بر اساس داده های آزمون با موبایل نت

با مقایسه نتایج شبکه MobileNetV2 آموزش دیده به دلیل عملکرد بهتر در متریک های مختلف، به عنوان مدل منتخب در سیستم نهایی استفاده خواهد شد. این مدل همچنین در مدت زمان کمتری نسبت به یک مدل پایه CNN به همگرایی رسیده و در صورت توسعه و تغییر سیستم به سرعت بالاتری می‌توان آن را به هنگام نمود. این زمان کمتر در حالتی است که تعداد ابعاد ورودی تصویر در CNN برابر ۱۵۰ تنظیم شده بود و این مقدار در MobileNetV2 برابر مقدار پیش فرض ۲۲۴ است و این افزایش جزئیات می‌تواند فرآیند آموزش را کند نماید. همچنین با مقایسه ماتریس ابهام دو روش مشاهده می‌شود که بسیاری از نمونه‌های تست در روش MobileNetV2 به طور کامل صحیح پیش بینی شده‌اند.

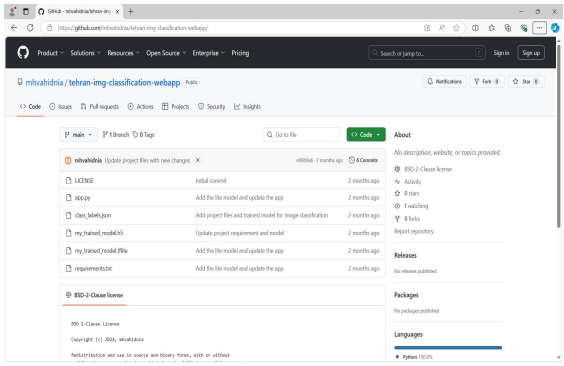
۳-۲- نتایج پیاده‌سازی سرویس وب پردازش تصویر با یادگیری عمیق

پس از بدست آوردن مدل بهینه این مدل در ساختار h5.* ذخیره و سپس برای اجرای سامانه خروجی گرفته شد. در مرحله بعد ایجاد یک سرویس پردازش تصاویر به کمک شبکه یادگیری شده در دستور کار قرار گرفت. مدل بهینه ابتدا به صورت محلی در یک سیستم عامل ویندوز مورد آزمایش قرار گرفت. هدف این بود که با یک عکس جدید از ۱۰ نماد زمینی مورد نظر، خارج از نمونه‌های استفاده شده برای آموزش و آزمون، برچسب آن توسط مدل پیش بینی شود. بدین منظور یک پروژه پایتون توسعه یافت که مهمترین بسته‌های مورد استفاده در آن عبارت بودند از tensorflow، flask و pillow. از چارچوب Flask برای راه‌اندازی یک وب سرویس استفاده شد که می‌تواند بر روی یک پلتفرم ابری یا به صورت محلی مستقر شود.

در مرحله بعد، سرویس وب ساخته شده تحت Flask را به یک خدمات ارائه‌دهنده سرویس ابری منتقل نمودیم. برای این منظور گزینه‌های فناوری مختلفی مانند Heroku، Google Cloud Run، Koyeb، AWS Elastic Beanstalk، Render و موارد دیگر قابل استفاده بودند. اما Render به دلیل قابلیت ارائه سرویس وب به صورت رایگان و همچنین سادگی انتخاب گردید. پیش از آن نیاز بود یک مخزن کد در GitHub ایجاد شود. زیرا در ایجاد سرویس‌های وب Render می‌توان از منابع GitHub به صورت مستقیم استفاده کرد. فایل‌های مهم پروژه از جمله

که app.py, requirements.txt, my_trained_model.h5 به ترتیب شامل برنامه ایجاد سرویس وب flask، مدل بهینه یادگیری عمیق و بسته‌های مورد نیاز پایتون بودند، به کمک پلتفرم git از سیستم محلی روی مخزن آنلاین منتقل و به هنگام شدند. این مخزن در حال حاضر در آدرس زیر قابل دسترس است (شکل ۱۱):

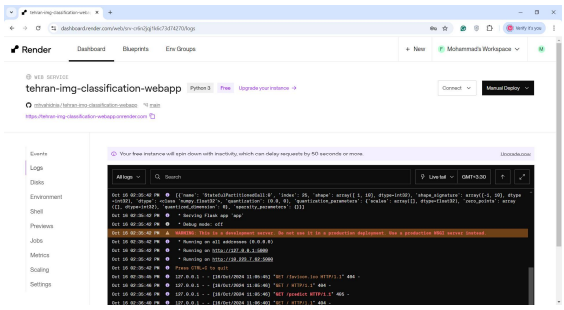
https://github.com/mhvahidnia/tehran-img-classification-webapp/



شکل ۱۱- مخزن Github برای منابع ساخت سرویس آنلاین شناسایی تصاویر نمادهای تهران با یادگیری عمیق

در ادامه در Render سرویس وب آنلاین با بکارگیری منابع ارائه شده در Github ساخته شد. از نکات قابل توجه، محدویت RAM به ۵۱۲ MB و فضای ذخیره‌سازی محدود در ایجاد سرویس‌های رایگان Render می‌باشد. برای اجرای مدل یادگیری عمیق لازم بود نسخه ای فشرده از مدل یا به عبارتی TensorFlow Lite model، ایجاد شود که با RAM محدود در سرویس رایگان ابری نیز قابل اجرا باشد. بنابراین، با توسعه یک برنامه پایتون فایل مدل به ساختار tflite.* تبدیل یافته و در مخزن جایگزین شد. نهایتاً با انجام تنظیمات، از جمله تنظیمات Environment و نصب بسته-های پیش نیاز و برنامه اصلی از مخزن Github، سرویس وب طبقه‌بندی نمادهای تهران با یادگیری عمیق مطابق شکل ۱۲ در محیط Render ایجاد شد. این سرویس در حاضر از آدرس زیر قابل فراخوانی و استفاده می‌باشد که البته محدود به ۱۰ نماد مد نظر در طرح می‌باشد:

https://tehran-img-classification-webapp.onrender.com/predict



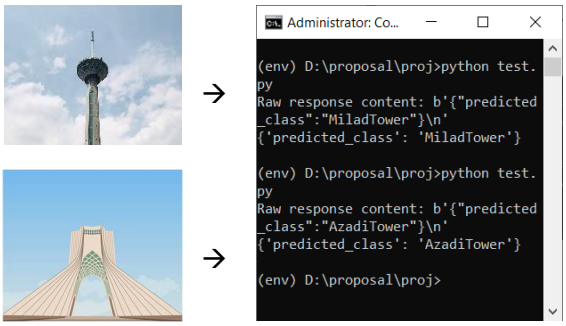
شکل ۱۲ – ایجاد سرویس وب طبقه بندی نمادهای تهران با یادگیری عمیق در محیط Render

ورودی سرویس یک عکس (مثلاً در فرمت jpg) و خروجی آن یک json حاوی اطلاعات برچسب نماد زمینی تشخیص داده شده می‌باشد. شکل ۱۳ نمونه‌ای از یک کد پایتون برای فراخوانی و پیش‌بینی نماد با معرفی یک تصویر دلخواه از سمت کلاینت می‌باشد. درخواست (Request) پس از ارسال با شیوه post ارسال شده و محتوای پاسخ (response) در قالب json چاپ می‌شود.

```
import requests
# Path to the image you want to predict
image_path = 'azadi.jpg'
# Send a POST request to the Flask server with the
image file
with open(image_path, 'rb') as img:
    response = requests.post('https://tehran-img-
classification-webapp.onrender.com/predict',
files={'file': img})
print("Raw response content:",
response.content) | # Print raw response content
try:
    print(response.json())  # Try to parse JSON
except requests.exceptions.JSONDecodeError as e:
    print("JSON decode error:", e)
```

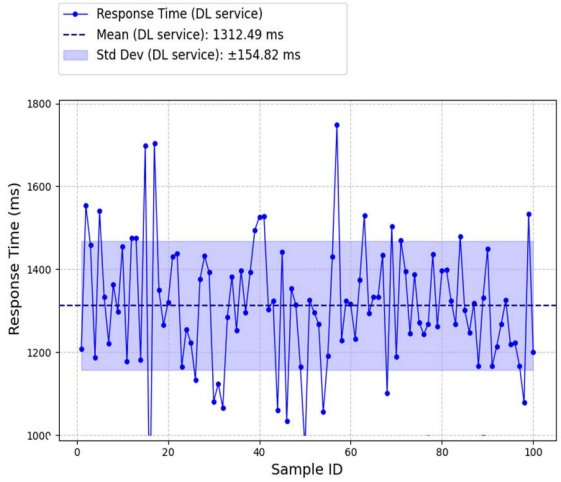
شکل ۱۳- یک نمونه کد پایتون برای فراخوانی سرویس وب بازیابی نماد

در شکل ۱۴ نمونه‌هایی از آزمایش سرویس آنلاین وب در یک پروژه پایتون محلی در محیط command prompt را با تصاویری خارج از مجموعه داده استفاده شده برای آموزش و آزمایش نشان می‌دهد. در این نمونه‌ها برچسب و اطلاعات برج آزادی و میلاد، به عنوان نمادهای تشخیص داده شده، با موفقیت از سرویس بازگردانده شده است.



شکل ۱۴- آزمایش سرویس وب به صورت آنلاین و گرفتن برچسب تصویر از فایل json

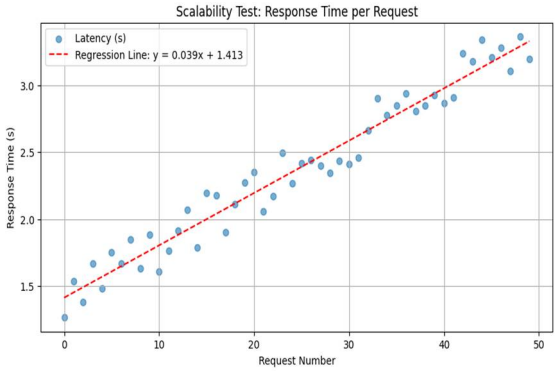
یکی از عناصر محوری در ارزیابی‌ها، زمان تشخیص است که می‌تواند به ویژه برای کاربران نهایی و گردشگران حیاتی باشد. یک ارزیابی زمانی برای سرویس پردازشی با ۱۰۰ تکرار انجام شد. زمان از لحظه ارسال عکس تا لحظه پاسخگویی سرور در نظر گرفته شد. همانطور که در شکل ۱۴ نشان داده شده است، مقادیر با میانگین زمان ۱۳۱۲.۴۹ میلی ثانیه و انحراف استاندارد ۱۵۴.۸۲ میلی ثانیه بدست آمد که نوسان چندانی را نشان نداده و رضایت‌بخش بودند.



شکل ۱۵- ارزیابی زمانی کشف نماد زمینی با استفاده از سرویس پردازش تصویر وب مبتنی بر یادگیری عمیق

به منظور ارزیابی مقیاس‌پذیری سرویس پردازش وب، چندین درخواست به صورت همزمان به سرور ارسال گردید و با تغییر تعداد درخواست‌ها از کم به زیاد مدت زمان پاسخگویی به هر درخواست اندازه‌گیری گردید. در این آزمایش ماکزیمم درخواست‌های همزمان برابر ۵۰ در نظر گرفته شد و درخواست‌های همزمان به ترتیب از ۱ تا ۵۰ کاربر افزایش یافت.

نتایج در شکل ۱۶ نمایش داده شده است. در این آزمایش در حدود ۹۰٪ درخواست‌ها پاسخ موفق از سمت سرور دریافت شد. به لحاظ زمانی یک افزایش زمانی خطی با شیب ملایم (۰/۰۳۹) با افزایش کاربران مشاهده شد. با توجه به اینکه در تحلیل پیچیدگی محاسباتی روند خطی، پیچیدگی بالایی محسوب نمی‌شود، می‌توان این نتیجه را امیدبخش قلمداد نمود. همچنین برای ۳۰ کاربر مدت زمان پاسخگویی سرور کمتر از ۴ ثانیه برآورد گردید که قابل قبول می‌باشد.



شکل ۱۶- ارزیابی مقیاس پذیری سرویس پردازش تصویر وب با افزایش کاربران

۴- نتیجه گیری

در این پژوهش یک سرویس پردازشی وب برای تشخیص نمادهای زمینی و مقاصد گردشگری بر اساس شبکه‌های عصبی پیچشی طراحی شد. زیرساخت نمونه مذکور می‌تواند به عنوان الگویی اولیه برای توسعه برنامه‌های کاربردی و خدمات مکان-مبنا باشد. طبق نتایج مدل MobileNetV2 که از پیش آموزش اولیه دیده است و به علت پارامترهای کمتر، سبک وزن به حساب می‌آید، به دلیل سرعت و کارایی برتر، انتخاب بهینه‌ای برای برنامه‌های کاربردی بلادرنگ محسوب می‌شود. مدل خفیف با وجود پارامترهای کمتر، به دلیل لایه پیش آموزش دیده، عملکرد بهتری در تمامی شاخص‌های صحت، دقت، و فراخوانی نشان داد. نتایج ارزیابی زمانی و مقیاس پذیری بر روی یک سرور ابری رایگان، کاملاً قابل قبول برآورد شد، هرچند برای تجاری‌سازی، سرورهای ابری با حافظه، فضای ذخیره‌سازی و قدرت پردازش بالاتری پیشنهاد می‌شود.

در این مطالعه عمده تصاویر استفاده شده در مرحله آموزش و تست مربوط به زمان روز و نور کافی بودند. به

عنوان مطالعات آتی می‌توان به قدرت استخراج لندمارک‌ها در فضای شهری با توجه به شرایط نوری مختلف زمان تصویربرداری پرداخت. چالش قابل آزمایش دیگر برای مطالعات آتی، قدرت مدل در تشخیص و تمایز نمادهای بسیار مشابه، به عنوان نمونه مساجد، یا هرگونه سازه های مشابه دیگر، می باشد. همچنین تلفیق سرویس پردازشی پیشنهادی با برنامه های کاربردی نقشه محور روی گوشی‌های هوشمند برای تجربه بهتر کاربر و توسعه خدمات گردشگری پیشنهاد می‌شود.

سپاسگزاری

این اثر تحت حمایت مادی صندوق حمایت از پژوهشگران و فناوران کشور (INSF) برگرفته شده از طرح شماره ۴۰۲۱۸۳۲ انجام شده است.

مراجع

[۱] Gössling, S. (2017). Tourism, information technologies and sustainability: an exploratory review. Journal of Sustainable Tourism, 25(7), 1024-1041.

[۲] Vahidnia, M. H., Minaei, M., & Behzadi, S. (2024). An ontology-based web decision support system to find entertainment points of interest in an urban area. Geo-Spatial Information Science, 27(2), 505-522.

[۳] Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. Journal of network and computer applications, 39, 319-333.

[۴] Yoder, R. M., Clark, B. J., & Taube, J. S. (2011). Origins of landmark encoding in the brain. Trends in neurosciences, 34(11), 561-571.

[۵] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[۶] Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. Neurocomputing, 396, 39-64.

[۷] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.

[۸] Sharma, V. K., & Mir, R. N. (2020). A comprehensive and systematic look up into deep learning based object detection techniques: A review. Computer Science Review, 38, 100301.

[۹] Samany, N. N. (2019). Automatic landmark extraction from geo-tagged social media photos using deep neural network. Cities, 93, 1-12.

[۱۰] Chaowanawatee, K., Silanon, K., & Kliangsuwan, T. (2022, December). Phuket Landmark Recognition using Fine-Tuned Convolutional Neural Network. In 2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter) (pp. 257-261). IEEE.

[۱۱] Razali, M. N., Tony, E. O. N., Ibrahim, A. A. A., Hanapi, R., & Iswandono, Z. (2023). Landmark recognition model for smart tourism using lightweight deep learning and linear discriminant analysis. International Journal of Advanced Computer Science and Applications, 14(2).

[۱۲] Madake, J., Padwal, A., Pande, Y., Nevase, P., & Bhatlawande, S. (2024, June). Advancing Outdoor Landmark Detection: A Vision Transformer Approach. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[۱۳] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[۱۴] Barbosa, I. B., Cristani, M., Caputo, B., Rognhaugen, A., & Theoharis, T. (2018). Looking beyond appearances: Synthetic training data for deep CNNs in re-identification. Computer Vision and Image Understanding, 167, 50-62.

[۱۵] Xue, H., Artico, J., Fontana, M., Moon, J. C., Davies, R. H., & Kellman, P. (2021). Landmark detection in cardiac MRI by using a convolutional neural network. Radiology: Artificial Intelligence, 3(5), e200197.

[۱۶] Ekvall, M., Bergenstråhle, L., Andersson, A., Czarnewski, P., Olegård, J., Käll, L., & Lundeberg, J. (2024). Spatial landmark detection and tissue registration with deep learning. Nature Methods, 21(4), 673-679.

[۱۷] Li, H., Guo, Z., Rhee, S. M., Han, S., & Han, J. J. (2022). Towards accurate facial landmark detection via cascaded transformers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4176-4185).

[۱۸] Bodini, M. (2019). A review of facial landmark extraction in 2D images and videos using deep learning. Big Data and Cognitive Computing, 3(1), 14.

[۱۹] Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020, December). MobileNetV2 model for image classification. In 2020 2nd International Conference on Information Technology and Computer Application (ITCA) (pp. 476-480). IEEE.

[۲۰] Wang, X., & Li, H. (2023). Investigation of Deep Learning Based Semantic Segmentation Models for Autonomous Vehicles. International Journal of Advanced Computer Science & Applications, 14(11).

[۲۱] Idris, N., Foozy, C. F. M., & Shamala, P. (2020). A generic review of web technology: Django and flask. International Journal of Advanced Science Computing and Engineering, 2(1), 34-40.