مقاله پژوهشی

مجله ماشین‌بینایی و پردازش‌تصویر
سال دهم، شماره چهارم، زمستان ۱۴۰۲

یک معماری جدید از شبکهYOLOv5 با بکارگیری مکانیسم توجه برای بهبود مصالحه دقت-سرعت در آشکارسازی میوه سیب

مرضیه محمودی‌فر[۱] و ندا فرجی[۲]

چکیده

آشکارسازی میوه با توجه به شرایط روشنایی متفاوت، انسداد و همپوشانی یک کار چالش‌برانگیز در ربات‌های برداشت مبتنی بر بینایی ماشین است. هدف از این مقاله بهبود مصالحه دقت-سرعت در آشکارسازی میوه سیب در سیستم بینایی ربات‌های برداشت کننده کشاورزی است. با توجه به کاربردهای اخیر ماژول‌های توجه در زمینه آشکارسازی شیء، معماری جدیدی از شبکه YOLOv5 پیشنهاد شده است که در آن ماژول توجه کانالی ECA در ستون فقرات شبکه، جایگزین ماژول C3 شده است. ماژول ECA علی‌رغم کاهش تعداد پارامترهای شبکه اثر قابل توجهی در کارایی آشکارسازی نداشت،و با افزایش سرعت به میزان% ۲۲نسبت به YOLOv5 نسخه نانو، توانست مصالحه بهتری بین دقت و سرعت برقرار کند. برای ارزیابی معماری پیشنهادی از سه نوع مجموعه داده KFuji، MinneApple و ACFR در مرحله آموزش و آزمون استفاده شد و در حالتی که پایگاه داده آموزش و آزمون یکی نبودند، روش یادگیری انتقالی برای بهبود نتایج آزمون به کار گرفته شد. در حالتی که داده‌های آموزش و آزمون یکی بودند، استفاده از معماری پیشنهادی منجر به بهبود نسبی عدد مصالحه به میزان ۲۱,۲% در مقایسه با ماژول C3 شد و در حالت یادگیری انتقالی که داده‌های آموزش و آزمون یکی نبودند، بهبود نسبی ۱۸% در عدد مصالحه به دست آمد.

کلید واژه‌ها

YOLOv5، آشکارسازی، مصالحه دقت-سرعت، ماژول توجه، یادگیری انتقالی.

۱- مقدمه

کشاورزی در رشد اقتصادی هر کشور نقشی حیاتی دارد. با افزایش جمعیت، تغییر مکرر در شرایط آب و هوایی و منابع محدود، تکمیل نیاز غذایی جمعیت حاضر یک کار چالش‌برانگیز است.کشاورزی دقیق یکی از زیر ساخت‌های کشاورزی هوشمند، ابزاری ابتکاری برای رفع چالش‌های موجود در کشاورزی سنتی[۱] و یکی از راه‌حل‌های تضمین امنیت غذایی برای کل جهان است [۲و۳].

استفاده از بینایی ماشین و الگوریتم‌های مرتبط با آن، کارایی، عملکرد، هوشمندی و تعامل از راه دور ربات‌های برداشت را در محیط‌های پیچیده کشاورزی بهبود می‌بخشد[۴]. سیستم برداشت هوشمند در مقایسه با روش‌های برداشت سنتی مزایای متعددی دارد، مانند نیاز به نیروی کار کمتر، بینش بهتر نسبت به محصولات، کاهش هزینه برداشت و تولید مقرون به صرفه [۱]. در حال حاضر در کشور ما ربات‌های کشاورزی پیشرفت چندانی نداشته‌اند اما برخی ربات‌های برداشت محصولات گلخانه‌ای مانند ربات برداشت فلفل شیرین در دست ساخت هستند و پروژه هوشمندسازی مزارع شامل استفاده از اینترنت اشیاء، مدیریت داده و دسترسی به GPS در حال اجرا هستند [۵-۷].

اولین وظیفه یک ربات برداشت میوه استفاده از حس بصری برای درک و یادگیری اطلاعات محصول به منظور تشخیص و موقعیت‌یابی هدف، تشخیص پس‌زمینه هدف و

این مقاله در اسفندماه سال ۱۴۰۱ دریافت شد در خردادماه ۱۴۰۲ بازنگری و سپس پذیرفته گردید.

[۱] دانش‌آموخته کارشناسی ارشد مهندسی برق گرایش مخابرات-سیستم، دانشگاه بین‌المللی امام خمینی(ره)، قزوین، ایران
رایانامه: m.mahmoudifar@edu.ikiu.ac.ir

[۲] دانشکده فنی و مهندسی، گروه مهندسی برق، دانشگاه بین‌المللی امام خمینی(ره)، قزوین، ایران
رایانامه: nfaraji@eng.ikiu.ac.ir

نویسنده مسئول: ندا فرجی

بازسازی سه بعدی است [۴].دقت تشخیص میوه و سرعت انجام این فرآیند، در ربات‌های برداشت میوه بسیار مهم است. با این حال، تشخیص میوه بطور کلی تحت تأثیر عوامل زیادی مانند تغییرات نور، انسداد و تشابه ویژگی‌های بصری میوه و پس‌زمینه بوده و بنابراین، یک مدل تشخیص میوه که بتواند به خوبی تعمیم یابد برای غلبه بر این چالش‌ها ضروری است.

بتازگی استفاده از یادگیری عمیق در حوزه آشکارسازی در تصاویر کشاورزی نیز گسترش یافته است [۸]. یکی از مهمترین مزایای استفاده از یادگیری عمیق در پردازش تصویر این است که یادگیری عمیق نیازی به مهندسی ویژگی ندارد و ویژگی‌های مهم را از طریق آموزش تعیین می‌کند[۹و ۱۰]. مهندسی ویژگی یک فرآیند پیچیده و زمانبر است که هر زمان که مسئله یا مجموعه داده تغییر کند باید تغییر یابد و براحتی قابل‌تعمیم نیست[۹].یک آشکارساز یادگیری عمیق تک مرحله‌ای مانند YOLO[1] یا SSD[2]، سریعتر از یک آشکارساز دومرحله‌ای مانند شبکه کانولوشنی مبتنی بر ناحیه سریعتر (Faster R-CNN) با دقت مشابه است. بنابراین بهینه‌سازی معماری‌های یادگیری عمیق تک‌مرحله‌ای به لحاظ سرعت و دقت، کمک شایانی به بهبود عملکردربات‌های برداشت‌کننده در تشخیص میوه می‌نماید[۱۱].

انتخاب یک معماری که به مصالحه مناسب دقت و سرعت منجر شود، برای یک کاربرد و بستر سخت‌افزاری معین مهم است.در برخی موارد سرعت و حافظه حیاتی است که نیازمند معماری‌هایی با سرعت بالا است که بتوان برای مثال روی یک سیستم تلفن همراه پیاده‌سازی کرد. در برخی موارد نیز دقت حیاتی است که باید از معماری‌هایی که دقت بالایی دارند استفاده کرد [۱۲]. آشکارسازهای شیءجدید مانند Faster R-CNN [۱۳]، R-FCN [۱۴]، SSD[۱۵]و پنج نسخه YOLO[۱۶-۲۰] دقت مناسبی دارند که می‌توانند در محصولات مصرفی بکار روند و برخی از آنها نیز برای اجرا در دستگاه‌های تلفن همراه به اندازه کافی سریع هستند [۱۲].

نکته دیگر اینکه انسان‌ها می‌توانند بطور طبیعی و موثر مناطق برجسته را در صحنه‌های بصری پیچیده پیدا کنند. مکانیسم‌های توجه با هدف تقلید از این جنبه از سیستم بینایی انسان وارد بینایی ماشین شدند. در دهه گذشته، مکانیسم‌های توجه نقش مهمی را برای بهبود نتایج و سرعت در بینایی کامپیوتر ایفا کرده‌اند[۲۱]. در چند سال اخیر تلفیق شبکه‌های آشکارساز جدید و مکانیسم توجه کاربرد زیادی در آشکارسازی اشیا داشته است.

آشکارسازی میوه با بهره‌گیری از روش‌های یادگیری عمیق در بسیاری از مطالعات انجام شده است[۱۱و۲۲-۲۷]. بارگوتی و همکارانش شبکه Faster R-CNN با VGG16 که دارای ۱۳ لایه برای استخراج ویژگی از تصاویر رنگی ورودی است،را برای آشکارسازی میوه سیب در پایگاه دادهACFRبکار بردند و به نمره F1 ۹۰,۴% و زمان استنتاج ۰,۱۳ ثانیه برای هر تصویر دست یافتند. البته این روش در خوشه‌های فشرده دقت آشکارسازی پایینی داشته و علی‌رغم دقت خوب در خوشه‌های عادي سرعت آن پایین است [۲۸].تیان و همکارانش برای آشکارسازی میوه در سه کلاس سیب نرسیده، سیب در حال رسیدن و سیب رسیده از شبکه YOLOv3 استفاده کردند که در آن شبکه DenseNet بجای لایه‌های انتقال اصلی برای بهبود انتشار ویژگی‌ها بکار رفته، و مقدار نمره F1 به ۰,۸۱۷ و سرعت به ۰,۳۰۴ ثانیه برای هر تصویر رسیده است[۲۹].در این مطالعه شبکه پیشنهادی با شبکه Faster R-CNN با VGG16[۱۳]،YOLOv2[۱۷]و نسخه اصلی YOLOv3[۱۸] مقایسه شده که در آن شبکه YOLOv2 به نمره F1 ۰,۷۳۸ و سرعت ۰,۲۷۳ ثانیه، شبکه YOLOv3 به نمره F1 ۰,۷۹۳ و سرعت ۰,۲۹۶ ثانیه و شبکه Faster R-CNN با VGG16 به نمره F1 ۰,۸۰۱ و سرعت ۲,۴۲ ثانیه به ازای هر تصویر دست یافته است. با توجه به نتایج این مقایسه شبکه پیشنهادی YOLOv3 با DenseNet مصالحه بهتری میان دقت و سرعت برقرار کرد و علی‌رغم اینکه سه مرحله از بلوغ میوه سیب در این کار بررسی شد اما جای بهبود دقت و سرعت همچنان وجود دارد.

کانگ و همکارانش از آشکارساز تک‌مرحله‌ای LedNet همراه با یک استخراج کننده ویژگی سبک که شامل ۹ بلوک bottleneck ResNet و ۵ بلوک Down-samplingاست برای استخراج ویژگی از تصاویر رنگیبا هدف کاهش حجم محاسبات و آشکارسازی در زمان واقعی استفاده کردند. این روش به نمره F1برابر با ۰,۸۳۴ و زمان آشکارسازی۲۸ میلی ثانیه برای هر تصویر روی میوه سیبدست‌یافته است [۳۰].خنه-مولا و همکارانش از شبکه Faster R-CNN و استخراج کننده ویژگی VGG16روی پایگاه داده KFuji با تصاویر ورودی رنگی-عمق-مادون قرمز استفاده کرده‌اند ونمره F1به ۰,۸۹۸ و سرعت به ۱۳,۶ تصویر در ثانیه رسیده است[۳۱]. در مقایسه با سایر شبکه‌هایی که از Faster R-CNN استفاده کرده‌اندین کار نتایج بهتری را دارد و علت آن استفاده از تصاویر عمق و مادون قرمز بعنوان دو کانال ورودی دیگر در کنار سه کانال مربوط به تصاویر رنگی است.چو و همکارانش از شبکه Mask R-CNN استفاده کردند که به آن یک شاخه سرکوبگر[3] برای سرکوب ویژگی‌های غیر سیب استخراج‌شده توسط شبکه اصلی Mask R-CNN اضافه کردند. این شبکه در آشکارسازی میوه سیب به نمره F1معادل با ۰,۹۰۵ و سرعت ۰,۲۵ ثانیه برای هر تصویر دست‌یافته که دقت خوبی نسبت به دیگر شبکه دومرحله‌ای معروف Faster R-CNN داشته است اما همچنان سرعت

[1]You Only Look Once
[2]Single Shot Detector
[3]Suppression branch

پایینی دارد و جا برای افزایش سرعت برای بهبود مصالحه دقت و سرعت وجود دارد [۳۲].

یان و همکارانش شبکه YOLOv5s را بهبود داده و از آن برای آشکارسازی میوه سیب استفاده کرده‌اند که به نمره F1برابر با ۸۷,۴۹% وزمان استنتاج ۰,۰۱۵ ثانیه برای هر تصویر دست‌یافته است[۳۳]. در این کار برای جلوگیری از آسیب رسیدن به عملگر انتهایی بازوی ربات، آشکارسازی برای دو حالت میوه‌های قابل چنگ‌زدن و غیرقابل چنگ‌زدن طراحی شده است. میوه‌های قابل چنگ‌زدن میوه‌هایی هستند که مسدود نبوده یا صرفا با برگ درخت مسدود شده‌اند. میوه‌های غیرقابل چنگ‌زدن میوه‌هایی هستند که با شاخه درخت یا میوه دیگر مسدود شده‌اند. این کار به منظور بهبود در استخراج کننده ویژگی شبکه YOLOv5 از ماژول توجه SE و تغییراتی در Bottleneck-CSP استفاده کرده و در مقایسه با شبکه‌های دومرحله‌ای سرعت خیلی زیاد با دقت قابل قبول داشته است [۳۳]. سان و همکارانش شبکه YOLOv5-PRE را برای آشکارسازی میوه سیب به منظور افزایش سرعت پیشنهاد داده‌اند که در این معماری از شبکه‌های ShuffleNet و GhostNet به منظور کاهش اندازه مدل و افزایش سرعت و همچنین از ماژول‌های توجه CA و CBAM برای بهبود دقت شبکه استفاده شده است. شبکه پیشنهادی توانسته بهنمره F1 برابر با ۰,۸۸۹ و زمان استنتاج ۲۷ میلی ثانیه در هر تصویر دست یابد[۳۴].هوانگ و همکارانش از شبکه YOLOv4 و مکانیسم توجه ECA به منظور آشکارسازی و تشخیص عناصر ترافیک جاده‌ای استفاده کردند که در این مطالعه میانگین دقت متوسط نسبت به شبکه اصلیYOLOv4 به میزان ۱۵/۸۰% افزایش یافته و میانگین دقت متوسط ۹۰/۴۵% حاصل شده است. در این مطالعه علاوه بر ماژول توجه ECA ماژول‌های توجه SE و CBAM نیز بررسی شده‌اند که بهترین نتایج با ماژول ECA حاصل شده است[۳۵]. کیم و همکارانش از شبکه YOLOv5 برای بهبود آشکارسازی اهداف کوچک در تصاویر هوایی که در آنها مشکلاتی از قبیل وضوح پایین و شباهت اهداف با پس زمینه وجود دارد، استفاده کردند. در این مطالعه از ماژول توجه ECA برای اصلاح ماژول C3بر روی پایگاه داده VEDAI استفاده شده که در آن میانگین دقت متوسط نسبت به YOLOv5 اصلی به میزان ۶/۹% افزایش یافته است و تعداد پارامترها به میزان ۱۰/۲ مگا کاهش یافته است[۳۶]. بوهونگ و همکارانش از YOLOv3 و ماژول توجه ECA برای آشکارسازی زباله استفاده کردند که در این مطالعه با حفظ سرعت آشکارسازی، میانگین دقت متوسط به میزان ۱/۰۷% افزایش پیدا کرد [۳۷].

سیب بعنوان یک میوه بسیار مفید در سبد غذایی برای آشکارسازی در این مقاله مورد مطالعه قرار گرفته است. از آنجایی که میوه سیب رنگ‌های متفاوتی دارد و عمده آن قرمز، زرد، و سبز در حالت رسیده است، بنابراین در آشکارسازی و برداشت آن بویژه برای سیب سبز که با برگ‌ها همرنگ است چالش وجود دارد. در این مقاله از YOLOv5برای آشکارسازی استفاده شده و معماری آن برای ایجاد مصالحه بهتر بین دقت و سرعت در آشکارسازی میوه سیب بهبود یافته است. از ماژول‌های توجه جدید که در چند سال اخیر در کار آشکارسازی استفاده شده، در تغییر معماری YOLOv5استفاده کرده و نهایتا یک معماری جدید با استفاده از ماژول توجه [۳۸ECA[1] پیشنهاد داده‌ایم. جدول (۱) بطور خلاصه نتایج مطالعات ارائه شده در زمینه آشکارسازی میوه سیب را نشان می‌دهد.

ساختار مقاله بدین ترتیب است. در بخش دوم به توضیح پایگاه‌هاي داده و روش‌های بکار رفته در این مقاله پرداخته شده و در بخش سوم معماری پیشنهادی مطرح می‌شود. در بخش چهارم به آزمایش‌ها و نتایج بکارگیری معماری YOLOv5در آشکارسازی میوه سیب، بررسی و بهبود تعمیم‌پذیری مدل با روش یادگیری انتقالی، بهبود سرعت شبکه با استفاده از ماژول توجه ECA، بررسی شبکه با فایل‌های حاشیه‌نویسی اصلاح شده پایگاه داده Kfuji و بررسی ACF که یک روش سنتی برای آشکارسازی اشیاء است روی پایگاه داده KFujiپرداخته می‌شود. در نهایت جمع‌بندی و نتیجه‌گیری در بخش پنجم بیان می‌شود.

۲- مواد و روش‌ها

در این قسمت پایگاه‌داده‌های مورد استفاده در مقاله و روش‌هایی که برای آشکارسازی استفاده شده، بیان می‌شوند.

۲-۱- پایگاه‌های داده بکاررفته

تصاویر استفاده شده در این مقاله از سه پایگاه داده در دسترس [۳۹KFuji]، [۴۰MinneApple] و [۲۸ACFR] است. پایگاه داده KFuji با Kinect v2 در مزرعه تاراسو، یک مزرعه سیب تجاری واقع در آگرامونت، کاتالونیای اسپانیا جمع‌آوری شده است. هر تصویر این پایگاه داده در فرمت JPG و شامل ۳ کانال رنگیRGB، عمق و مادون قرمز است. تصاویر خام دارای وضوح 1080*1600پیکسل بوده که در شب و زیر نور مصنوعی جمع‌آوری شده است. در این پایگاه داده بدلیل اینکه تعداد سیب‌ها در هر تصویر زیاد است و اندازه سیب‌ها نسبت به تصویر خیلی کوچک است هر تصویر را به ۹ بخش ۳۷۳*۵۴۸ تقسیم کرده‌اند.۹۶۷ تصویر بصورت دستی و با کادرهای محصورکننده مستطیلی برچسب‌گذاری شده و در فرمت‌های Xml و CSV ذخیره شده است.

[1]Efficient channel attention

جدول (۱): جدول نتایج مربوط به چند نمونه از مطالعات انجام شده روی آشکارسازی میوه سیب.

Model	F1	FPS
Faster R-CNN, Vgg16 [28]	90.4	7.7
Yolov3, DenseNet [29]	81.7	3.3
LedNet, LW-backbone [30]	83.4	35.7
Faster R-CNN, Vgg16 [31]	89.8	13.6
Mask R-CNN, NMS branch [32]	90.5	4
Yolov5, Bottleneck-CSP2+SE [33]	87.49	66.67
Yolov5-PRE, ShuffleNet+GhostNet, CBAM+CA [34]	88.9	37

برای استفاده از این مجموعه داده در شبکه YOLOv5 باید فرمت حاشیه‌نویسی‌ها متناسب با فرمت مورد قبول شبکه YOLOv5باشد. تغییر فرمت حاشیه‌نویسی‌نیاز طریق سایتwww.roboflow.comدر سال ۱۴۰۱انجام شده است. فرمت حاشیه‌نویسی شبکهYOLOv5 شامل کلاس شی، مختصات مرکز کادر و عرض و ارتفاع آن است. همچنین ما در این مقاله فایل‌های برچسب‌گذاری پایگاه داده KFuji را با LabelIMG اصلاح کرده‌ایم.پایگاه داده MinneApple یکی دیگر از پایگاه‌های داده مشهور در زمینه آشکارسازی و تقسیم‌بندی میوه سیب است. داده‌های این پایگاه در مرکز تحقیقات باغبانی دانشگاه مینه‌سوتا بین ژوئن ۲۰۱۵ تا سپتامبر ۲۰۱۶ با گوشی سامسونگ گلکسی S4 تحت شرایط روشنایی متفاوت، ظاهر مختلف میوه‌ها روی درختان و انسداد میوه‌ها با شاخ و برگ و سایر میوه‌ها در فرمت PNGجمع‌آوری شده و از تصاویر مربوط به سیب‌های سبز آن در این مطالعه استفاده شده است. از آنجا که تصاویر این پایگاه داده مربوط به کل درخت است و تعداد سیب‌ها در هر تصویر زیاد بوده و اندازه میوه‌ها نسبت به اندازه تصویر بسیار کوچک است، بنابراین هر تصویر به ۹ تصویر فرعی تقسیم شده و حاشیه‌نویسی برای تصاویر فرعی انجام شده است.پایگاه دادهACFR از تصاویر میوه‌های انبه،بادام و سیب تشکیل شده است که از حسگر UGV+PointGrey LadyBug برای جمع‌آوری میوه سیب، حسگر UGV+Prosilica GT3300c برای میوه انبه و دوربین دستیCanon EOS60D برای میوه بادام استفاده شده است. همچنین حاشیه‌نویسی مستطیلی برای میوه‌های انبه و بادام و حاشیه‌نویسی دایروی برای میوه سیب بکار رفته است. برای تصاویر سیب ماسک‌های تصاویر نیز در فرمت PNG آماده شده است. برای استفاده از شبکه YOLOv5 لازم است برچسب‌گذاری‌ها به حالت مستطیلی باشد.بنابراینبرچسب‌گذاری این پایگاه داده نیز اصلاح شده است.جدول (۲) ساختار پایگاه‌های داده مورد مطالعه و تقسیم‌بندی داده‌ها برای مراحل آموزش، اعتبارسنجی و آزمون در شبکه‌های عصبی مورد مطالعه را نشان می‌دهد.

در این جدول عبارت‌هایKF، MAو AC به اختصار بیان‌کننده پایگاه‌های داده KFuji، MinneApple و ACFR هستند.

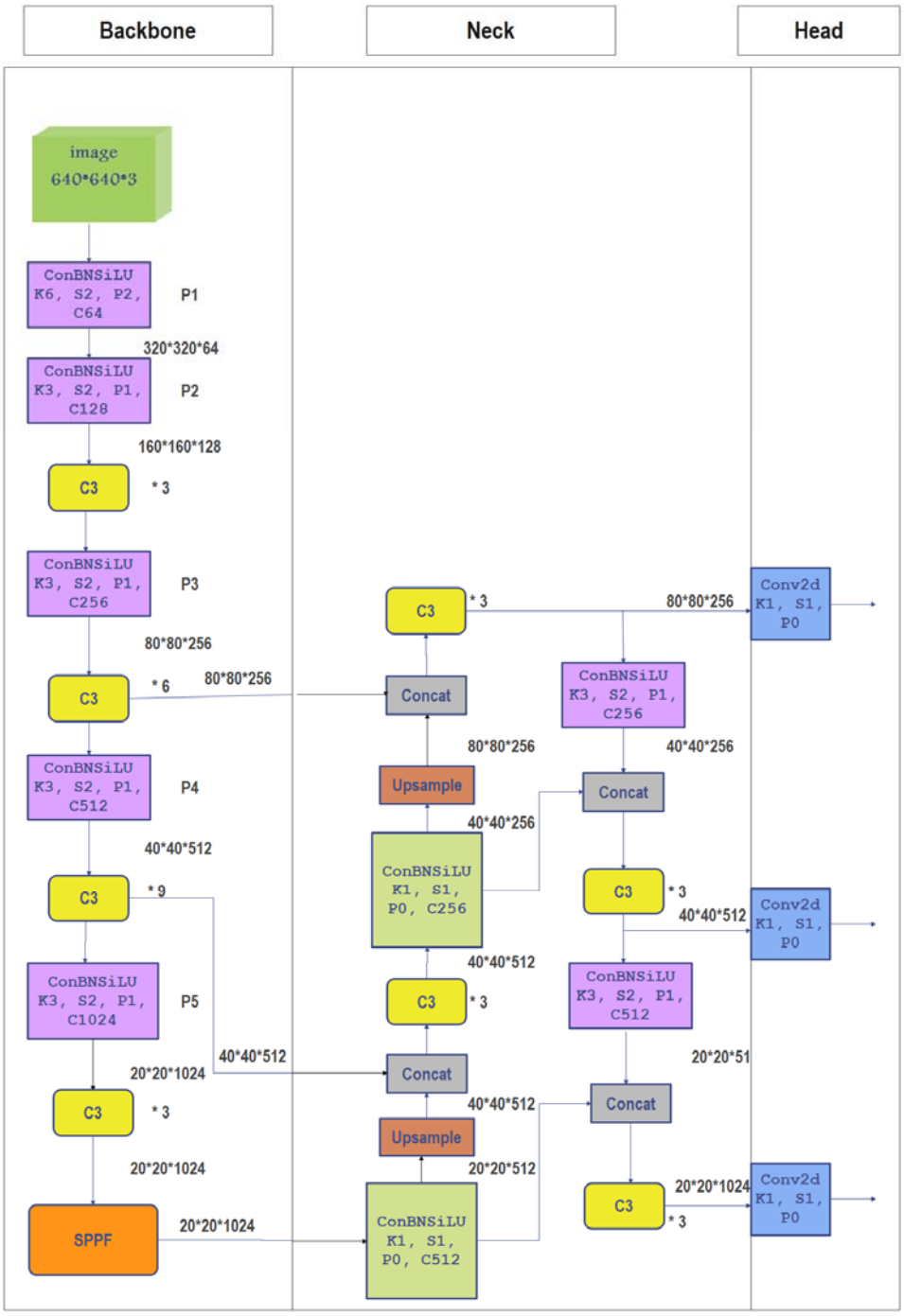
جدول (۲): ساختار پایگاه‌های داده مورد استفاده و نحوه تقسیم‌بندی داده‌ها برای آموزش، اعتبارسنجی و آزمون

Datasets	Raw image	Sub-image (px)	Training (70%)	Validation (20%)	Test (10%)
KF	1920*1080	548*373	677	193	97
MA	720*1280	240*426	450	128	66
AC	1616*1232	202*308	784	224	112

۲-۲- معماری شبکه YOLOv5

YOLOv5نسخه بهبودیافته شبکه YOLO است که ایده ثابت سری YOLO را در طراحی الگوریتم ادامه می‌دهدو شامل پنج نسخه Nano، Small، Medium، Large و Xlarge است که تنها در اندازه معماری یعنی تعداد لایه‌ها و پارامترها تفاوت دارند. در این مقاله از نسخه نانو که کوچکترین معماری از YOLOv5است برای دستیابی به مصالحه دقت و سرعت و تغییر معماری آن با استفاده از مکانیسم توجه استفاده شده است.YOLOv5 شامل چهار بخش ورودی، ستون فقرات، گردن و سر است که معماری آن در شکل (۱) نشان داده شده است. تصویری که باید شناسایی شود از طریق یک لایه ورودی پردازش شده و برای استخراج ویژگی به ستون فقرات ارسال می‌شود. ستون فقرات نقشه‌های ویژگی با اندازه‌های مختلف را بدست می‌آورد و سپس این ویژگی‌ها را از طریق شبکه ادغام ویژگی (گردن) ترکیب می‌کند تا در نهایت سه نقشه ویژگیP3، P4 و P5 با اندازه ۸۰×۸۰ ، ۴۰×۴۰ و ۲۰×۲۰ به ترتیب برای تشخیص اجسام کوچک، متوسط و بزرگ در تصویر تولید کند.بخش سر هم مربوط به آشکارسازی و طبقه‌بندی شیء است. پس از ارسال سه نقشه ویژگی به بخش سر، یک آرایه چندبعدی شامل کلاس شی، احتمال کلاس، مختصات و اطلاعات عرض و ارتفاع جعبه استنتاج می‌شود. سپس برای فیلترکردن اطلاعات بی‌فایده آرایه، یک عملیات پس‌پردازش شامل حد آستانه اطمینان[1] به منظور انتخاب جعبه‌هایی با احتمال بالاتر از حد آستانه و الگوریتم سرکوب غیرحداکثری[2] برای انتخاب یک جعبه با بالاترین احتمال از میان جعبه‌های انتخاب شده، اعمال می‌شود.ستون فقرات شامل چندین ماژول (Conv+BatchNorm+SiLU) ConBNSiLU و ماژول‌هایC3 و در نهایت یک ماژول SPPF[3] است.

[1]Confidence threshold
[2]Non-maximum suppression
[3]Spatial Pyramid Pooling-Fast



شکل (۱): معماری [YOLOv5[۴۳].

ماژول ConBNSiLU برای کمک به ماژول C3 در استخراج ویژگی استفاده می شود، در حالی که ماژول SPPF،یک لایه ادغام هرمی فضایی سریع است که محدودیت ثابت‌بودن اندازه شبکه را حذف می کند، یعنی دیگر لازم نیست ورودی شبکه یک تصویر با اندازه ثابت باشد. به طور خاص، یک لایهSPPF در بالای آخرین لایه کانولوشن اضافه می‌شود. لایهSPPF ویژگی‌ها را ترکیب می‌کند و خروجی‌هاییبا طول ثابت تولید می‌کند که سپس به لایه‌های کاملاً متصل وارد میشوند.در ستون فقرات YOLOv5، مهمترین لایه ماژول C3 است که ایده اصلی آن از [CSPNet[۴۱] آمده است. این لایه توانایی استخراج ویژگی در ستون فقرات را با حذف تکرار اطلاعات گرادیان تضمین می‌کند. بخش گردن روش PAN[1]را بکار می‌برد که یک مسیر ترکیب ویژگی از پایین به بالا است و برای افزایش دقت تشخیص اشیا در مقیاس‌های مختلف استفاده می‌شود.

[1]Path Aggregation Network

۲-۲-۱- نسخه‌های مختلف شبکه YOLO و YOLOv5

اساس کار تمام نسخه‌های YOLO یکسان است. به این صورت که تصاویر به s^2 سلول با ابعاد یکسان تقسیم می‌شود و هر سلول مسئول آشکارسازی اشیایی است که مرکز آنها درون سلول قرار گیرد. تفاوت‌های عمده‌ای که بین نسخه‌های مختلف YOLO وجود دارد در بخش ستون فقرات و گردن است. در YOLOv1 از GoogleNet بعنوان ستون فقرات استفاده شده است. در نسخه دوم شبکه YOLO از Darknet19 بجای GoogleNet بکار رفته و در YOLOv3 از Darknet53 بعنوان ستون فقرات استفاده شده و مشکل آشکارسازی اشیا کوچک در نسخه دوم آن با عمیق‌کردن تعداد لایه‌های شبکه استخراج‌کننده ویژگی از ۱۹ لایه به ۵۳ بهبود یافته است. در نسخه‌های چهارم و پنج نیز از CSPDarknet53 بعنوان ستون فقرات استفاده شده و تفاوت عمده این دو نسخه اخیر در الگوریتم استفاده شده در بخش گردن و محاسبه تلفات است[۴۲]. تنها تفاوت نسخه‌های مختلف YOLOv5 در تعداد لایه‌ها و پارامترها است که هرچه این مقدار بیشتر باشد در زمان و دقت آموزش تاثیرگذار است.

۲-۳- ماژول توجه ECA

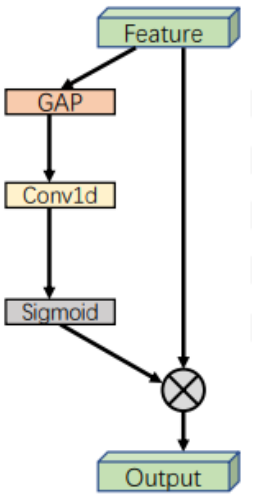
ماژول توجه کانالی ECA که در شکل (۲) نشان داده شده است، ویژگی‌های W*H*C استخراج شده در لایه قبلی را می‌گیرد و از طریق یک GAP[1] تبدیل به یک تانسور 1*1*C می‌کند[۳۵]. به این ترتیب یک میانگین کلی از ویژگی‌ها در هر کانال بدست آمده که توسط یک لایه کانولوشنی یک بعدی با وزن‌های قابل یادگیری فیلتر شده و در ادامه از یک تابع فعالساز سیگموئید استفاده می‌شود تا وزن‌هایی برای کانال‌ها با توجه به اهمیت آنها بدست آید. در نهایت وزن‌های بدست آمده که یک تانسور 1*1*C هستند در ویژگی‌های ورودی ضرب می‌شوند. خروجی، یک تانسور از ویژگی‌ها با ابعاد W*H*C است که کانال‌های آن با توجه به اهمیت آنها وزن‌دهی شده‌اند.

۲-۴- معیارهای ارزیابی آشکارسازی شی

رابطه (۱) مربوط به معیار صحت است که نشان می‌دهد چند پیش‌بینی مثبت انجام شده درست هستند. در این رابطه، TP مثبت حقیقی و FP مثبت کاذب است.

$$P = \frac{TP}{TP + FP} \qquad (۱)$$

رابطه (۲) مربوط به معیار یادآوری است که نشان می‌دهد طبقه‌بندی کننده در بین تمام موارد مثبت در داده‌ها چند مورد از موارد مثبت را به درستی پیش‌بینی کرده است و FN نشان‌دهنده تعداد منفی‌های کاذب است.



شکل (۲): معماری ماژول توجه [ECA۱۸].

$$R = \frac{TP}{TP + FN} \qquad (۲)$$

رابطه (۳) مربوط به معیار F1 است که صحت و یادآوری را با هم ترکیب می‌کند و بطور کلی بعنوان میانگین هارمونیک این دو توصیف می‌شود.

$$F_1 = 2 \times \frac{P \times R}{P + R} \qquad (۳)$$

معیار معروف دیگر در بررسی عملکرد الگوریتم‌های آشکارسازی mAP است که با در نظر گرفتن میانگین دقت متوسط (AP) در حد آستانه‌های مختلف اشتراک بر اجتماع[2] (IoU) محاسبه می شود (رابطه (۴)). در این رابطه Q تعداد حدآستانه‌های بکاررفته و AveP(q) صحت متوسط برای حد آستانه q‌ام است.

$$mAP = \sum_{q=1}^{Q} \frac{AveP(q)}{Q} \qquad (۴)$$

IoU نیز به صورت رابطه (۵)تعریف می‌شود:

$$IoU = \frac{GT \cap Pr}{GT \cup Pr} \qquad (۵)$$

که در آن GTمساحت کادر حقیقت مبنا و Prمساحت کادر پیش‌بینی‌شده توسط شبکه است.علائم ∩ و ∪ نیز به ترتیب مربوط به عملیات اشتراک و اجتماع است.

برای بررسی سرعت الگوریتم آشکارسازی، یک معیار استفاده از FPS است که تعداد فریم‌های استنتاج شده در واحد ثانیه را نشان می‌دهد. البته این معیار کاملا به سخت‌افزاری که مدل آشکارسازی روی آن اجرا می‌شود بستگی داشته و بنابراین گزارش تعداد پارامترهای یک مدل آشکارساز روش دیگری برای مقایسه مدل‌های مختلف آشکارسازی شیء به لحاظ سرعت است.

[1]Global Average Pooling

[2]Intersection over Union

۳- معماری YOLOv5 پیشنهادی

برای بهبود مصالحه دقت و سرعت از ماژول توجه کانالیECA برای تغییر ستون فقرات شبکه استفاده شده است. در معماری پیشنهادی تمامی ماژول‌های C3 در ستون فقرات با ماژول توجه ECA جایگزین شده و از بهینه ساز SGD برای آموزش استفاده شده است.معماریYOLOv5 پیشنهادی در شکل (۳)نشان داده شده است. YOLOv5 نانو با ماژول C3 در فاز ارزیابی دارای ۱۸۵ لایه، ۱۷۶۰۵۱۸ پارامتر و ۴/۱ GFLOPs است. با جایگزینی ماژول توجه ECA این اعداد به ۱۵۰ لایه، ۱۲۷۴۹۳۹ پارامتر و ۲/۹ GFLOPs می‌رسد که نشان‌دهنده کاهش قابل توجه در پیچیدگی مدل است.

۴- آزمایش‌ها و نتایج

در این بخش تمام آزمایش‌های انجام شده در این مقاله، جداول، تصاویر و تحلیل نتایج آنها بیان شده است.

۴-۱- تنظیمات پیاده‌سازی

شبیه‌سازی‌ها روی سیستم عامل Ubuntu 20.04، پردازنده Intel® Xeon(R) silver 4210 CPU @ 2.20GHZ × 20، رم ۱۲۸ گیگا بایت، کارت گرافیک NVIDIA GeForce RTX 3090، پایتون ۳.۸.۱۰، کودای۱۱.۱ و پایتورچ ۱.۹.۰ انجام شده است. پارامترها از قبیل مومنتوم، نرخ یادگیری اولیه، کاهش وزن و سایر پارامترها به پارامترهای اصلی در مدل YOLOv5 نسخه نانو اشاره دارد.پیاده‌سازیYOLOv5 با کمک کدهای موجود در https://github.com/ultralytics/yolov5 انجام شده و از فایلrequirements.txt کتابخانه‌های مورد نیازروی سیستم نصب شده است. ساختار پوشه مربوط به پایگاه داده به این صورت باید باشد: پوشه پایگاه داده شامل سه پوشه با نام‌های پوشه آموزش، پوشه اعتبارسنجی و پوشه آزمون است. در داخل هر کدام از این پوشه‌ها، دو پوشه با نام‌های تصاویر و برچسب‌ها وجود دارد که باید تصاویر و برچسب‌های مربوط داخل آن قرار گرفته شود. سپس برای دادن مسیر این تصاویر و برچسب‌ها باید یک فایل با فرمت Yaml ایجاد شود و مسیر تصاویر و برچسب‌ها در این فایل قرار گیرد و این فایل بعنوان مسیر ورودی به شبکه داده شود.

۴-۲- معیار پیشنهادی برای ارزیابی مصالحه دقت-سرعت

در رابطه (۶)، فرمول پیشنهادی مصالحه دقت و سرعت آورده شده است که در آن $F1^{MAX}$ و FPS^{MAX} به ترتیب مقدار بیشینه F1 و FPS را در آزمایش‌ها روی یک پایگاه داده نشان داده و مبنا برای مقایسه F1 و FPS قرار می‌گیرند و TO نیز عدد مصالحه (Trade-Off) است. در واقع ما در این رابطه یک نرمال‌سازی نسبت به بیشترین مقادیر F1 و FPS در هر آزمایش انجام می‌دهیم.

$$TO = \frac{F1}{F1^{MAX}} \times \frac{FPS}{FPS^{MAX}} \quad (۶)$$

۴-۳- آزمایش اول: مقایسه نسخه‌های مختلف YOLOv5در آشکارسازی میوه سیب

در آزمایش اول ابتدا هر پنج نسخه YOLOv5را با هر سه پایگاه داده آموزش داده و آزمون کرده‌ایم. نتایج حاصل از این آزمایش در سه پایگاه داده مد نظر در جداول (۳) تا (۵) و اعداد مصالحه حاصل از این آزمایش در جدول (۶) نشان داده شده است. با توجه به جدول (۶)، بهترین عدد مصالحه در هر پایگاه داده با استفاده از نسخه نانو YOLOv5برقرار است. البته قابل ذکر است که به غیر از پایگاه داده KF، نتایج F1 با استفاده از YOLOv5نسخه نانو در دو پایگاه داده MA و AC با کاهش اندکی مواجه شده، اما با افزایش سرعت دو برابری مدل آشکارساز نسبت به نسخه‌های بزرگتر در مجموع مصالحه بهتری را برقرار می‌کند. بنابراین ما در ادامه آزمایش‌ها از نسخه نانو YOLOv5استفاده کرده‌ایم.

جدول (۳): نتایج حاصل از ارزیابی پنج نسخه متفاوت از YOLOv5 روی پایگاه داده KFuji.

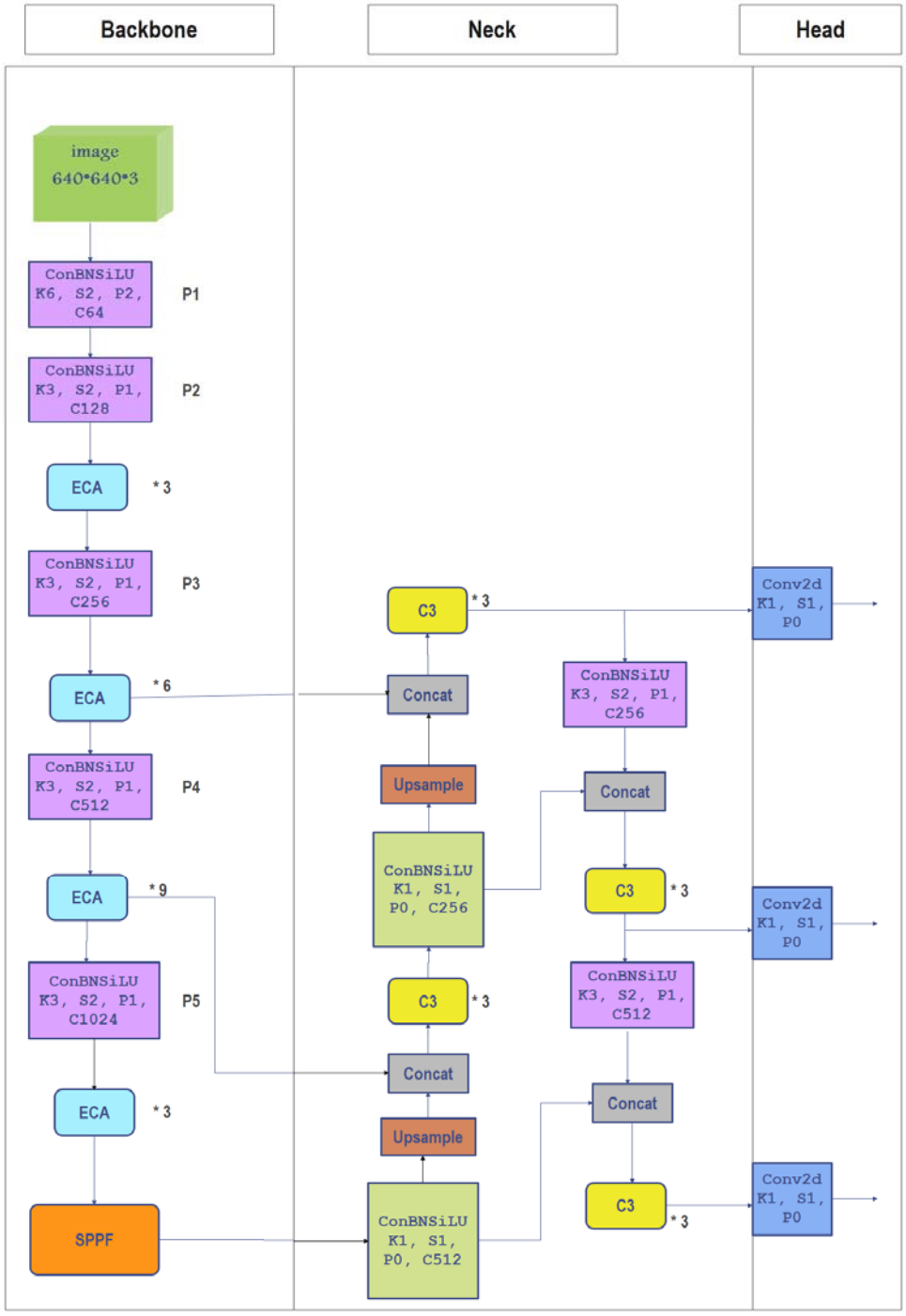
Model	mAP (0.5)	P	R	F1	FPS
YOLOv5n	0.935	0.884	0.86	**0.872**	**137**
YOLOv5s	0.926	0.892	0.841	0.866	122
YOLOv5m	0.925	0.882	0.857	0.869	81
YOLOv5l	0.927	0.885	0.836	0.86	67
YOLOv5x	0.932	0.889	0.842	0.865	49

جدول (۴): نتایج حاصل از ارزیابی پنج نسخه متفاوت از YOLOv5روی پایگاه داده MinneApple.

Model	mAP (0.5)	P	R	F1	FPS
YOLOv5n	0.947	0.92	0.883	0.901	**137**
YOLOv5s	0.942	0.927	0.876	0.901	122
YOLOv5m	0.944	0.928	0.879	**0.903**	81
YOLOv5l	0.942	0.92	0.887	**0.903**	67
YOLOv5x	0.936	0.916	0.886	0.901	49

جدول (۵): نتایج حاصل از ارزیابی پنج نسخه متفاوت از YOLOv5 روی پایگاه داده ACFR.

Model	mAP (0.5)	P	R	F1	FPS
YOLOv5n	0.855	0.814	0.78	0.79	**137**
YOLOv5s	0.854	0.907	0.761	0.828	122
YOLOv5m	0.865	0.956	0.733	0.83	81
YOLOv5l	0.868	0.926	0.754	**0.831**	67
YOLOv5x	0.844	0.936	0.714	0.81	49



شکل (۳): معماری YOLOv5 پیشنهادی با استفاده از ماژول توجه کانالی ECA در بخش ستون‌فقرات شبکه.

۴-۴- آزمایش دوم: بررسی تعمیم‌پذیری آشکارساز میوه سیب

با توجه به اینکه لازم است در آموزش شبکه‌های عصبی عمیق که به روش بانظارت عمل می‌کنند، به تعداد کافی داده همراه با برچسب موجود باشد و از طرفی برچسب زنی داده‌ها کار دشواری است، لذا موردی که مطرح می‌شود بحث تعمیم‌پذیری مدل است. درکار ما تعمیم‌پذیری به این معنی است که آیا مدل آموزش‌دیده بر روی یک پایگاه داده قادر است عملکرد نسبتا خوبی را در آشکارسازی میوه در باغ‌های دیگر با شرایط روشنایی و پس‌زمینه و حتی رنگ سیب متفاوت نشان دهد.

در آزمایش دوم به منظور بررسی تعمیم‌پذیری مدل از یک باغ به باغ دیگر شبکه YOLOv5نسخه نانو را هر بار با یکی از سه پایگاه داده آموزش داده‌ایم و با دو پایگاه داده دیگر آزمون کرده‌ایم؛ که نتایج این آزمایش در جدول (۷) آمده است. با توجه به نتایج این جدول مشخص است که تعمیم‌پذیری مدل به پایگاه‌های داده دیگر پایین است.

جدول (۶): جدول مقایسه اعداد مصالحه دقت-سرعت (TO) برای پنج نسخه YOLOv5 بر روی هر سه پایگاه داده.

Model	KF	MA	AC
YOLOv5n	**1**	**0.998**	**0.951**
YOLOv5s	0.884	0.889	0.887
YOLOv5m	0.589	0.591	0.591
YOLOv5l	0.482	0.489	0.489
YOLOv5x	0.355	0.357	0.349

جدول (۷): نتایج حاصل از ارزیابی YOLOv5نسخه نانو برای بررسی تعمیم‌پذیری

Train Data	Test Data	P	R	F1	Ave. F1
KF	**AC**	0.40	0.22	0.28	0.195
	MA	0.36	0.07	0.11	
AC	**KF**	0.75	0.51	0.61	0.42
	MA	0.53	0.15	0.23	
MA	**KF**	0.69	0.52	0.59	0.57
	AC	0.72	0.45	0.55	

همچنین تصاویر نمونه حاصل از آزمون مدل روی هر سه پایگاه داده زمانی که شبکه بطور جداگانه با هر سه پایگاه داده آموزش می‌بیند در شکل (۴) نشان داده شده است.

نتیجه دیگری که از جدول (۷) می‌توان گرفت این است که در حالتی که پس‌زمینه و شیء تشابه بیشتری دارند (مانند MA)، بدلیل اینکه شبکه ویژگی‌های سطح بالاتری را می‌آموزد قدرت تعمیم‌پذیری مدل بالاتر می‌رود و این استدلال از روی نتایج نیز حاصل می‌شود. برای مثال زمانی که شبکه با پایگاه داده KF که شیء و پس‌زمینه نسبت به دو پایگاه داده دیگر خیلی متفاوت هستند آموزش می‌بیند و روی دو پایگاه داده دیگر آزمون می‌شود نتایج خوبی در مقایسه با حالتی که شبکه با MAیا AC آموزش داده شده و روی دو پایگاه داده دیگر آزمون شود ندارد (۰,۱۹۵ در مقایسه با ۰,۴۲ و ۰,۵۷).

در مقایسه تعمیم‌پذیری میان پایگاه‌های داده MA و AC نیز شبکه آموزش‌دیده با MA بهتر از شبکه آموزش دیده با AC عمل می‌کند; بدلیل اینکه شبکه با MA که تصاویر پس‌زمینه و سیب‌ها هر دو سبز هستند، آموزش دیده و برای تمایز سیب و پس‌زمینه ناگزیر ویژگی‌های سطح بالاتری مانند شکل شیء را علاوه بر رنگ شیء استخراج کرده است. بنابراین در مقایسه با پایگاه داده AC تعمیم‌پذیری بالاتری دارد (۰,۵۷ در مقایسه با ۰,۴۲).

با توجه به نتایج آموزش شبکه با دو پایگاه داده KF و AC و ارزیابی روی پایگاه داده MA نیز می‌توان نتیجه گرفت که با توجه به شرایط روشنایی پیچیده‌تر AC، به تعمیم‌پذیری بهتری در پایگاه داده MAرسیده است (۰,۲۳ در مقابل ۰,۱۱).

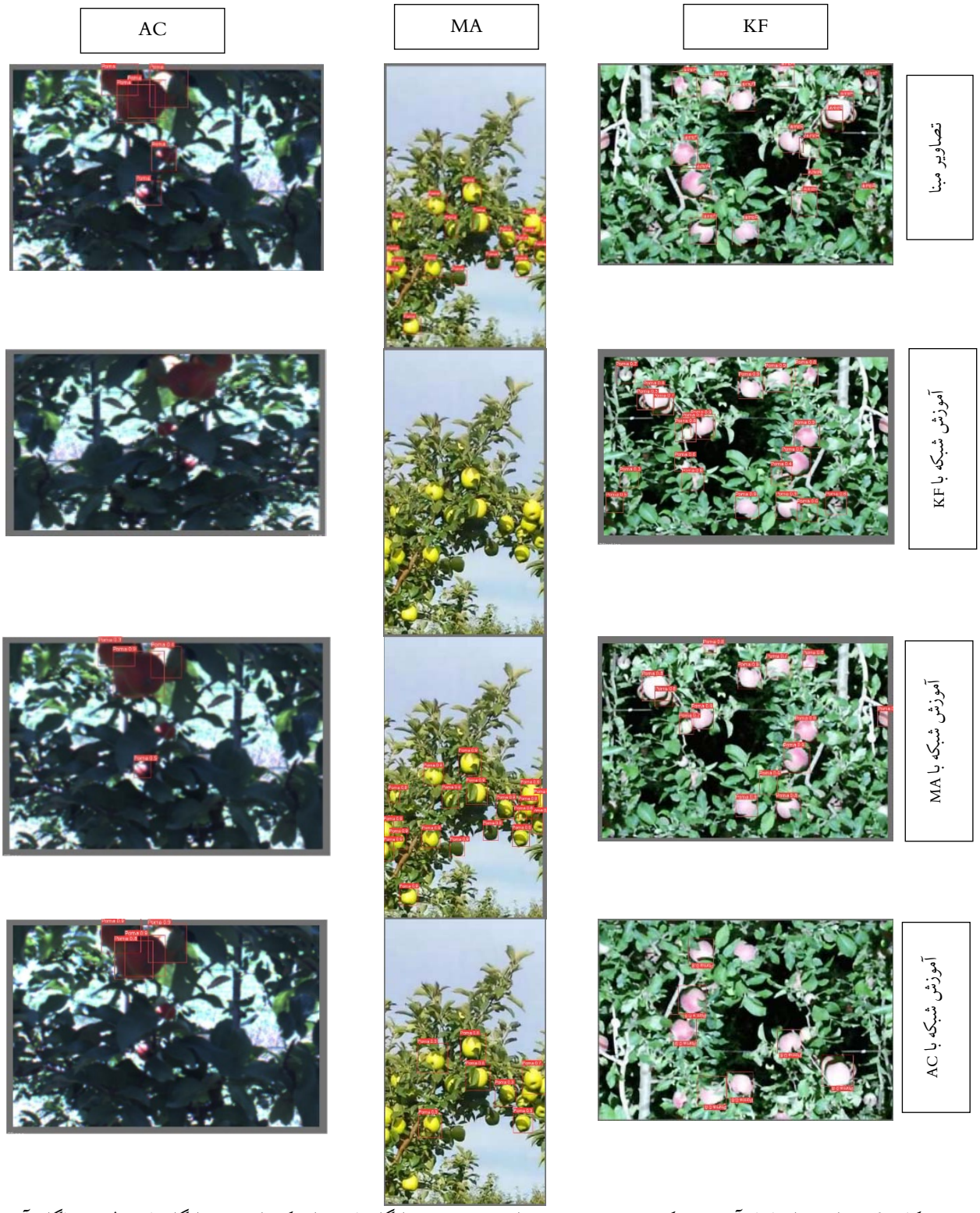
۴-۵- آزمایش سوم: یادگیری انتقالی برای بهبود تعمیم‌پذیری مدل

برای بهبود تعمیم‌پذیری به باغ دیگر می‌توانیم با تعداد بسیار کمی از داده‌های حاشیه‌نویسی شده از تصاویر باغ مقصد شبکه را آموزش داده و به عبارتی از روش یادگیری انتقالی استفاده کنیم. بدین منظور، در ابتدا ستون فقرات شبکه با داده‌های وسیعی که حاشیه‌نویسی آن‌ها موجود است و به آن‌ها تصاویر مبدا می‌گوییم، آموزش داده شده است و وزن‌های مدل فریز شده است. سپس با یادگیری انتقالی لایه‌های آخر شبکه که همان بخش سر شبکه است با تعداد متعادلی داده از تصاویر مبدا و مقصد، یعنی ۵۰ داده از تصاویر مقصد که بعنوان تصاویر انتقالی هستند بعلاوه ۵۰ داده از تصاویر مبدا آموزش داده شده است. هر کدام از این مجموعه داده شامل ۴۰ داده آموزشی و ۱۰ داده اعتبارسنجی است. جدول (۸) نتایج ارزیابی را در دو حالتی که از یادگیری انتقالی استفاده شده (TF) و از یادگیری انتقالی استفاده نشده(Wo-TF) نشان می‌دهد. از روی این نتایج درمی‌یابیم که با یادگیری انتقالی نتایج بهبود قابل توجهی یافته‌اند. سرعت آشکارسازی نیز ۱۳۷ فریم در ثانیه است.

۴-۶- آزمایش چهارم: بررسی معماری YOLOv5 پیشنهادی

در آزمایش چهارم به منظور کاهش پیچیدگی مدل، کاهش محاسبات و حافظه مورد نیاز ماژول توجه ECA که یک ماژول توجه کانالی است جایگزین ماژول C3 صرفا در قسمت ستون فقرات ساختار YOLOv5نسخه نانو شده است (شکل (۳)). نتایج این آزمایش در هر سه پایگاه داده در جدول (۹) تا (۱۳) نمایش داده شده است. در این جداول، معماری پیشنهادی با ECA و معماری اصلی با C3 مشخص شده که به ترتیب به سرعت استنتاج ۱۶۷ و ۱۳۷ فریم در ثانیه روی داده‌هاي آزمون رسیده‌اند و تفاوت چنداني در نتایج آشکارسازي طبق معیار F1 بین دو روش وجود ندارد. در بررسی تمام حالت‌های برجسته شده در جداول (۱۰) و (۱۲)، معیار Pبا شبکه پیشنهادی ۰,۱% و معیار mAP به میزان ۰,۷% بیشتر از شبکه YOLOv5نسخه نانو است. در جدول (۱۱) معیارR شبکه YOLOv5به میزان ۵%بیشتر از معماری پیشنهادی است.

میانگین اعداد مصالحه در جدول (۱۳) در حالتی که دیتاست آموزش (مبدا) و آزمون (مقصد) یکی نباشند، با یادگیری انتقالی گزارش شده است. با توجه به جدول (۱۳) در مجموع در هر سه پایگاه داده میانگین عدد مصالحه در بکارگیری ماژول C3 و ECA (پیشنهادی) به ترتیب برابر ۰,۷۶ و ۰,۹۲ بدست آمده است.



شکل (۴): نتایج حاصل از آزمون شبکه YOLOv5 نسخه نانو روی هر سه پایگاه داده زمانی که با هر سه پایگاه داده بطور جداگانه آموزش می‌بینند.

۴-۷- آزمایش پنجم: بررسی روی دیتاست اصلاحی KF

آزمایش پنجم در این مقاله، بررسی مجدد پایگاه داده KF با برچسب‌گذاری‌های اصلاح شده است.در بیشتر تصاویر این پایگاه داده برخی سیب‌ها برچسب‌گذاری نشده بودند و شبکه آن‌ها را سیب تشخیص داده بود که این باعث افزایش مثبت کاذب می‌شد. اصلاح فایل‌های حاشیه‌گذاری باعث کاهش مثبت کاذب و طبق رابطه (۱) افزایش صحت شد. در این آزمایش نتایج را با پایگاه داده اصلاح شده در جدول (۱۴) گزارش کرده‌ایم.

جدول (۸): نتایج F1 حاصل از ارزیابی شبکه YOLOv5نسخه نانو بر روی هر سه پایگاه داده با /بدون روش یادگیری انتقالی

Train Data	Test Data: KF		AC		MA	
	Wo-TF	TF	Wo-TF	TF	Wo-TF	TF
KF	**0.872**	–	0.283	**0.617**	0.113	**0.799**
AC	0.606	**0.811**	**0.790**	–	0.231	**0.834**
MA	0.592	**0.817**	0.554	**0.679**	**0.901**	–

جدول (۹): نتایج F1 حاصل از ارزیابی معماری پیشنهادی در آزمایش چهارم FPS(C3)=137 و FPS(ECA)=167.

Train Data	Model	Test Data: KF		AC		MA	
		Wo-TF	TF	Wo-TF	TF	Wo-TF	TF
KF	**C3**	0.872	–	0.283	0.617	0.113	0.799
	ECA	**0.880**	–	0.036	**0.626**	0.285	**0.822**
AC	**C3**	0.606	**0.811**	**0.790**	–	0.231	0.834
	ECA	0.655	0.808	0.781	–	0.298	**0.835**
MA	**C3**	0.592	**0.817**	0.554	**0.679**	**0.901**	–
	ECA	0.426	0.806	0.173	0.666	0.886	–

جدول (۱۰): نتایجP حاصل از ارزیابی معماری پیشنهادی در آزمایش چهارم FPS(C3)=137 و FPS(ECA)=167.

Train Data	Model	Test Data: KF		AC		MA	
		Wo-TF	TF	Wo-TF	TF	Wo-TF	TF
KF	**C3**	0.884	–	0.400	**0.804**	0.355	0.862
	ECA	**0.902**	–	0.053	0.670	0.649	**0.914**
AC	**C3**	0.752	**0.838**	0.800	–	0.528	0.888
	ECA	0.712	0.827	**0.894**	–	0.354	0.888
MA	**C3**	0.692	**0.832**	0.722	**0.782**	**0.920**	–
	ECA	0.446	0.831	0.205	0.769	0.916	–

جدول (۱۱): نتایج R حاصل از ارزیابی معماری پیشنهادی در آزمایش چهارم FPS(C3)=137 و FPS(ECA)=167.

Train Data	Model	Test Data: KF		AC		MA	
		Wo-TF	TF	Wo-TF	TF	Wo-TF	TF
KF	**C3**	**0.860**	–	0.219	0.500	0.067	0.744
	ECA	**0.860**	–	0.027	**0.587**	0.183	**0.746**
AC	**C3**	0.508	0.786	**0.780**	–	0.148	0.786
	ECA	0.624	**0.790**	0.693	–	0.257	**0.788**
MA	**C3**	0.517	**0.803**	0.450	**0.600**	**0.883**	–
	ECA	0.408	0.782	0.150	0.589	0.857	–

جدول (۱۲): نتایج mAP(0.5) حاصل از ارزیابی معماری پیشنهادی در آزمایش چهارم FPS(C3)=137 و FPS(ECA)=167.

Train Data	Model	Test Data: KF		AC		MA	
		Wo-TF	TF	Wo-TF	TF	Wo-TF	TF
KF	C3	0.935	–	0.202	0.614	0.214	0.843
	ECA	**0.940**	–	0.004	**0.642**	0.300	**0.863**
AC	C3	0.612	**0.863**	**0.855**	–	0.146	0.875
	ECA	0.665	0.854	0.802	–	0.226	**0.883**
MA	C3	0.535	**0.875**	0.505	0.661	**0.947**	–
	ECA	0.353	0.871	0.041	**0.689**	0.931	–

جدول (۱۳): نتایج مصالحه دقت_سرعت (TO) حاصل از ارزیابی معماری پیشنهادی در آزمایش چهارم.

Train Data	Model	Test Data: KF		AC		MA	
		Wo-TF	TF	Wo-TF	TF	Wo-TF	TF
KF	C3	0.812	–	–	0.641	–	0.728
	ECA	1	–	–	**0.792**	–	**0.912**
AC	C3	–	0.756	0.82	–	–	0.759
	ECA	–	**0.918**	**0.988**	–	–	**0.927**
MA	C3	–	0.762	–	0.705	0.82	–
	ECA	–	**0.916**	–	**0.843**	**0.983**	–

جدول (۱۴): نتایج F1 حاصل از ارزیابی معماری پیشنهادی در آزمایش پنجم قبل و بعد از تصحیح حاشیه‌نویسی پایگاه داده KF.

Model	Test: Before Annotation Correction	After Annotation Correction
C3	0.872	0.903
ECA	0.880	0.910

نتایج F1 زمانی که با پایگاه داده KF آموزش داده و آزمون می‌کنیم، در هر دو معماری شبکه اصلی و شبکه پیشنهادی در حدود ۳% افزایش یافته است.

۴-۸- آزمایش ششم: بررسی آشکارساز شیء مبتنی بر ACF روی دیتاست اصلاحی KF

آشکارساز ACF[1] در جعبه ابزار MATLAB موجود است و شامل یک طبقه‌بندی‌کننده AdaBoost بر اساس درخت‌های تصمیم‌گیری دودویی پشت هم سری شده است. الگوریتم AdaBoost روشی برای طبقه‌بندی دودویی از طریق فرآیند تطبیق بر اساس ویژگی‌های تصاویر آموزشی است.

[1]Aggregated Channel Features

جدول (۱۵): نتایج F1 حاصل از ارزیابی شبکه ACF بعد از تصحیح حاشیه‌نویسی پایگاه داده KF.

Model	P	R	F1
ACF	0.625	0.327	0.430

جدول (۱۵) نتایج حاصل از آشکارساز ACF را روي پایگاه داده KF اصلاح‌شده نشان می‌دهد.برای ایجاد یک آشکارساز مبتنی بر AdaBoost اولین گام جمع‌آوری داده‌های آموزشی بوده که بیان‌کننده دو کلاس شامل تصاویر حاوی شیء و تصاویر فاقد شیء است[۴۴ تا ۴۶]. به تصاویر حاوی شیء نمونه‌های مثبت و تصاویر فاقد شیء نمونه‌های منفی می‌گویند. هدف از این آزمایش مقایسه الگوریتم ACF که یک الگوریتم سنتی است با الگوریتم یادگیری عمیق می‌باشد.

برای آموزش شبکه ACF تصاویر پایگاه داده KF را به قسمت‌های کوچکتر با اندازه‌های مختلف برش دادیم، بطوری‌که این تصاویر برش داده شده شامل یك تا سه سیب و یک سری تصاویر فاقد سیب باشد. همچنین تصاویر برش‌خورده را برچسب زده و در فرمت مناسب شبکه ACF ذخیره کردیم. از نرم افزار متلب نسخه ۲۰۲۱ برای پیاده‌سازی این آزمایش استفاده شده است. دستور trainACFObjectDetector() برای آموزش شبکه بکاررفته که در آن آرگومان‌های ورودی NegativeSampleFactor را که نسبت تعداد نمونه‌های منفی به تعداد نمونه‌های مثبت است برابر ۲، NumStages که تعداد تکرارهای آموزش است را برابر ۱۰۰ وMaxWeakLearners که تعداد حداکثر یادگیرنده‌ها در انتهای هر مرحله است را برابر ۲۰۵۶ قرار دادیم.

با توجه به [۴۵] آشکارساز ACF به علت تلفات اطلاعات مربوط به ویژگی‌های اشیای دارای انسداد ، آنها را با خطای زیادی آشکار می‌کند به همین علت FN افزایش می‌یابد و این امر باعث می‌شود با توجه به رابطه R که در بخش ۲-۴ بیان شد، میزان R افت زیادی داشته باشد. به همین دلیل استفاده از ACF به تنهایی نتایج مطلوبی ندارد و باید درکنار یک شبکه عصبی استفاده شود تا باعث بهبود نتایج شود.

۵- نتیجه‌گیری

مصالحه دقت-سرعت در آشکارسازی میوه سیب در ربات‌های برداشت‌کننده کشاورزی بعنوان موضوع این پژوهش مدنظر قرار گرفته است. شبکه YOLOv5بعنوان یک شبکه تک‌مرحله‌ای بعنوان شبکه اصلی آشکارساز شیء در این مقاله مورد استفاده قرار گرفته است. YOLOv5خود شامل پنج نسخه است که با توجه به آموزش شبکه با هر کدام از این پنج نسخه بصورت مجزا و ارزیابی آن دریافتیم که نسخه نانو بالاترین عدد مصالحه را به ترتیب ۱، ۰,۹۹۸ و ۰,۹۵۱ روی هر سه پایگاه داده سیب مورد ارزیابی (KFuji، MineApple و ACFR) به خود اختصاص می‌دهد. بنابراین در آزمایش‌های بعدی از نسخه نانو استفاده شد.

آزمایش دوم مربوط به بررسی تعمیم‌پذیری شبکه آشکارساز میوه است برای حالتی که شبکه با یک پایگاه داده آموزش دیده و با پایگاه داده دیگری ارزیابی می‌شود. از این آزمایش دریافتیم که هرچه پیچیدگی تصاویر در دیتاست آموزش بیشتر شود، مواردی مانند تشابه رنگ سیب، نتایج ارزیابی مدل آموزش‌دیده شده روی پایگاه داده دیگر نیز بهتر خواهد بود. چون سطح پیچیدگیپایگاه‌های داده به ترتیب در پایگاه دادهMA بیشتر ازAC و AC بیشتر ازKF است، نتایج مدل به ترتیب زمانی که شبکه با پایگاه دادهAC،MA و KF آموزش می‌بینند و روپیایگاه‌های دیگر آزمون می‌شود بهتر است.

در آزمایش سوم بهبود تعمیم‌پذیری مدل با روش یادگیری انتقالی مورد ارزیابی قرار گرفت. در آزمایش چهارم، معماری پیشنهادی بررسی شد. در این معماری با جایگزینی ماژول توجه ECA بجای ماژول C3در بخش ستون‌فقرات شبکه YOLOv5 نسخه نانو تعداد لایه‌ها به میزان ۳۵ و پارامترها به میزان ۴۸۵۵۷۹ کاهش یافت که این جایگزینی با کاهش چشم‌گیر پیچیدگی مدل، تاثیر قابل‌توجهی درF1نداشته و در عین‌حال منجر به افزایش نسبی حدود ۲۰ درصد در سرعت آشکارسازی شده است. با توجه به محدودیت منابع محاسباتی در ربات‌های برداشت، با بکارگیری مدل‌هایی که حجم کمتر و سرعت بالاتر و در عین حال دقت آشکارسازی مطلوبی دارندمی‌توان سخت‌افزارهایی با قیمت پایین‌تر برای بکارگیری در ربات‌ها استفاده کرد.

البته در این مقاله تنها ایده بکارگیری ماژول توجه روی شبکه YOLOv5 که جزو سریع‌ترین معماری‌های حال حاضر است، پیاده‌سازی شده و آشکارسازی روی میوه سیب انجام شده است. اما می‌توان از انواع ماژول‌های توجه کارآمد در هر نوع مدل شبکه آشکارساز و برای هر نوع پایگاه داده دیگر نیز استفاده کرد. از آنجایی که ماژول ECA معماری ساده‌ای داشته، سرعت را افزایش می‌دهد. بنابراین با کاهش تعداد پارامترها می‌توان دقت بکارگیری آن را در معماری‌های دیگر مورد بررسی قرار داد.

مراجع

[1] A. Sharma, A. Jain, P. Gupta, and V. Chowdary, *"Machine learning applications for precision agriculture: A comprehensive review,"* IEEE Access, vol. 9, pp. 4843-4873, 2020.

[2] N. Zhang, M. Wang, and N. Wang, *"Precision agriculture-a worldwide overview,"* Computers and electronics in agriculture, vol. 36, no. 2-3, pp. 113-132, 2002.

[3] R. Gebbers and V. I. Adamchuk, *"Precision agriculture and food security,"* Science, vol. 327, no. 5967, pp. 828-831, 2010.

[4] Y. Tang et al., *"Recognition and localization methods for vision-based fruit picking robots: A review,"* Frontiers in Plant Science, vol. 11, p. 510, 2020.

[۵]ربات+به+برداشت+محصولات+کشاورزی+هم+رسید+|+بازار+بذر+گ+کشـــاورزی-bbk)+iran.com)&cvid=bbd4cb8b0a1f439eb0743fd0494be4c0&aqs=edge..69i57.983j0j1&FORM=ANNTA1&PC=U531

[6]https://www.bing.com/search?pglt=41&q=
مزرعه+هوشمند+چیست؟+مروری+بر+نسل+آینده+مزارع+در+عصر+-+تکنولـــوژی+azimmedia&cvid=601edc9f23c847d0a4f99e99d3242ef3&qs=edge..69i57.1323j0j1&FORM=ANNTA1&PC=U531

[7]https://digiato.com/article/2021/07/18/هوشمندسازی-مزارع-کشور-همراه-اول

[8] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "*Deep learning–Method overview and review of use for fruit detection and yield estimation*," Computers and electronics in agriculture, vol. 162, pp. 219-234, 2019.

[9] A. Kamilaris and F. X. Prenafeta-Boldú, "*Deep learning in agriculture: A survey*," Computers and electronics in agriculture, vol. 147, pp. 70-90, 2018.

[۱۰]فاطمه معادی، ندا فرجی، محمد رضا حسن نژاد بی بالان, "**یک روش کارا برای غربالگری اولیه بیماری گلوکوم بر اساس محاسبه نسبت کاپ به دیسک نوری با استفاده از شبکه‌های عصبی کانولوشنی**"، مجله ماشین بینایی و پردازش تصویر, شماره سوم، ص ۲۷ تا ۴۳، ۱۴۰۰.

[11] O. M. Lawal, "*YOLOMuskmelon: quest for fruit detection speed and accuracy using deep learning*," IEEE Access, vol. 9, pp. 15221-15227, 2021.

[12] J. Huang et al., "*Speed/accuracy trade-offs for modern convolutional object detectors*," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7310-7311.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "*Faster r-cnn: Towards real-time object detection with region proposal networks*," Advances in neural information processing systems, vol. 28, 2015.

[14] J. Dai, Y. Li, K. He, and J. Sun, "*R-fcn: Object detection via region-based fully convolutional networks*," Advances in neural information processing systems, vol. 29, 2016.

[15] W. Liu et al., "*Ssd: Single shot multibox detector*," in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, 2016: Springer, pp. 21-37.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "*You only look once: Unified, real-time object detection*," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[17] J. Redmon and A. Farhadi, "*YOLO9000: better, faster, stronger*," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263-7271.

[18] J. Redmon and A. Farhadi, "*Yolov3: An incremental improvement*," arXiv preprint arXiv:1804.02767, 2018.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "*Yolov4: Optimal speed and accuracy of object detection*," arXiv preprint arXiv:2004.10934, 2020.

[۲۰] سیده فروه موسوی، اعظم اکرمی,"**تشخیص هوشمند تاسل در تصاویر پهپادی با استفاده از یادگیری عمیق برای تعیین تاریخ گل دهی**"، مجله ماشین بینایی و پردازش تصویر, شماره سوم، ص ۴۹ تا ۶۳، ۱۴۰۱.

[21] M.-H. Guo et al., "*Attention mechanisms in computer vision: A survey*," Computational Visual Media, pp. 1-38, 2022.

[22] X. Liu, G. Li, W. Chen, B. Liu, M. Chen, and S. Lu, "*Detection of dense Citrus fruits by combining coordinated attention and cross-scale connection with weighted feature fusion*," Applied Sciences, vol. 12, no. 13, p. 6600, 2022.

[23] L. Fu et al., "*Fast and accurate detection of banana fruits in complex background orchards*," IEEE Access, vol. 8, pp. 196835-196846, 2020.

[24] S. Tu et al., "*Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images*," Precision Agriculture, vol. 21, no. 5, pp. 1072-1091, 2020.

[25] Z. Liu et al., "*Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion*," IEEE Access, vol. 8, pp. 2327-2336, 2019.

[26] H. Zang et al., "*Detection method of wheat spike improved YOLOv5s based on the attention mechanism*," Convolutional neural networks and deep learning for crop improvement and production, vol. 16648714, p. 168, 2023.

[27] M. Cao, H. Fu, J. Zhu, and C. Cai, "*Lightweight tea bud recognition network integrating GhostNet and YOLOv5*," Mathematical biosciences and engineering: MBE, vol. 19, no. 12, pp. 12897-12914, 2022.

[28] S. Bargoti and J. Underwood, "*Deep fruit detection in orchards*," in 2017 IEEE international conference on robotics and automation (ICRA), 2017: IEEE, pp. 3626-3633.

[29] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "*Apple detection during different growth stages in orchards using the improved YOLO-V3 model*," Computers and electronics in agriculture, vol. 157, pp. 417-426, 2019.

[30] H. Kang and C. Chen, "*Fast implementation of real-time fruit detection in apple orchards using deep learning*," Computers and Electronics in Agriculture, vol. 168, p. 105108, 2020.

[31] J. Gené-Mola, V. Vilaplana, J. R. Rosell-Polo, J.-R. Morros, J. Ruiz-Hidalgo, and E. Gregorio, "*Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities*," Computers and Electronics in Agriculture, vol. 162, pp. 689-698, 2019.

[32] P. Chu, Z. Li, K. Lammers, R. Lu, and X. Liu, "*Deep learning-based apple detection using a suppression mask R-CNN*," Pattern Recognition Letters, vol. 147, pp. 206-211, 2021.

[33] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "*A real-time apple targets detection method for picking robot based on*

improved YOLOv5," Remote Sensing, vol. 13, no. 9, p. 1619, 2021.

[34] L. Sun et al., "*Lightweight Apple Detection in Complex Orchards Using YOLOV5-PRE*," Horticulturae, vol. 8, no. 12, p. 1169, 2022.

[35] L. Huang, M. Qiu, A. Xu, Y. Sun, and J. Zhu, "*UAV imagery for automatic multi-element recognition and detection of road traffic elements*," Aerospace, vol. 9, no. 4, p. 198, 2022.

[36] M. Kim, J. Jeong, and S. Kim, "*ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image*," Remote Sensing, vol. 13, no. 23, p. 4851, 2021.

[37] L. Bohong and W. Xinpeng, "*Garbage Detection Algorithm Based on YOLO v3*," in 2022 IEEE International Conference on Electrical Engineering, BigData and Algorithms (EEBDA), 2022: IEEE, pp. 784-788 .

[38] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "*ECA-Net: Efficient channel attention for deep convolutional neural networks*," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11534-11542 .

[39] J. Gené-Mola, V. Vilaplana, J. R. Rosell-Polo, J.-R. Morros, J. Ruiz-Hidalgo, and E. Gregorio, "*KFuji RGB-DS database: Fuji apple multi-modal images for fruit detection with color, depth and range-corrected IR data*," Data in brief, vol. 25, p. 104289, 2019.

[40] N. Häni, P. Roy, and V. Isler, "*Minneapple: a benchmark dataset for apple detection and segmentation*," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 852-858, 2020.

[41] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "*CSPNet: A new backbone that can enhance learning capability of CNN*," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390-391.

[42] U. Nepal and H. Eslamiat, "*Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs*," Sensors, vol. 22, no. 2, p. 464, 2022.

[43] https://github.com/ultralytics/yolov5/issues/6998

[44] J. B. Kim, "*Efficient vehicle detection and distance estimation based on aggregated channel features and inverse perspective mapping from a single camera*," Symmetry, vol. 11, no. 10, p. 1205, 2019.

[45] J. Yuan, P. Barmpoutis, and T. Stathaki, "*Pedestrian detection using integrated aggregate channel features and multitask cascaded convolutional neural-network-based face detectors*," Sensors, vol. 22, no. 9, p. 3568, 2022.

[46] T. Chen, S. Lu, and J. Fan, "*S-CNN: Subcategory-aware convolutional networks for object detection*," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 10, pp. 2522-2528, 2017.

مرضیه محمودی‌فر مدرک کارشناسی خود را در رشته مهندسی برق گرایش مخابرات از دانشگاه بین‌المللی امام خمینی (ره) قزوین دریافت کرد. سپس مدرک کارشناسی ارشد خود را در گرایش مخابرات-سیستم از همان دانشگاه اخذ نمود. حوزه‌های پژوهشی و مورد علاقه ایشان پردازش‌سیگنال، پردازش‌تصویر، شناسایی آماری الگو، بینایی‌ماشین، یادگیری‌ماشین، یادگیری عمیق و هوش مصنوعی است.

ندا فرجی مدرک کارشناسی خود را در رشته مهندسی برق گرایش الکترونیک از دانشگاه علم و صنعت ایران دریافت نمود. سپس در رشته مهندسی برق گرایش الکترونیک دیجیتال و در مقاطع کارشناسی ارشد و دکترا از دانشگاه صنعتی امیر کبیر فارغ‌التحصیل گردید. ایشان در سال ۱۳۹۰ یک دوره فرصت مطالعاتی نه ماهه را در دانشگاه دلفت هلند گذرانده و از سال ۱۳۹۲ تا کنون استادیار گروه مهندسی برق-مخابرات سیستم در دانشگاه بین المللی امام خمینی (ره) هستند. زمینه کاری مورد علاقه ایشان، پردازش سیگنال گفتار، یادگیری‌ماشین، یادگیری عمیق و پردازش‌تصویر است.