

# Detection of Cucumber Fruit on Plant in a Greenhouse Environment Using the YOLOv8 Object Detection Algorithm

A. Soleimanipour<sup>1\*</sup>

1- Department of Biosystem Engineering, Gorgan University of Agricultural Sciences and Natural Resources, Gorgan, Iran

(\*- Corresponding Author Email: [asoleimani@gau.ac.ir](mailto:asoleimani@gau.ac.ir))

<https://doi.org/10.22067/jam.2025.92068.1339>

## Introduction

The increasing demand for automation in agriculture, particularly for repetitive and labor-intensive tasks, has driven the development of robotic harvesting systems. Recent advances in computer vision, deep learning, and the availability of large image datasets have made it possible to create robust object detection models for agricultural applications. Traditional harvesting methods, such as bulk harvesting, often lead to fruit damage and loss owing to non-selective picking. Selective harvesting, particularly with the use of robotic systems, offers a promising alternative by combining the precision of human labor with the efficiency of automation. This study presents a deep learning-based model for detecting cucumber fruits on plants in a real greenhouse environment, which is an essential step towards developing autonomous harvesting robots that selectively pick ripe cucumbers.

## Materials and Methods

A dedicated image dataset was curated in a commercial greenhouse, comprising 300 images of cucumber plants captured under various lighting conditions (morning, noon, and evening), to ensure robustness against real-world variability. Images were manually labeled to identify the cucumber fruits and their pedicels. To enhance the model training and prevent overfitting, data augmentation techniques were applied to the training set. Several architectures of the YOLO (You Only Look Once) object detection algorithm were evaluated, including the nano-scale versions YOLOv5n and YOLOv8n, and the small-scale YOLOv8s, in addition to the RT-DETR model.

The YOLOv8 algorithm is known as one of the state-of-the-art algorithms in computer vision because of its high speed, detection accuracy, and adaptability. The YOLOv8 architecture consists of three main parts: backbone, neck, and head, which are responsible for extracting image features, combining and enriching features, and predicting bounding boxes and object classes, respectively.

These models were trained, and their performances were compared based on the detection accuracy and inference time metrics. Training and evaluation were conducted using a suitable computational platform.

## Results and Discussion

The performances of different YOLO models and RT-DETR were rigorously evaluated. The results demonstrated that the YOLOv8n model achieved the highest detection accuracy of 87.5%, surpassing the performances of the other tested models. Importantly, the YOLOv8n model also exhibited a favorable balance between the accuracy and inference time, making it suitable for real-time applications. The analysis considered the trade-off between the number of parameters and detection speed, highlighting the efficiency of YOLOv8n.

The YOLOv8n model demonstrated superior performance in terms of pedicel detection accuracy compared to YOLOv5n, achieving a fitness score of 91.08% (calculated as a weighted average of mAP@50 and mAP@50-95). While exhibiting strong performance in fruit and pedicel detection (Figure 6), the sensitivity of the model for pedicel detection (88.0%) was comparatively lower than that for fruit detection (96.1%). The highest F1 score (0.89) was observed at a confidence level of 39.5%, indicating the effectiveness of the model in balancing the precision and recall for pedicel detection. Overall, YOLOv8n outperformed the other tested models in identifying the class and location of the fruit pedicel. The superior performance of YOLOv8n can be attributed to its architectural advancements and optimized training processes.

## Conclusion

This study successfully developed a deep learning-based model for accurate and efficient cucumber fruit detection in a greenhouse environment. The YOLOv8n model demonstrated superior performance compared with the other evaluated architectures, achieving a detection accuracy of 87.5% while maintaining a good

processing speed. These findings suggest that the YOLOv8n model has significant potential for integration into autonomous vegetable harvesting robots, contributing to the automation of agricultural processes and increased efficiency in greenhouse operations. Future works should explore further optimization and testing under diverse environmental conditions.

**Keywords:** Cucumber, Deep learning, Greenhouse, Harvesting robots, YOLO

# تشخیص میوه خیار روی بوته در محیط گلخانه با استفاده از الگوریتم تشخیص اشیاء YOLOv8

علی‌رضا سلیمانی پور<sup>۱\*</sup>

تاریخ دریافت: ۱۴۰۳/۱۱/۱۸

تاریخ پذیرش: ۱۴۰۴/۰۱/۲۳

## چکیده

در سال‌های اخیر، توسعه اتوماسیون و کاربرد ربات‌ها برای انجام عملیات تکرارشونده و سنگین کشاورزی سرعت بیشتری پیدا کرده است. این امر ناشی از افزایش قدرت رایانه‌ها در پردازش اطلاعات، توسعه پایگاه داده‌های تصویری کلان، و همچنین توسعه مدل‌ها و الگوریتم‌های کاربردی در هوش مصنوعی است. در این مقاله، یک مدل تشخیص میوه خیار روی بوته در محیط واقعی گلخانه، مبتنی بر یادگیری عمیق و بینایی رایانه‌ای، طراحی و توسعه داده شد. ابتدا یک پایگاه تصویری مشتمل بر ۳۰۰ تصویر با میزان شدت نور مختلف که در اوقات مختلف روز تهیه شده بود ایجاد شد. پس از برچسب‌گذاری تصاویر، عملیات داده‌افزایی روی تصاویر آموزشی انجام شد. سپس، با آموزش الگوریتم تشخیص اشیاء YOLO میوه خیار و ساقه آن در محیط طبیعی گلخانه تشخیص داده شد. عملکرد چندین معماری مختلف الگوریتم YOLO شامل نسخه ۵ در مقیاس نانو (YOLOv5n) و نسخه ۸ در مقیاس نانو و کوچک (YOLOv8n و YOLOv8s) و همچنین مدل RT-DETR بر روی پایگاه تصویر جمع‌آوری شده آموزش و ارزیابی شد. نتایج مدل‌ها براساس پارامترهای مختلف دقت و سرعت تشخیص با یکدیگر مقایسه شد. نتایج نشان داد مدل YOLOv8n، در مقایسه با سایر مدل‌های هم‌ارز از دقت تشخیص بالاتری برخوردار است. صحت تشخیص این مدل در تشخیص میوه خیار و ساقه آن، به‌ترتیب برابر با ۹۶/۱ و ۸۸/۰ درصد بود. این در حالی بود که این مدل به لحاظ تعداد پارامتر و سرعت تشخیص هم عملکرد بسیار خوبی داشت. از این سامانه می‌توان در توسعه روبات برداشت صیفی‌جات خودران استفاده کرد.

**واژه‌های کلیدی:** خیار، ربات برداشت محصول، گلخانه، مدل YOLO، یادگیری عمیق

## مقدمه

تولید و سطح زیر کشت خیار در ایران را در بازه ۱۰ ساله اخیر روند نزولی داشته است. عوامل مختلف اقتصادی و مرتبط با فناوری تولید می‌تواند در این موضوع موثر باشد. به‌عنوان مثال؛ (۱) هزینه‌های بالای نیروی کار باعث کاهش سودآوری و رقابت‌پذیری محصولات شده است، (۲) ظهور قوی کشورهایی با مزیت‌های رقابتی مانند هزینه‌های پایین زمین، دستمزد پایین و در برخی موارد با تکنولوژی کشت و تولید بالاتر، (۳) و دشواری فزاینده در یافتن نیروی کار به دلیل فصلی بودن مشاغل مرتبط، شرایط کاری سخت و جذابیت کم به دلیل یک شغل کم مهارت بودن، از عمده دلایل هستند. به‌طور مشخص، سرعت رشد میوه خیار بالا است و نیاز به برداشت میوه‌ها دو یا سه بار در هفته دارد. این فراوانی بالا یکی از دلایلی است که کار برداشت خیار را نسبت به هر میوه سبزی دیگر فشرده‌تر می‌کند. این موارد نیاز به طراحی راهبردهای موثری برای تقویت مزیت‌های تولید این محصول کشاورزی را برجسته می‌کند که در نتیجه بهبود کارایی فرآیندهای درگیر در عملیات تولید آن حاصل می‌شود. ربات‌های خدماتی در حال تبدیل شدن به بخش کلیدی بسیاری از بخش‌های جامعه، از جمله کشاورزی دقیق هستند، جایی که از آن‌ها خواسته می‌شود نقش مهمی در بهبود رقابت و تولید پایدار ایفا کنند

خیار<sup>۲</sup>، یک محصول کشاورزی از خانواده cucurbitaceae، حاوی فیبر، پتاسیم، منگنز، منیزیم و ویتامین A، C، K و نیز شامل گروهی از ویتامین‌های گروه B مانند تیامین، ریبوفلاوین و B-6 هستند که به متابولیسم و تولید انرژی کمک می‌کنند (Mallick, 2022). در سال ۲۰۲۱، تولید جهانی خیار تازه‌خوری<sup>۳</sup> و خیارشور یا سبزیجات را از نظر حجم کل تولید بعد از گوجه‌فرنگی و پیاز در رتبه سوم قرار داد. این آمار نشان‌دهنده اهمیت جهانی خیار و خیارشور از نظر کشاورزی و اقتصادی است. ایران پس از چین و ترکیه در رتبه سوم تولید خیار در جهان قرار دارد (FAO, 2023). با این حال مقدار

۱- گروه مهندسی مکانیک بیوسیستم، دانشکده آب و خاک، دانشگاه علوم کشاورزی و منابع طبیعی گرگان، گرگان، ایران  
(\*) نویسنده مسئول: Email: asoleimani@gau.ac.ir

<https://doi.org/10.22067/jam.2025.92068.1339>

2- Cucumis sativus

3- Cucumber

4- Gherkins

(Fernandez et al., 2018).

دو نوع رویکرد برداشت توسط متخصصان کشاورزی برای کاهش هزینه‌های کارگری باغات اجرا شده است: برداشت فله و برداشت انتخابی. روش برداشت فله بر اساس اصل اعمال ارتعاش در درخت میوه برای جداسازی اجباری میوه است (Mehta, MacKunis, & Burks, 2016). این روش توسط پرورش‌دهندگان میوه‌های مختلف مانند سیب، پرتقال و گیلاس پذیرفته شده است (Zhang et al., 2020). اگرچه سیستم‌های برداشت انبوه با راندمان بالایی همراه هستند (Sola-Guirado, Castro-Garcia, Blanco-Roldán, Gil-Ribes, & González-Sánchez, 2020)، معایب قابل توجهی نیز وجود دارد. آسیب بیش از حد به میوه‌ها (Moseley, House, & Roka, 2012) و تلفات ناشی از برداشت میوه‌های نارس و بیش از حد بالغ (Zhou, Wang, Au, Kang, & Chen, 2022) از آن جمله‌اند. برداشت انتخابی یک روش برداشت برای سامانه‌های رباتیک است که از بازوهای مکانیکی با عملکردهای نهایی برای گرفتن میوه استفاده می‌کنند. این گروه ربات‌ها معمولاً با نصب یک عملکرد نهایی روی یک پلتفرم متحرک و تجهیز آن به بینایی ماشین برای چیدن انتخابی میوه‌های رسیده پیاده‌سازی می‌شوند (Bac, Van Henten, Hemming, & Edan, 2014).

تلاش‌های تحقیقاتی گسترده‌ای از گذشته تاکنون در رابطه با سامانه‌های رباتیک برداشت محصولات کشاورزی صورت گرفته است. قبل از ظهور فناوری هوش مصنوعی به شکل امروزی، از روش‌های سنتی پردازش تصویر و یادگیری ماشین استفاده می‌شد (Xiao, et al., 2023). هرچند با گسترش مفهوم و کاربرد یادگیری عمیق، عمده تحقیقات در زمینه ربات‌های برداشت بر استفاده از این الگوریتم‌ها متمرکز شده است (Jin & Han, 2024). برداشت رباتیک با استفاده از یادگیری عمیق، در مقایسه با روش‌های سنتی پردازش تصویر و یادگیری ماشین، مزایای متعددی دارد: این مزایا عبارتند از: افزایش دقت در شناسایی و برداشت محصولات، قابلیت انطباق با انواع مختلف محصولات زراعی، توانایی تصمیم‌گیری فوری و بلادرنگ در حین برداشت، مدیریت و پردازش صحنه‌های پیچیده مزرعه، یادگیری مستقیم از داده‌های خام تا تصمیم نهایی، کاهش نیاز به دخالت انسانی و در نهایت کاهش هزینه‌های عملیاتی در بلندمدت (Alaudeen, Selvarajan, Manoharan, & Jhaveri, 2024; Salim & Mohammed, 2024). به‌طور کلی، سامانه‌های رباتیک به‌طور بالقوه می‌توانند کارایی ماشین‌ها را با گزینش‌پذیری انسان‌ها به‌صورت طولانی‌مدت ترکیب کنند (Shewfelt & Prussia, 2022). با توسعه سریع و مداوم فناوری‌های هوش مصنوعی و رباتیک، اعتقاد بر این است که رویکرد برداشت رباتیک می‌تواند جایگزین کارگران برداشت محصول شود (Bharad & Khanpara, 2024; Droukas et al., 2023; Kootstra, Wang, Blok, Hemming, & Van

(Henten, 2021; Park et al., 2023).

الگوریتم‌های تشخیص شیء مبتنی بر یادگیری عمیق به دو دسته تک‌مرحله‌ای (مانند YOLO، SSD، RetinaNet) و دو مرحله‌ای (مانند Faster R-CNN و FPN) تقسیم می‌شوند. مدل‌های تک‌مرحله‌ای به سرعت بالاتر در تحلیل تصویر معروف هستند و در کاربردهای بلادرنگ موثرترند، در مقابل مدل‌های دو مرحله‌ای معمولاً از دقت تشخیص بالاتری برخوردارند (Tajane, KhutaleKore, Khairmode, & Girawale, 2024). نسخه‌های اخیر مدل تک‌مرحله‌ای YOLO پیشرفت خوبی در دقت و سرعت تشخیص نسبت به نسخه‌های پیشین داشته است (Jocher, Chaurasia, & Qiu, 2023; Mahani & Karami, 2023). با وجود قابلیت بالای مدل YOLO، کاربرد این مدل در تشخیص میوه نسبتاً کم است. در تحقیقی، محققان از مدل YOLO-V3 بهبودیافته برای تشخیص گوجه‌فرنگی استفاده کردند. آن‌ها در الگوریتم مورد استفاده ابتدا ساختار شبکه در بخش لایه‌های متراکم<sup>۱</sup> مربوط به مدل YOLO-V3 را بهبود دادند و در مرحله دوم الگوریتم حذف غیر حداکثرها<sup>۲</sup> را بهبود دادند. در نتیجه مدل حاصل در پیش‌بینی مکان گوجه‌فرنگی در تصاویر عملکرد بهتری نسبت به الگوریتم Yolo-V3 داشت (Liu, Nouaze, Touko Mbouembe, & Kim, 2020). در تحقیق دیگری، کوزنتسوا و همکاران (Kuznetsova, Maleva, & Soloviev, 2020) با استفاده از برخی تکنیک‌های پیش‌پردازش و پس‌پردازش نتایج خوبی از لحاظ مدت زمان تشخیص برای استفاده از الگوریتم YOLO-V3 در تشخیص سیب به‌دست آوردند. سرعت تشخیص بالا راحتی قابل توجهی را برای برداشت ربات فراهم می‌کرد. نتایج پیاده‌سازی در محیط واقعی نیز نشان داد که استفاده از این الگوریتم، میانگین زمان تشخیص را کاهش می‌دهد و نرخ خطای پایین را حفظ می‌کند. با معرفی مدل YOLO-V4، دقت در تشخیص اشیایی که قسمتی از آن‌ها در تصویر مخفی است بهبود پیدا کرد (Bochkovskiy, Wang, & Liao, 2020). در پژوهشی، وو و همکاران (Wu, Lv, Jiang, & Song, 2020) از YOLO-V4 برای تشخیص شکوفه‌های سیب استفاده کردند. این مدل تشخیص سیب را ساده‌تر کرد و کانال‌ها را برای اطمینان از سرعت و کارایی تشخیص کوتاه کرد. این الگوریتم در مقایسه با الگوریتم‌های Faster R-CNN، YOLO-V2، YOLO-V3 و سایر الگوریتم‌ها، عملکرد بهتری داشت و توانست به تشخیص لحظه‌ای و دقیق گل‌های سیب دست یابد.

اخیراً، از الگوریتم بهبودیافته YOLOv8 در زمینه تشخیص میوه گوجه‌فرنگی استفاده شده است (Yue et al., 2023). در این پژوهش،

1- Dense layers

2- Non-maximum suppression

Galaxy A12 گرفته شدند. ابعاد تصاویر گرفته شده  $3000 \times 3000$  پیکسل بود. البته در مرحله پیش پردازش برای کاهش حجم داده، ابعاد تصاویر کاهش داده شد. از آنجایی که مدل تشخیص اشیاء YOLO در فرآیند تحلیل تصویر، تصاویر را به صورت جزئی به مربع های  $32 \times 32$  بررسی می کند، بنابراین بهتر است ابعاد تصاویر ضریبی از عدد ۳۲ باشد. با توجه به جزئیات موجود در تصاویر خیار در محیط واقعی گلخانه، ابعاد تصاویر پایگاه داده با بالاترین مقدار ممکن برای مدل YOLO نسخه P5 برابر  $640 \times 640$  است، بنابراین ابعاد تصویر در هنگام آموزش مدل ها به  $640 \times 640$  کاهش داده شد.

به طور کلی پایگاه تصویر مورد استفاده در این تحقیق شامل ۳۰۰ تصویر از بوته خیار درختی بود. در مجموع ۱۲۰۹ برچسب روی تصاویر زده شد که ۷۳۳ مورد میوه خیار با وضوح خوب در تصور و ۴۷۶ مورد ساقه متصل به خیار بود. علت برابر نبودن تعداد میوه و ساقه متصل به آن در تصاویر، پوشانده شدن ساقه برخی میوه های موجود با برگ، میوه و یا ساقه اصلی است. در برخی موارد نیز فقط بخشی از میوه در تصویر وجود داشت و قسمت ساقه آن نمایان نبود. در شکل ۱ نمونه ای از تصاویر مورد استفاده برای آموزش مدل های مورد نظر آورده شده است. از این شکل کامل مشخص است که با اعمال تغییرات مختلف در مرحله داده افزایی (جدول ۱)، پیچیدگی در تصاویر به مقدار زیادی افزایش یافته است. بدیهی است این موضوع منجر به قوی تر شدن مدل های آموزش یافته خواهد شد. با انجام عملیات داده افزایی، تعداد تصاویر مورد استفاده در مرحله آموزش مدل به ۳ برابر افزایش داده شد.

### الگوریتم YOLOv8

الگوریتم YOLOv8، یکی از قوی ترین نسخه های مدل تشخیص اشیاء You Only Look Once است که در سال ۲۰۲۳ معرفی شد. این الگوریتم در سرعت تحلیل تصویر، دقت تشخیص و تطبیق پذیری جزو الگوریتم های پیشرو در زمینه بینایی رایانه است و این قابلیت ها آن را به انتخابی جذاب برای کاربردهای مختلف تبدیل می کند. برای درک بهتر قابلیت های این الگوریتم، آشنایی با پیچیدگی های معماری آن ضروری است. معماری الگوریتم YOLO در شکل ۲ نشان داده شده است. نسخه های مختلف این الگوریتم دارای سه بخش اصلی شامل ستون فقرات<sup>۲</sup>، گردن<sup>۳</sup> و سر<sup>۴</sup> است.

یک شبکه بهبود یافته از YOLOv8s-Seg برای تشخیص برخط و مؤثر میوه گوجه فرنگی، رنگ سطح و ویژگی های سطحی معرفی شد. با انجام اصلاحاتی از قبیل جایگزینی ماژول  $C2f$  با ماژول RepBlock (شامل RepConv انباشته)، این الگوریتم بهبود قابل توجهی در قابلیت ترکیب ویژگی ها از خود نشان داد. علاوه بر این، ادغام کانولوشن SimConv، استفاده از تابع ReLU به جای تابع SiLU به عنوان تابع فعال سازی در شبکه ترکیب ویژگی<sup>۱</sup>، و جایگزینی لایه های کانولوشن معمولی باقی مانده با SimConv به بهبود عملکرد الگوریتم کمک می کرد. در تحقیق دیگری، فاتحی و همکاران (Fatehi, Bagherpour, & Amiri Parian, 2025) از الگوریتم YOLOv8s برای شناسایی گل های محمدی شکفته در مزرعه روباز استفاده کردند. نتایج نشان داد مدل YOLOv8s در مقایسه با نسخه های قدیمی تر، شامل YOLOv5s و YOLOv6s، از عملکرد بهتری به لحاظ دقت و سرعت تشخیص برخوردار بود.

یافته های این مطالعات بر پتانسیل الگوریتم های مبتنی بر مدل YOLO و به طور ویژه مدل YOLOv8 در پیشبرد روش های فناورانه در تشخیص میوه ها در محیط واقعی تاکید می کند و راه حلی امیدوارکننده برای کاربردهای بلادرنک و استخراج کارآمد ویژگی های سطحی ارائه می دهد. این موضوع دورنمای ارزشمندی را در این زمینه تحقیقاتی در رابطه با کاربرد مورد بینایی رایانه و پردازش تصویر کشاورزی نشان می دهد. تکامل مداوم این الگوریتم ها با هدف پرداختن به چالش ها در تشخیص و تجزیه و تحلیل خودکار میوه ها در محیط واقعی یک موضوع تحقیقاتی مهم و به روز به شمار می رود. بنابراین، در این تحقیق عملکرد مدل های مختلف یادگیری عمیق مبتنی بر مدل YOLOv8 در تشخیص میوه خیار در محیط واقعی گلخانه و تحت شرایط نورپردازی مختلف مورد بررسی قرار گرفته است.

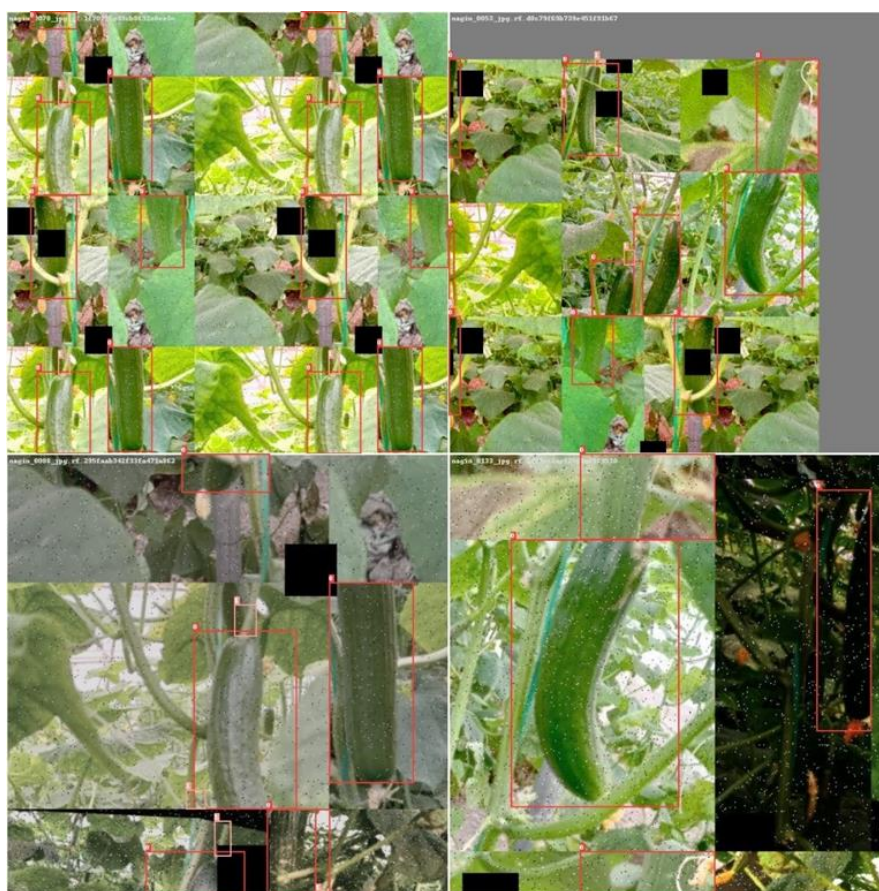
### مواد و روش ها

به منظور ایجاد یک پایگاه تصویر از میوه خیار در شرایط واقعی محیط گلخانه، تصاویری از بوته های خیار درختی در داخل گلخانه گرفته شد. تصاویر از دو گلخانه مختلف تهیه شد و رقم خیار کشت شده در این گلخانه ها رقم ناگین بود. این دو گلخانه واقع در شهرستان پاکدشت، استان تهران بودند. تصاویر در طول روز و در سه وعده صبح، ظهر و عصر تهیه شدند و تصویرگیری تحت نورپردازی طبیعی روز بود و هیچ نورپردازی دیگری انجام نشد. تصاویر از فاصله ۷۰ تا ۱۰۰ سانتی متری دوربین از بوته ها گرفته شدند. قابل ذکر است، تصاویر با استفاده از یک دوربین تلفن همراه مدل Samsung

2- Backbone  
3- Neck  
4- Head

1- Feature Fusion network





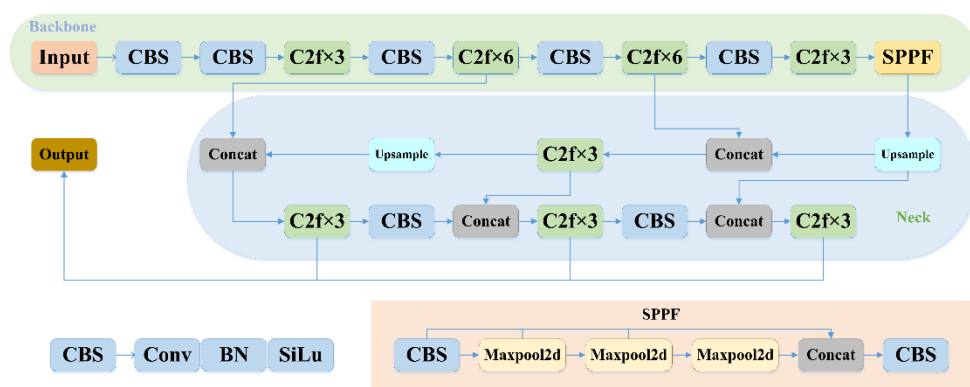
شکل ۱- نمونه از تصاویر مورد استفاده برای آموزش مدل های مورد نظر

Fig. 1. Sample of images used to train the models

جدول ۱- نوع عملیات داده افزایی و مشخصات هر یک برای تقویت پایگاه تصاویر

Table 1- Type of operations and specifications performed for image dataset augmentation

| نوع عملیات داده افزایی | مشخصات                |
|------------------------|-----------------------|
| Type of augmentation   | Specification         |
| برگردان                | Horizontal, Vertical  |
| Flip                   |                       |
| چرخش                   | Between -15° and +15° |
| Rotation               |                       |
| رنگ                    | Between -15° and +15° |
| Hue                    |                       |
| اشباع                  | Between -25% and +25% |
| Saturation             |                       |
| روشنایی                | Between -15% and +15% |
| Brightness             |                       |
| درخشندگی               | Between -10% and +10% |
| Exposure               |                       |
| نوفه                   | Up to 0.1% of pixels  |
| Noise                  |                       |



شکل ۲- معماری الگوریتم YOLOv8

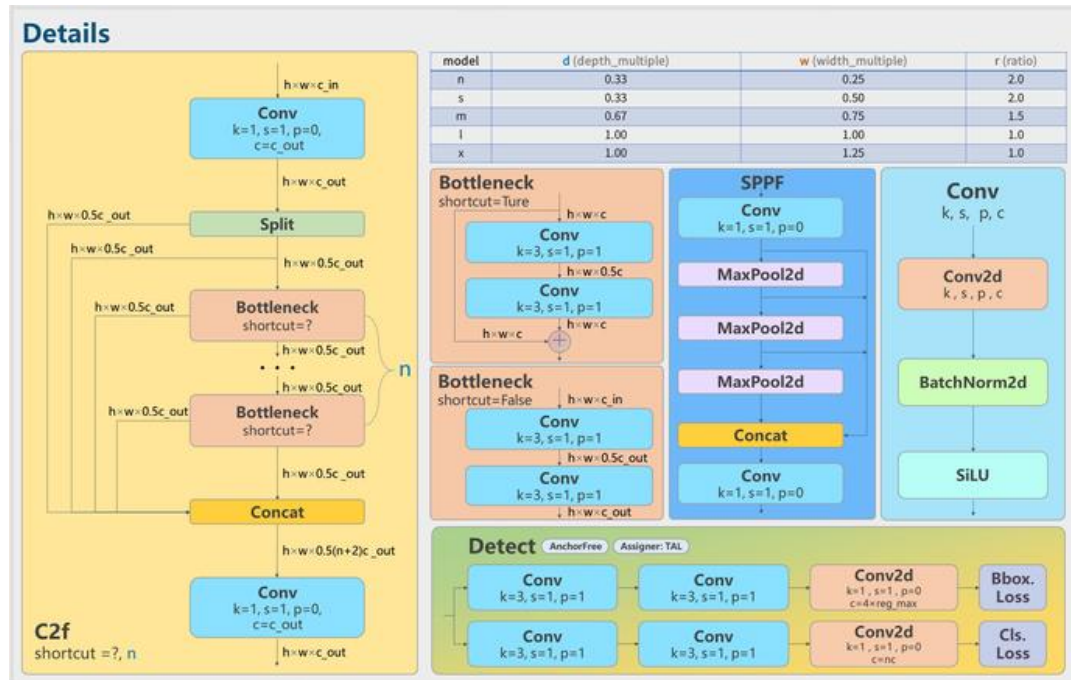
Fig. 2. YOLOv8 architecture

برای ترکیب موثر ویژگی‌های سطح بالا و سطح پایین استفاده می‌کند، فضای ویژگی را غنی می‌کند و به محلی‌سازی دقیق اشیاء کمک می‌کند. جزئیات ماژول‌های استفاده‌شده در الگوریتم YOLOv8 در شکل ۳ نشان داده است.

**سر:** ویژگی‌های پردازش‌شده را می‌گیرد و آن‌ها را به پیش‌بینی تبدیل می‌کند. این بخش شامل چندین ماژول تشخیص است که هرکدام مسئول شناسایی اشیاء با اندازه‌ها و مقیاس‌های مختلف هستند. این ماژول‌ها از جعبه‌های لنگر برای پیش‌بینی جعبه‌های محدودکننده و نمرات اطمینان برای اشیاء احتمالی استفاده می‌کنند. خروجی نهایی لیست تصفیه‌شده‌ای از اشیاء شناسایی‌شده با جعبه‌های محدودکننده و احتمالات مربوطه است.

**ستون فقرات:** به عنوان بلوک پایه، مسئول استخراج ویژگی‌های معنی‌دار از تصاویر ورودی است. مدل YOLOv8 از ماژول CSPDarknet53، که نسخه‌ای ارتقا یافته از معماری Darknet است، استفاده می‌کند که از اتصالات باقی‌مانده و اتصالات جزئی بین مرحله‌ای استفاده می‌کند. این اتصالات جریان کارآمد اطلاعات را در سراسر شبکه تضمین می‌کنند و به بخش ستون فقرات اجازه می‌دهد تا نشانه‌های بصری متنوعی مانند اشکال، بافت‌ها و لبه‌ها را استخراج کند.

**گردن:** ویژگی‌های استخراج‌شده در مقیاس‌های مختلف وجود دارند. گردن به عنوان پلی عمل می‌کند که این ویژگی‌های چند مقیاس را در یک نمایش واحد ترکیب می‌کند. الگوریتم YOLOv8 برای این کار از یک ماژول C2f جدید استفاده می‌کند. این ماژول از مکانیسم‌های ادغام ویژگی و توجه فضایی



شکل ۳- جزئیات ماژول‌های موجود در الگوریتم YOLOv8

Fig. 3. Details of modules in YOLOv8 algorithm

پیشین می‌توان به ماژول  $++PANet$ <sup>۱</sup> (موثر در ترکیب اطلاعات از سطوح مختلف ویژگی‌های تصویر)، سر تشخیص جداسده<sup>۲</sup> (موثر در پیش‌بینی دقیق‌تر با جداسازی پیش‌بینی مختصات جعبه محصورکننده شیء<sup>۳</sup>، احتمال شیء‌بودن و احتمالات کلاس) و استفاده از مکانیسم‌های توجه فضایی و کانالی برای تمرکز بر مناطق مرتبط و حذف نویز که کیفیت تشخیص را بهبود می‌بخشد، اشاره کرد (Jocher et al., 2023). امکان انجام بخش‌بندی نمونه<sup>۴</sup> و تشخیص حالت<sup>۵</sup> با مدل YOLO از نسخه ۸ به آن اضافه شده است.

### آموزش مدل و ارزیابی عملکرد

پایگاه تصویر جمع‌آوری شده به دو بخش تصاویر آموزش و آزمون به نسبت ۲ به ۱ تقسیم شد. به این ترتیب که برای آموزش هر مدل ۲۰۰ عکس و ۱۰۰ عکس باقی‌مانده برای ارزیابی مدل‌ها استفاده شد. البته تصاویر آموزش با استفاده از تکنیک‌های مختلف داده‌افزایی، تصاویر به سه برابر افزایش داده شدند. مدل پیشنهادی با استفاده از فریمورک Pytorch پیاده‌سازی شد. آموزش مدل پیشنهادی و مدل YOLOv8 با مقیاس‌های مختلف بر روی تصاویر آموزش از یک

در نسخه ۸ مدل تشخیص شیء YOLO تغییرات زیادی نسبت به نسخه‌های پیشین ایجاد شده است. ماژول  $SimConv$  (جایگزین ماژول  $Conv$  در نسخه‌های قبل) در YOLOv8 مانند یک ابزار ساده و سریع برای پردازش ابتدایی تصاویر است. این ماژول از سه بخش اصلی تشکیل شده: یک لایه کانولوشن استاندارد برای استخراج ویژگی‌های اولیه، نرمال‌سازی دسته‌ای برای بهبود پایداری آموزش و تابع فعال‌سازی  $SiLU$  برای ایجاد غیرخطی بودن. برخلاف ماژول‌های پیچیده‌تری مانند  $C2f$ ، ماژول  $SimConv$  به دنبال انجام وظایف اساسی با کمترین پیچیدگی است. این ماژول معمولاً در لایه‌های ابتدایی شبکه استفاده می‌شود تا ابعاد تصاویر را کاهش دهد یا تعداد کانال‌های ویژگی را تنظیم کند. این کار باعث می‌شود که پردازش‌های بعدی توسط ماژول‌های پیچیده‌تر مانند  $C2f$  سریع‌تر و کارآمدتر انجام شود. ماژول  $SimConv$  با تمرکز بر سرعت و کارایی، به YOLOv8 کمک می‌کند تا تشخیص اشیاء را به صورت بلادرنگ و با حفظ ساختار ساده و قابل تغییر انجام دهد. همچنین، ماژول  $C2f$  از چندین لایه کوچک و موازی برای استخراج جزئیات بیشتر از تصاویر استفاده می‌کند. این کار باعث می‌شود مدل بتواند اشیاء را با دقت بالاتری تشخیص دهد، در حالی که حجم محاسبات آن زیاد نمی‌شود. به همین دلیل، مدل سریع و سبک باقی می‌ماند و برای استفاده در برنامه‌هایی که نیاز به پردازش سریع دارند مناسب است. از دیگر تغییرات انجام‌شده در مدل YOLOv8 نسبت به نسخه‌های

- 1- Enhanced path aggregation network
- 2- Decoupled detection head
- 3- Bounding box
- 4- Instance segmentation
- 5- Pose detection



برابر با ۴ انتخاب شد و تعداد تکرار آموزش مدل‌ها برابر با ۲۰۰ قرار داده شد. سایر هایپرپارامترهای مدل‌ها همان مقادیر پیش‌فرض آن‌ها در نظر گرفته شدند. در جدول ۲ مقادیر هایپرپارامترهای مورد استفاده در آموزش مدل‌ها آورده شده است.

رایانه پردازش سریع با مشخصات کارت گرافیک NVIDIA® GeForce RTX™ 4080 با رم ویدیویی ۱۶ VRAM گیگابایت، پردازنده Intel® Core™ i5-13600KF (دارای ۱۴ هسته و ۲۰ رشته محاسباتی)، حافظه موقت RAM ۳۲ گیگابایت و سیستم عامل Windows 11 استفاده شد. قابل ذکر است اندازه هر بسته آموزش

جدول ۲- هایپرپارامترهای مورد استفاده در آموزش مدل‌های مختلف YOLOv8

Table 2- Hyperparameters used in training different YOLOv8 models

| هایپرپارامتر    | مقدار  | توصیف                                                                                                  |
|-----------------|--------|--------------------------------------------------------------------------------------------------------|
| Hyperparameter  | Value  | Description                                                                                            |
| epochs          | 200    | Number of epochs to train for                                                                          |
| patience        | 50     | Epochs to wait for no observable improvement for early stopping of training                            |
| Batch           | 4      | Number of images per batch (-1 for AutoBatch)                                                          |
| Imgsz           | 640    | Size of input images as integer                                                                        |
| device          | gpu    | Device to run on, i.e. cuda device = 0 or device = 0, 1, 2, 3 or device = cpu                          |
| optimizer       | 'auto' | Optimizer to use, choices = [SGD, Adam, Adamax, AdamW, NAdam, RAdam, RMSProp]                          |
| Freeze          | None   | Freeze the first n layers (int or list, optional), or freeze the list of layer indices during training |
| lr0             | 0.01   | Initial learning rate (i.e. SGD = 1E-2, Adam = 1E-3)                                                   |
| Lrf             | 0.01   | Final learning rate (lr0 * lrf)                                                                        |
| momentum        | 0.937  | SGD momentum/Adam beta1                                                                                |
| weight_decay    | 0.0005 | Optimizer weight decay 5E-4                                                                            |
| warmup_epochs   | 3.0    | Warmup epochs (fractions ok)                                                                           |
| warmup_momentum | 0.8    | Warmup initial momentum                                                                                |
| warmup_bias_lr  | 0.1    | Warmup initial bias lr                                                                                 |

## نتایج و بحث

در آموزش همه مدل‌ها تعداد دفعات آموزش ۲۰۰ تنظیم شد. هرچند تا انتهای دفعات آموزش، تابع زیان<sup>۴</sup> در حال کاهش و مقادیر دقت، حساسیت و میانگین دقت متوسط برای تصاویر آموزش در حال افزایش بود. اما برای تصاویر مربوط به ارزیابی مدل‌ها، این موارد در دفعه آموزش حدود ۱۶۰ به بعد روند نسبتاً ثابتی داشتند و بهبود قابل توجهی قابل مشاهده نبود (شکل ۴). این نشان می‌دهد که تعداد دفعه آموزش ۲۰۰ انتخاب خوبی برای آموزش مدل‌ها بود. پس از تکمیل آموزش، مدل مورد آموزش دو فایل که حاوی وزن‌های مدل آموزش‌یافته بود به دست می‌داد. فایل last.pt حاوی وزن‌های مدل در ۲۰۰ امین بار آموزش بود و فایل best.pt حاوی وزن‌های مدل در آن دفعه آموزشی‌ای بود که بهترین عملکرد را داشت. در ارزیابی مدل‌ها از مدل با وزن‌های best.pt استفاده شد. همچنین، مدت زمان لازم برای آموزش برای مدل‌های مختلف متفاوت بود که کمترین آن ۰/۱۹۴ برای مدل YOLOv8n و طولانی‌ترین زمان آموزش مربوط به مدل YOLOv8x و برابر با ۰/۳۸۵ ساعت بود.

ارزیابی مدل‌ها براساس معیارهای دقت<sup>۱</sup>، حساسیت<sup>۲</sup>، معیار F1 و میانگین دقت متوسط<sup>۳</sup> انجام شد. فرمول هر یک از این معیارها به صورت زیر است.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (۱)$$

$$Precision = \frac{TP}{TP + FP} \quad (۲)$$

$$Recall = \frac{TP}{TP + FN} \quad (۳)$$

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \quad (۴)$$

$$AP = \int_0^1 PdR \quad (۵)$$

$$mAP = \frac{1}{n} \sum_{i=0}^n AP_i \quad (۶)$$

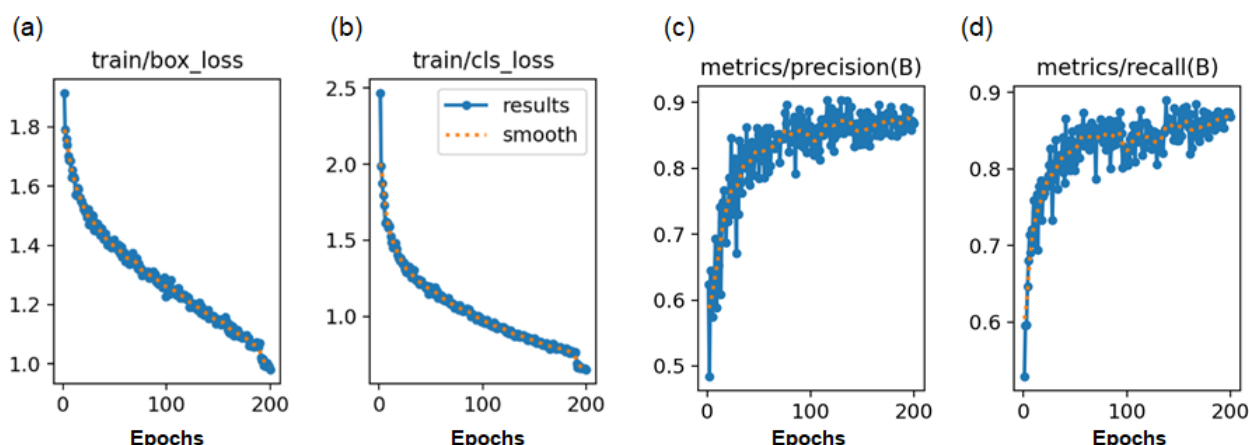
در رابطه‌های (۱) تا (۶)، TP تعداد اشیای درست تشخیص داده شده، FP تعداد اشیایی که در تصویر نیست اما به اشتباه تشخیص داده شده‌اند و FN تعداد اشیایی است که در تصویر وجود دارند ولی به اشتباه تشخیص داده نشده‌اند. همچنین AP مساحت زیر منحنی دقت و حساسیت است.

1- Precision

2- Recall

3- mAP

4- Loss Function



شکل ۴- نمودار تغییرات مقادیر تابع زیان، دقت و حساسیت در طی آموزش مدل YOLOv8n

Fig. 4. Trends in (a) bounding box loss, (b) classification loss, (c) precision, and (d) sensitivity during YOLOv8n model training

اطمینان، بازیابی-اطمینان، امتیاز F1- اطمینان و منحنی دقت- بازیابی، عملکرد مدل YOLOv8n با دقت تشخیص کلی ۸۸/۸۷ درصد بهترین عملکرد را داشت. این مدل در بسیاری از پارامترها بهتر و یا هم‌تراز با سایر مدل‌های حجیم‌تر از خود عمل می‌کند و به‌طور کلی کارایی این مدل در دقت و سرعت تشخیص و یافتن محل قرار گرفتن میوه خیار در تصویر بسیار خوب ارزیابی می‌شود. این موضوع از آنجایی اهمیت دارد که مدل مدنظر برای پیاده‌سازی باید قابلیت تشخیص به‌صورت زمان-واقعی در کاربردهای برخط داشته باشد. خصوصاً در حالتی که رایانه مورد استفاده در پیاده‌سازی مجهز به واحد پردازش گرافیکی (GPU) نباشد.

در این تحقیق چهار مدل مختلف تشخیص اشیاء مبتنی بر یادگیری عمیق، شامل مدل تشخیص اشیاء YOLOv8 در دو مقیاس مختلف، نانو (YOLOv8n) و کوچک (YOLOv8s)، مدل تشخیص YOLOv5 در مقیاس نانو (YOLOv5n)، و مدل مبدل تشخیص زمان-واقعی (RT-DETR) با استفاده از داده‌های آموزشی آموزش داده شدند. در ادامه عملکرد هر یک از این مدل‌ها در تشخیص درست میوه و ساقه متصل به آن و نیز تشخیص صحیح محل قرارگیری آن‌ها در تصویر آورده شده است. با توجه به جدول ۳ و عملکرد چهار مدل آموزش‌یافته با تصاویر پایگاه داده این تحقیق و ارزیابی آن‌ها براساس منحنی‌های دقت-

جدول ۳- مقایسه عملکرد مدل‌های مختلف مورد استفاده در تشخیص کلاس و مکان قرارگیری اشیاء در تصاویر پایگاه داده آموزش و ارزیابی

Table 3- Performance comparison of different models used in detecting the class and location of the objects

| نام مدل<br>Model | دقت<br>Precision |                 | حساسیت<br>Recall |                 | میانگین دقت ۵۰٪<br>mAP50 |                 | میانگین دقت ۵۰ تا ۹۵٪<br>mAP50-95 |                 | برازندگی<br>Fitness |
|------------------|------------------|-----------------|------------------|-----------------|--------------------------|-----------------|-----------------------------------|-----------------|---------------------|
|                  | آموزش<br>Train   |                 | ارزیابی<br>Test  |                 | آموزش<br>Train           |                 | ارزیابی<br>Test                   |                 |                     |
|                  | آموزش<br>Train   | ارزیابی<br>Test | آموزش<br>Train   | ارزیابی<br>Test | آموزش<br>Train           | ارزیابی<br>Test | آموزش<br>Train                    | ارزیابی<br>Test |                     |
| YOLOv5n          | 0.9039           | 0.8841          | 0.8902           | 0.8748          | 0.9082                   | 0.9024          | 0.5598                            | 0.5603          | 0.5945              |
| YOLOv8n          | 0.8990           | 0.8887          | 0.9073           | 0.8922          | 0.9240                   | 0.9207          | 0.5780                            | 0.5787          | 0.6129              |
| YOLOv8s          | 0.8989           | 0.8835          | 0.9153           | 0.8878          | 0.9239                   | 0.9224          | 0.5638                            | 0.5633          | 0.5992              |
| RT-DETR          | 0.9102           | 0.8853          | 0.9622           | 0.8801          | 0.9325                   | 0.9075          | 0.5640                            | 0.5642          | 0.5985              |

گیاه می‌باشد، وجود خطای بالا خصوصاً در تشخیص ساقه متصل به میوه اجتناب‌ناپذیر است. یک راهکار برای کاهش این خطا، توسعه یک الگوریتم دو مرحله‌ای است که در آن ابتدا فقط میوه تشخیص داده شود و سپس الگوریتم دیگری در نواحی سر و ته میوه به دنبال ساقه بگردد. هرچند با دو مرحله‌ای شدن، زمان تشخیص اندکی بیشتر

نتایج عملکرد مدل YOLOv8n برای کلاس‌های مختلف شامل میوه خیار و ساقه متصل به آن در جدول ۴ گزارش شده است. به‌طور مشخص عملکرد این مدل در تشخیص میوه خیار خیلی بهتر از عملکرد آن در تشخیص ساقه می‌باشد. با توجه به این که میوه خیار و ساقه متصل به آن دقیقاً هم‌رنگ با برگ و سایر قسمت‌های ساقه

می‌شود. اما پیش‌بینی می‌شود با این روش دقت تشخیص ساقه به دقت تشخیص میوه نزدیک‌تر شود. این موضوع در تحقیقات آینده مورد بررسی قرار خواهد گرفت.

**جدول ۴- عملکرد مدل YOLOv8n از لحاظ دقت، بازیابی mAP50 و mAP50-95**

**Table 4- Performance of the YOLOv8n model in terms of precision, recall, mAP50, and mAP50-95**

| نام کلاس<br>Class name       | تعداد تصویر<br>Image No. | تعداد نمونه<br>Sample No. | دقت            |                 | حساسیت         |                 | میانگین دقت ۵۰٪ |                 | میانگین دقت ۵۰ تا ۹۵٪ |                 |
|------------------------------|--------------------------|---------------------------|----------------|-----------------|----------------|-----------------|-----------------|-----------------|-----------------------|-----------------|
|                              |                          |                           | Precision      |                 | Recall         |                 | mAP50           |                 | mAP50-95              |                 |
|                              |                          |                           | آموزش<br>Train | ارزیابی<br>Test | آموزش<br>Train | ارزیابی<br>Test | آموزش<br>Train  | ارزیابی<br>Test | آموزش<br>Train        | ارزیابی<br>Test |
| همه کلاس‌ها<br>All           | 90                       | 367                       | 0.899          | 0.889           | 0.907          | 0.892           | 0.924           | 0.921           | 0.578                 | 0.579           |
| خیار<br>Cucumber             | 90                       | 222                       | 0.921          | 0.905           | 0.964          | 0.95            | 0.973           | 0.968           | 0.797                 | 0.796           |
| ساقه متصل به میوه<br>Pedicel | 90                       | 145                       | 0.877          | 0.863           | 0.812          | 0.8             | 0.84            | 0.837           | 0.322                 | 0.325           |

مدل‌های مختلف براساس پارامترهای مهم برای ارزیابی سرعت آن‌ها گزارش شده است. سرعت پردازش در سامانه‌های بینایی ماشین بسیار متأثر از نوع سخت‌افزار است (Hussain, 2024).

مدت‌زمان صرف‌شده برای ارزیابی یک تصویر نیز یک معیار مهم در بررسی عملکرد مدل‌ها می‌باشد. طبق جدول ۵ مدل‌های مختلف مورد استفاده سرعت‌های تشخیص نزدیک به هم داشتند و به‌طور جزئی مدل YOLOv5n عملکرد بهتری داشت. در جدول ۵ عملکرد

**جدول ۵- سرعت تشخیص و اندازه مدل‌های آموزش‌یافته**

**Table 5- Inference time and size of trained models**

| نام مدل<br>Model | تعداد لایه<br>Layers No. | تعداد پارامتر<br>Parameters No. | GFLOPs | زمان آموزش<br>Training time (hr) | دفعات آموزش<br>Epochs | سرعت تشخیص<br>Inference (ms) | حجم مدل<br>Volume (MB) |
|------------------|--------------------------|---------------------------------|--------|----------------------------------|-----------------------|------------------------------|------------------------|
| YOLOv5n          | 193                      | 2503334                         | 7.1    | 0.948                            | 200                   | 4.1                          | 5.3                    |
| YOLOv8n          | 168                      | 3006038                         | 8.1    | 0.502                            | 200                   | 7.7                          | 6.3                    |
| YOLOv8s          | 168                      | 11126358                        | 28.4   | 0.316                            | 123                   | 5.7                          | 22.5                   |
| RT-DETR          | 498                      | 31987850                        | 86.28  | 0.775                            | 80                    | 9.2                          | 66.1                   |

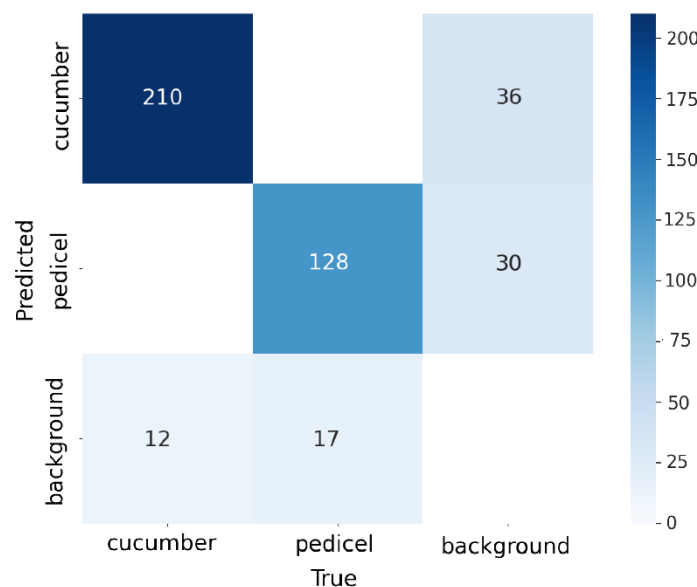
میوه موجود در پایگاه تصاویر ارزیابی به اشتباه پس‌زمینه تشخیص داده شده‌اند. در مقابل ۱۷ مورد (۱۲ درصد) ساقه متصل به میوه موجود در تصاویر به اشتباه زمینه تشخیص داده شده‌اند. نکته دیگر این‌که هیچ موردی از خطای مدل در تشخیص اشتباه میوه به جای ساقه و یا ساقه به جای میوه گزارش نشده است.

به‌منظور بررسی دقیق‌تر عملکرد مدل YOLOv8n، منحنی دقت-بازیابی این مدل بر روی تصاویر ارزیابی در شکل ۶ آورده شده است. این مدل نسبت به مدل هم‌ارز خود از نسخه پنج (YOLOv5n) عملکرد بهتری در دقت تشخیص ساقه داشت. تابع برازندگی<sup>۱</sup> این مدل ۹۱/۰۸ درصد به‌دست آمد. این تابع از میانگین وزنی میانگین دقت متوسط ۵۰ (به میزان اثرگذاری ۱۰٪) و میانگین

با توجه به نتایج جدول ۵، در صورت بهره‌گیری از سخت‌افزار مورد استفاده در این تحقیق، همه مدل‌های موردبررسی را می‌توان در کاربردهای زمان-واقعی استفاده کرد. مقایسه سرعت پردازش بین مدل‌ها زمانی اهمیت پیدا می‌کند که از سیستم‌های مبتنی بر CPU و یا با کارت گرافیک ضعیف‌تر برای پیاده‌سازی استفاده شود. به‌طور مشخص، ماژول‌های سخت‌افزاری Jetson محصول شرکت Nvidia گزینه مناسبی برای اجرای مدل‌های یادگیری عمیق در شرایط مزرعه‌ای می‌باشند. مصرف انرژی کم و توان پردازشی بالا از ویژگی‌های منحصر به فرد این سخت‌افزارها است.

در شکل ۵ ماتریس درهم‌ریختگی مدل YOLOv8n آورده شده است. این ماتریس برای تصاویر دسته ارزیابی این مدل ترسیم‌شده است که مجموعاً حاوی ۲۲۲ شیء از کلاس میوه خیار و ۱۴۵ شیء از کلاس ساقه میوه می‌باشد. به توجه با این شکل، ۱۲ مورد (۵ درصد)

دقت متوسط ۵۰ تا ۹۵ (با میزان اثرگذاری ۹۰٪) به دست می‌آید.

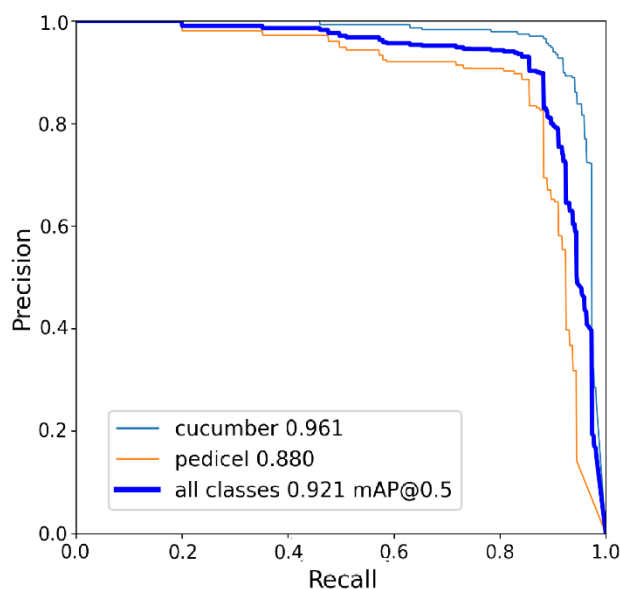


شکل ۵- ماتریس درهم‌ریختگی مدل YOLOv8n

Fig. 5. Confusion matrix of the YOLOv8n model

محسوب می‌شود. به طور کلی، این مدل در مقایسه با سایر مدل‌های مورد بررسی در این تحقیق بهترین عملکرد را در تشخیص کلاس و مکان قرارگیری ساقه متصل به میوه را داشت. صحت تشخیص این مدل در تشخیص میوه خیار و ساقه آن، به ترتیب برابر با ۹۶/۱ و ۸۸/۰ درصد بود.

علی‌رغم دقت خوب مدل YOLOv8n در تشخیص خیار و ساقه متصل به آن، حساسیت این مدل برای تشخیص ساقه بسیار ضعیف‌تر از حساسیت آن در تشخیص میوه است (شکل ۶). حداکثر مقدار امتیاز F1 برای این مدل در درصد اطمینان ۳۹/۵ درصد رخ داده و برابر با ۰/۸۹ است. از آنجایی که این پارامتر میانگین هندسی دقت و حساسیت است، معیار بهتری نسبت به این دو در ارزیابی مدل



شکل ۶- منحنی دقت-حساسیت مدل YOLOv8n

Fig. 6. Precision-Recall Curve of the YOLOv8n Model

## نتیجه گیری

در تشخیص میوه و ساقه بود. غالب مدل‌ها عملکرد نسبتاً خوبی در تشخیص میوه داشتند ولی در تشخیص ساقه متصل به میوه عملکرد خوبی مشاهده نشد. این موضوع می‌تواند به دلیل هم‌رنگ و هم‌شکل بودن ساقه متصل به میوه با سایر بخش‌های ساقه گیاه باشد. برای فائق آمدن بر این چالش، توسعه الگوریتم‌های دو مرحله‌ای که ساقه را صرفاً در قسمتی از تصویر جستجو کند که در آن میوه کشف شده است، پیشنهاد می‌شود. این رویکرد می‌تواند به بهبود عملکرد الگوریتم‌های یادگیری عمیق در تشخیص بهتر ساقه متصل به میوه کمک کند. افزایش تعداد تصاویر پایگاه داده با تمرکز بر بهبود کیفیت نمایش ساقه متصل به میوه در تصاویر نیز می‌تواند بر عملکرد بهتر تشخیص ساقه منجر شود.

## مشارکت نویسندگان

علی‌رضا سلیمانی‌پور: مفهوم‌سازی، روش‌شناسی، جمع‌آوری داده‌ها، پردازش داده‌ها، تصویرسازی نتایج، استخراج و تهیه متن.

چندین مدل تشخیص اشیاء مبتنی بر یادگیری عمیق شامل YOLO (دو نسخه ۵ و ۸) و RT-DETR برای تشخیص میوه خیار و ساقه متصل به آن از روی تصاویر گرفته‌شده در محیط واقعی گلخانه آموزش داده‌شده و ارزیابی شدند. پایگاه تصاویر مورد استفاده شامل ۳۰۰ تصویر گرفته‌شده در شرایط بدون نورپردازی و در اوقات مختلف روز بود. تصاویر مورد استفاده برای آموزش الگوریتم‌ها با تکنیک‌های مختلف به سه برابر افزایش داده شدند تا مدل حاصل با تنوع بیشتری از تصاویر آموزش داده شود. ارزیابی مدل‌های آموزش‌یافته نشان داد، مدل آموزش‌یافته با تکنیک یادگیری انتقالی روی مدل YOLOv8n از سایر مدل‌های مورد نظر عملکرد بهتری در تشخیص میوه و ساقه دارد. به‌طور مشخص، اختلاف عملکرد مدل YOLOv8 با مقیاس‌های مختلف چندان زیاد نبود، ولی نسبت به مدل‌های دیگر برتری داشتند. این برتری در عملکرد را می‌توان به معماری کارآمد، قابلیت‌های استخراج ویژگی قوی و سازگاری با مجموعه داده‌های محدود نسبت داد. نکته دیگر تفاوت عملکرد مدل‌ها

## References

- Alaudeen, K. M., Selvarajan, S., Manoharan, H., & Jhaveri, R. H. (2024). Intelligent robotics harvesting system process for fruits grasping prediction. *Scientific Reports*, 14(1), 2820. <https://doi.org/10.1038/s41598-024-52743-8>
- Bac, C. W., Van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, 31(6), 888-911. <https://doi.org/10.1002/rob.21525>
- Bharad, N. B., & Khanpara, B. M. (2024). Agricultural fruit harvesting robot: An overview of digital agriculture. *Plant Archives*, 24, 154-160. <https://doi.org/10.51470/PLANTARCHIVES.2024.v24.SP-GABELS.023>
- Bochkovski, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *ArXiv Preprint ArXiv:2004.10934*. <https://doi.org/10.48550/arXiv.2004.10934>
- Droukas, L., Doulgeri, Z., Tsakiridis, N. L., Triantafyllou, D., Kleitsiotis, I., Mariolis, I., ..., & Bochtis, D. (2023). A survey of robotic harvesting systems and enabling technologies. *Journal of Intelligent & Robotic Systems*, 107(2), 21. <https://doi.org/10.1007/s10846-022-01793-z>
- FAO. (2023). *Food and Agriculture Organization, 2021 Food and Agriculture Data*. [Online]. <https://www.fao.org/faostat/en/#data/QCL/visualize>
- Fatehi, F., Bagherpour, H., & Amiri Parian, J. (2025). Investigating the Potential of the Innovative YOLOv8s Model for Detecting Bloomed Damask Roses in Open Fields. *Journal of Agricultural Machinery*, (in Press). <https://doi.org/10.22067/jam.2024.88066.1249>
- Fernandez, R., Montes, H., Surdilovic, J., Surdilovic, D., Gonzalez-De-Santos, P., & Armada, M. (2018). Automatic detection of field-grown cucumbers for robotic harvesting. *IEEE Access*, 6, 35512-35527. <https://doi.org/10.1109/ACCESS.2018.2851376>
- Jin, T., & Han, X. (2024). Robotic arms in precision agriculture: A comprehensive review of the technologies, applications, challenges, and future prospects. *Computers and Electronics in Agriculture*, 221, 108938. <https://doi.org/10.1016/j.compag.2024.108938>
- Hussain, M. (2024). Sustainable machine vision for industry 4.0: a comprehensive review of convolutional neural networks and hardware accelerators in computer vision. *AI*, 5(3), 1324-1356. <https://doi.org/10.3390/ai5030064>
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics. <https://ultralytics.com/yolo>
- Kootstra, G., Wang, X., Blok, P. M., Hemming, J., & Van Henten, E. (2021). Selective harvesting robotics: current research, trends, and future directions. *Current Robotics Reports*, 2, 95-104. <https://doi.org/10.1007/s43154-020-00034-1>
- Kuznetsova, A., Maleva, T., & Soloviev, V. (2020). Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot. *Agronomy*, 10(7), 1016. <https://doi.org/10.3390/agronomy10071016>



14. Liu, G., Nouaze, J. C., Touko Mbouembe, P. L., & Kim, J. H. (2020). YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors*, 20(7), 2145. <https://doi.org/10.3390/s20072145>
15. Mahani, S. F. N., & Karami, A. (2023). Maize tassel detection and counting using deep learning techniques. *Journal of Agricultural Machinery*, 13(2), 175-194. <https://doi.org/10.22067/jam.2022.72477.1062>
16. Mallick, P. K. (2022). Evaluating potential importance of cucumber (*Cucumis sativus* L.-Cucurbitaceae): a brief review. *International Journal of Applied Sciences and Biotechnology*, 10(1), 12-15. <https://doi.org/10.3126/ijasbt.v10i1.41466>
17. Mehta, S. S., MacKunis, W., & Burks, T. F. (2016). Robust visual servo control in the presence of fruit motion for robotic citrus harvesting. *Computers and Electronics in Agriculture*, 123, 362-375. <https://doi.org/10.1016/j.compag.2016.02.019>
18. Moseley, K. R., House, L., & Roka, F. M. (2012). Adoption of mechanical harvesting for sweet orange trees in Florida: addressing grower concerns on long-term impacts. *International Food and Agribusiness Management Review*, 15(1030-2016-82762), 83-98. <https://doi.org/10.22004/ag.econ.127108>
19. Park, Y., Seol, J., Pak, J., Jo, Y., Kim, C., & Son, H. I. (2023). Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector. *Computers and Electronics in Agriculture*, 212, 108116. <https://doi.org/10.1016/j.compag.2023.108116>
20. Salim, N. O., & Mohammed, A. K. (2024). Comparative Analysis of Classical Machine Learning and Deep Learning Methods for Fruit Image Recognition and Classification. *Traitement du Signal*, 41(3). <https://doi.org/10.18280/ts.410322>
21. Shewfelt, R. L., & Prussia, S. E. (2022). Challenges in handling fresh fruits and vegetables. In *Postharvest Handling* (pp. 167–186). Elsevier. <https://doi.org/10.1016/B978-0-12-822845-6.00007-0>
22. Sola-Guirado, R. R., Castro-Garcia, S., Blanco-Roldán, G. L., Gil-Ribes, J. A., & González-Sánchez, E. J. (2020). Performance evaluation of lateral canopy shakers with catch frame for continuous harvesting of oranges for juice industry. *International Journal of Agricultural and Biological Engineering*, 13(3), 88-93. <https://doi.org/10.25165/j.ijabe.20201303.4998>
23. Tajane, K., Khutale, P., Kore, A., Khairmode, A., & Girawale, S. (2024). Object Detection: A comprehensive study of famous Deep Learning approaches with case study of Helmet Detection dataset. In *2024 2nd International Conference on Advancements and Key Challenges in Green Energy and Computing (AKGEC)* (pp. 1-5). IEEE. <https://doi.org/10.1109/AKGEC60574.2024.10507194>
24. Wu, D., Lv, S., Jiang, M., & Song, H. (2020). Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Computers and Electronics in Agriculture*, 178, 105742. <https://doi.org/10.1016/j.compag.2020.105742>
25. Xiao, F., Wang, H., Li, Y., Cao, Y., Lv, X., & Xu, G. (2023). Object detection and recognition techniques based on digital image processing and traditional machine learning for fruit and vegetable harvesting robots: An overview and review. *Agronomy*, 13(3), 639. <https://doi.org/10.3390/agronomy13030639>
26. Yue, X., Qi, K., Na, X., Zhang, Y., Liu, Y., & Liu, C. (2023). Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage. *Agriculture*, 13(8), 1643. <https://doi.org/10.3390/agriculture13081643>
27. Zhang, Z., Igathinathane, C., Li, J., Cen, H., Lu, Y., & Flores, P. (2020). Technology progress in mechanical harvest of fresh market apples. *Computers and Electronics in Agriculture*, 175, 105606. <https://doi.org/10.1016/j.compag.2020.105606>
28. Zhou, H., Wang, X., Au, W., Kang, H., & Chen, C. (2022). Intelligent robots for fruit harvesting: Recent developments and future challenges. *Precision Agriculture*, 23(5), 1856-1907. <https://doi.org/10.1007/s11119-021-09836-5>