



سجش از دور

GIS ایران



سنجش از دور و GIS ایران سال چهاردهم، شماره چهارم، زمستان ۱۴۰۱  
Iranian Remote Sensing & GIS Vol.14, No. 4, Winter 2023

۱-۱۸

مقاله پژوهشی

## مقایسه روش‌های یادگیری عمیق و طبقه‌بندی‌کننده آبخاری در تشخیص اشیا،

### در فرود اتوماتیک پرنده‌های بدون سرنشین

بهروز مرادی<sup>۱\*</sup>، عباسعلی مهربان<sup>۲</sup>، مرتضی محمدی<sup>۳</sup>

۱. کارشناس ارشد فتوگرامتری، دانشکده مهندسی نقشه‌برداری، دانشگاه صنعتی خواجه نصیرالدین طوسی
۲. دکتری هوافضا، دانشکده هوافضا، دانشگاه فردوسی مشهد

تاریخ پذیرش مقاله: ۱۳۹۸/۰۸/۰۹

تاریخ دریافت مقاله: ۱۳۹۸/۰۶/۰۳

#### چکیده

فرود اتوماتیک یکی از موضوعات و چالش‌های مهم در حوزه کنترل و اتوماسیون پهپادهای بدون سرنشین است. توسعه الگوریتم‌های فرود اتوماتیک نیازمند تعیین موقعیت پهپاد نسبت به محل فرود است که این کار، در حوزه‌های پردازش تصویر، به تشخیص دقیق و سریع محل فرود نیاز دارد. از جمله روش‌های معمول، در این زمینه، طبقه‌بندی‌کننده آبخاری و تناظریابی و قطعه‌بندی تصویر است که به نظر می‌رسد، با تغییرات آب‌وهوایی و مقیاس متفاوت، این الگوریتم‌ها با چالش مواجه شوند. از طرف دیگر، در سال‌های اخیر شبکه‌های کانولوشنی عمیق به‌منزله مدل‌هایی قوی به‌منظور شناسایی و تشخیص اشیا در تصاویر به‌کار رفته‌اند؛ با این حال با توجه به بار محاسباتی زیاد، این مدل‌ها هنوز در حوزه پرنده‌های بدون سرنشینی که از لحاظ سخت‌افزاری سبک‌اند و قدرت پردازش ضعیفی دارند، کاربرد جدی نیافته‌اند. هدف این مقاله مقایسه دو روش شبکه‌های عمیق کانولوشنی و طبقه‌بندی‌کننده آبخاری برای تشخیص آنی محل فرود است. نتایج عملی کردن روش ارائه‌شده روی یک پرنده Parrot AR Drone 2.0 نشان می‌دهد که شبکه‌های کانولوشنی در مقابل دوران، مقیاس، انتقال و حتی پنهان‌شدگی پایداری بسیار زیادی دارند. دقت تشخیص در این روش ۹۹/۱ است که، در قیاس با روش طبقه‌بندی‌کننده آبخاری، ۳٪ بیشتر است و درعین حال از لحاظ سرعت نیز، مناسب کاربردهای آنی است.

**کلیدواژه‌ها:** یادگیری عمیق، فرود اتوماتیک، پرنده‌های بدون سرنشین.

\* نویسنده مکاتبه‌کننده: تهران، خیابان ولیعصر، تقاطع میرداماد غربی، دانشگاه صنعتی خواجه نصیرالدین طوسی، دانشکده مهندسی نقشه‌برداری. تلفن: ۸۸۷۷۰۲۱۸

## ۱- مقدمه

سامانه‌های GPS و INS برای فواصل دور و دقت اندک در پرواز مناسب‌اند و در فواصل نزدیک ممکن است مشکل‌ساز باشند. در مقایسه با سامانه‌های یادشده، وزن اندک، مصرف کم انرژی و ابعاد کوچک ناوبری از مزیت‌های استفاده از دوربین در ناوبری است. علاوه‌براین تجهیزات مذکور تحت تأثیر عوامل خارجی نیستند و این روش فقط به دوربین و تکنیک‌های پردازش تصویر نیاز دارد و بنابراین، در بسیاری از سامانه‌های ناوبری در سال‌های اخیر، به کار رفته‌اند.

در بیشتر تحقیقات انجام‌شده در حوزه فرود اتوماتیک با استفاده از پردازش تصویر، از روش‌های مرسوم و رایج بهره می‌برند؛ روش‌هایی مانند تبدیلات هاف<sup>۳</sup> (Tsai et al., 2006; Verbandt et al., 2014; Pan et al., 2008; Saripalli et al., 2003; Fan et al., 2008) ممان تصویر (Chen & Liu, 2016; Saripalli et al., 2003) و تناظریابی (Miller et al., 2008; Fucen et al., 2009). به‌تازگی روش طبقه‌بندی‌کننده آبخاری<sup>۴</sup> (De Oliveira & Wehrmeister, 2016) در شناسایی محل فرود به کار رفته است که الگوریتمی مناسب برای شناسایی محسوب می‌شود. این تکنیک‌ها توانسته‌اند، در بیشتر شرایط، عملکرد مناسبی داشته باشند؛ با این حال چالش‌های اصلی این نوع الگوریتم‌ها در تشخیص محل فرود، درموردی همچون تغییرات پس‌زمینه متنوع، شرایط آب‌وهوایی نامناسب و تغییرات ارتفاعی زیاد است.

برخلاف روش‌های برای استخراج ویژگی، شبکه‌های عصبی عمیق<sup>۵</sup> از چندین لایه مخفی استفاده می‌کنند تا ویژگی‌های چندسطحی و سلسله‌مراتبی را به صورت اتوماتیک کشف کنند. در یادگیری عمیق، امید به جایگزینی استخراج ویژگی‌ها به دست بشر، با روش‌های کامل خودکار بدون نظارت و نیمه‌نظارتی وجود دارد.

پرنده‌های بدون سرنشین، در سال‌های اخیر، توجه بسیاری را به خود جلب کرده‌اند و کاربردهای بسیار متنوعی یافته‌اند؛ مانند نظارت، جست‌وجو، کشاورزی، گشت مرزی، زمینه‌های تجارتي و اقتصادی و زمینه‌های علمی. یکی از مسائل مهم و جالب‌توجه، در این حوزه‌ها، فرود اتوماتیک در شرایط محیطی و نوری متفاوت است (Chen & Liu, 2016). با پیشرفت تکنولوژی رایانه‌ها و الگوریتم‌های پردازش تصویر، فرود امن و اتوماتیک براساس تکنیک‌های پردازش تصویر در پرنده‌های بدون سرنشین توجه بسیاری از محققان را به خود جلب کرده است. فرود اتوماتیک نیازمند تشخیص دقیق و تخمین موقعیت پرنده نسبت به محل فرود است که معمولاً با استفاده از سامانه تعیین موقعیت ماهواره‌ای<sup>۱</sup> نمی‌توان آن را انجام داد. دقت عملیات فرود باید بسیار بالا باشد؛ در غیر این صورت، ممکن است پرنده سقوط کند. فرود اتوماتیک مبتنی بر پردازش تصویر پرنده بدون سرنشین شامل دریافت تصویر از باند فرود و سپس پردازش سریع، دقیق و مطمئن است. پس از آن تصویر پردازش‌شده برای تعیین وضعیت پرنده نسبت به محل فرود و ترکیب آن با دیگر سنسورهای پرنده، به منظور فرود امن، استفاده می‌شود.

در حال حاضر، سامانه‌های اتوماتیک به منظور فرود امن شامل سامانه ناوبری اینرسی<sup>۲</sup>، سامانه تعیین موقعیت ماهواره‌ای، سامانه‌های مبتنی بر تصویر و ترکیب سامانه‌های بیان‌شده است (Wang et al., 2007). سامانه ناوبری اینرسی یکی از سامانه‌های اصلی و پرکاربرد در تعیین موقعیت به‌شمار می‌رود. این سامانه پارامترهای ناوبری، شامل موقعیت و سرعت را با استفاده از انتگرال محاسبه می‌کند؛ بدین ترتیب، خطای تجمعی افزایش می‌یابد و احتمالاً تأثیر زیادی در اندازه‌گیری دقیق دارد (Kong et al., 2014). یکی از مشکلات سامانه‌های مبتنی بر تعیین موقعیت ماهواره‌ای در دسترس نبودن همیشگی، قطع شدن سیگنال در کنار ساختمان‌ها، موانع و دقت اندک آنهاست. در حالت کلی،

1. Global Positioning System (GPS)
2. Inertial Navigation System (INS)
3. Hough
4. Cascade Classifier
5. Deep Neural Networks

شده، هنوز قابلیت استفاده از آن در پهپادها و تعیین موقعیت بررسی نشده است. هدف این مقاله مقایسه روش مبتنی بر شبکه کانولوشنی، با عنوان Tiny-Yolo و طبقه‌بندی کننده آبخاری است. در ادامه، بر تحقیقات پیشین مروری خواهیم داشت و سپس در بخش روش‌ها، دو روش طبقه‌بندی کننده آبخاری و Tiny-Yolo را توضیح می‌دهیم. در بخش اجرا، این روش‌ها روی پرندۀ Parrot AR Drone2.0 اجرا می‌شود و به منظور ارزیابی این روش‌ها، سرعت و دقت آنها مقایسه می‌شود.

## ۲- پیشینه پژوهش

در این بخش، ابتدا روش‌های تشخیص محل فرود، با استفاده از الگوریتم‌های پردازش تصویر، مرور می‌شود. سپس روش‌های یادگیری عمیق برای تشخیص شیء مورد بررسی قرار می‌گیرد.

### ۲-۱- تشخیص محل فرود با استفاده از روش‌های پردازش تصویر

برای شناسایی محل فرود و تعیین وضعیت پرندۀ، اغلب از شیوه‌های مرسوم در پردازش تصویر استفاده شده است. برای نمونه، تسای<sup>۳</sup> و همکاران (۲۰۰۶) از راه تشخیص لبۀ کنی<sup>۴</sup> و تشخیص خط، با استفاده از تبدیلات هاف، توانستند محل فرود «T» شکل را تشخیص دهند. آنها سپس، با استفاده از تبدیل مختصات بین سامانه مختصات دوربین و سامانه مختصات زمین، پارامترهای  $\alpha$ ،  $\beta$  و  $\gamma$  را محاسبه کردند و توانستند خطای  $4/8$ ،  $4/2$ ،  $4/6$  درجه را، به ترتیب برای  $\alpha$  و  $\beta$  و  $\gamma$  به دست آورند. سریپالی<sup>۵</sup> و همکاران (۲۰۰۳) یک سامانه اتوماتیک فرود پیشنهاد دادند.

فرایند یادگیری مدل‌های عمیق معمولاً نیازمند داده‌های زیاد است که خوشبختانه در سال‌های اخیر این مشکل، با ایجاد پایگاه داده‌های بزرگ مانند ImageNet (Russakovsky et al., 2015)، حل شده است. همچنین این الگوریتم‌ها به سامانه‌های سخت‌افزاری با قدرت محاسباتی بالا، مانند واحدهای پردازش گرافیکی<sup>۱</sup>، نیاز دارند که ظاهراً استفاده از این تکنیک‌ها، در پردازش‌های آنی روی سخت‌افزارهای سبک و ارزان در پرندۀ‌های بدون سرنشین تقریباً ناممکن است.

در سال‌های اخیر، الگوریتم‌های بسیاری با شیوه یادگیری عمیق برای تشخیص شیء معرفی شده است (Guo et al., 2016). در میان ساختارهای موجود در این زمینه، ساختار شبکه‌های کانولوشنی<sup>۲</sup> را محققان بسیاری برای تشخیص اشیا استفاده کرده‌اند (Redmon et al., 2016; Sermanet et al., 2013; Ren et al., 2015).

با توجه به پتانسیل بالای این الگوریتم‌ها در حوزه تشخیص شیء مانند YOLO (Redmon & Farhadi, 2016) SSD (Liu et al., 2016) و YOLOv2 (Russakovsky et al., 2015)، به نظر می‌رسد این تکنیک‌ها بتوانند در مسائل دنیای واقعی پرندۀ‌های بدون سرنشین، مانند فرود اتوماتیک و تشخیص شیء، کارآمد باشند. در سال‌های اخیر، تلاش‌های بسیاری به منظور کاهش زمان اجرای الگوریتم‌های تشخیص اشیا، با استفاده شبکه‌های کانولوشنی در تصاویر پهپادها صورت گرفته (De Oliveira & Wehrmeister, 2017; Song han et al., 2015; Lee et al., 2016) اما، با توجه به بار محاسباتی زیاد، امکان استفاده از آنها در پهپادهای سبک دشوار است. در بین الگوریتم‌های شبکه‌های کانولوشنی بر مبنای تشخیص شیء، روش YOLOv2 سرعت بسیار بالا و بار محاسباتی پایینی دارد. ساختار کم‌عمق این ایده، به شکل ساده‌تر، در قالب Tiny-Yolo ارائه شده است. اگرچه معماری این روش برای تشخیص اشیا در مقالات گوناگونی استفاده

1. Graphics Processing Unit (GPU)
2. Convolutional Neural Networks
3. Tsai
4. Canny
5. Saripalli

دارد). چن و لی یو<sup>۵</sup> (۲۰۱۶) الگوریتم‌های فرود اتوماتیک را با استفاده از تکنیک‌های پردازش تصویر، بررسی کرده‌اند. بسیاری از الگوریتم‌های ذکر شده در این مقاله از الگوریتم‌های رایج شمرده می‌شوند و در عین حال در مسائل دنیای واقعی ناکارآمدند. بیشتر الگوریتم‌ها مبتنی بر بهبود تصویر، آستانه‌گذاری، تبدیلات هاف، تشخیص لبه، ممان اینرسی و قطعه‌بندی گزارش شده‌اند که به نظر می‌رسد در مسائل واقعی مانند تغییر شرایط نوری، تنوع پیش‌زمینه و پنهان‌شدگی جسم کارآمدی لازم را ندارند. د اولیویرا<sup>۶</sup> و ورمایستر (۲۰۱۶) روش‌های طبقه‌بندی کننده آبشاری را در شناسایی محل فرود مطرح کرده‌اند. روش کلی توصیفگرهای چندمقیاسه<sup>۷</sup> میانگین مقادیر قطعات بلوک به جای تک پیکسل را محاسبه می‌کند. از دیگر سو، ویژگی‌های Haar با اختلاف شدت پیکسل‌های موجود در یک ناحیه تعیین می‌شوند. هر یک از این توصیفگرهای ذکر شده با استفاده از طبقه‌بندی کننده آبشاری که شامل چندین طبقه‌بندی کننده ضعیف و پی‌درپی می‌شوند، می‌تواند تصویر ورودی را تفسیر کند.

## ۲-۲- تشخیص شیء با استفاده از یادگیری عمیق

بسیاری از محققان در حوزه ماشین بینایی، شبکه‌های کانولوشنی عمیق را در حوزه تشخیص شیء به کار برده‌اند. یکی از نخستین روش‌ها، در این حوزه، روش R-CNN (Girshick et al., 2014) است که ابتدا نواحی منتخب بالقوه را در تصویر تولید و سپس طبقه‌بندی کننده را روی این نواحی برگزیده، ارزیابی می‌کند. زمان اجرا در این روش، به‌ویژه در مورد تولید نواحی بالقوه در

در این سامانه، با اعمال فیلتر مدین روی تصویر ورودی، سپس قطعه‌بندی و اجزای متصل<sup>۱</sup> و در نهایت، استفاده از ممان اینرسی، موقعیت محل فرود در تصویر محاسبه شد. نتایج گویای دقت ۷ سانتی‌متر برای جابه‌جایی و دقت ۴ درجه برای دوران بود. فوسن<sup>۲</sup> و همکاران (۲۰۰۹) در مقاله‌ای، برای تشخیص جهت پهپاد در طراحی محل فرود، شکلی مثلثی به شکل «H» اضافه کردند. سپس برای شناسایی آن، الگوریتم جدیدی براساس تناظریابی پیشنهاد دادند و نیز تبدیلات هاف را برای شناسایی مثلث به کار بردند که شیوه‌ای سریع و مطمئن بود. طبق خروجی این الگوریتم، زمان اجرا ۶۰ میلی‌ثانیه و دقت ۹۷/۴۲٪ بود.

فن<sup>۳</sup> و همکاران (۲۰۰۸) با بهبود تصویر آستانه‌گذاری، سپس قطعه‌بندی تصویر آستانه‌گذاری شده و اعمال ممان اینرسی، محل فرود را در تصویر شناسایی کردند. برای تعیین وضعیت پرند، ابتدا مرکز محل فرود و سپس پیکسلی که بیشترین فاصله را از آن دارد مشخص می‌شود. پس از آن قطر اصلی، با استفاده از این دو پیکسل، تعیین خواهد شد. ممکن است برای شکل «H» یک قطر منجر به دو وضعیت شود. برای محاسبه زاویه بین دو نقطه از گوشه H، تصویر اصلی به کار می‌رود و سپس نقطه اصلی، به منظور تعیین وضعیت، شناسایی می‌شود. در نهایت، زاویه  $\gamma$  که نشان دهنده زاویه دوران حول محور z است، با اندازه‌گیری زاویه بین نقاط گوشه و شیب قطر اصلی، محاسبه خواهد شد. در این مقاله، دقت موقعیتی ۱۰ سانتی‌متر و دقت زاویه‌ای ۳ درجه گزارش شد.

چی<sup>۴</sup> و همکاران (۲۰۱۳) از دو دوربین استریو برای تشخیص و تعیین موقعیت استفاده کردند. از مشکلات تناظریابی تصاویر استریو، امکان تشخیص نقاط اشتباه به جای نقاط گوشه است. برای حل این مسئله، این محققان اختلاف گرادیان نقاط گوشه و نقاط غیر گوشه را محاسبه کردند. این مقدار نمایانگر اختلاف بین نقاط گوشه و نقاط غیر گوشه است (هرچه مقدار این عدد بیشتر باشد، نقطه گوشه بیشتر به نقطه غیر گوشه تعلق

1. Connected Component

2. Fucen

3. Fan

4. Chi

5. Chen & Liu

6. De Oliveira & Wehrmeister

7. Multi Scale Block LBP

(۲۰۱۶) و نیز با همکاری فرهادی<sup>۴</sup> (۲۰۱۶) با عنوان YOLO و YOLOV2، توانست هم از لحاظ دقت و هم از لحاظ سرعت (در بهترین حالت، ۹۰ فریم بر ثانیه) از بیشتر سامانه‌های تشخیص شیء پیشی بگیرد. در این مقاله‌ها، بخش‌های متفاوت سامانه تشخیص شیء، به صورت سامانه‌ای واحد در شبکه عصبی کانولوشنی ایجاد شده‌اند. با توجه به اینکه کل سامانه تشخیص در یک شبکه جای می‌گیرد، فرایند یادگیری می‌تواند به صورت یکپارچه انجام شود.

### ۳- روش‌های تشخیص محل فرود

در این مقاله، دو روش طبقه‌بندی‌کننده آشناری و ویژگی‌های Haar و Tiny-YOLO مقایسه می‌شوند. هریک از این دو روش در این بخش توضیح داده خواهد شد.

#### ۳-۱- طبقه‌بندی‌کننده آشناری

طبقه‌بندی‌کننده‌های آشناری از طبقه‌بندی‌کننده‌های سلسله‌مراتبی‌اند که خروجی طبقه‌بندی‌کننده‌های قبلی‌شان را به صورت اطلاعات اضافی، برای ورودی طبقه‌بندی‌کننده بعدی، به کار می‌برند. طبقه‌بندی‌کننده‌های آشناری شامل طبقه‌بندی‌کننده‌های ضعیف و غیردقیق می‌شوند. با پشت‌سرهم قرار گرفتن این طبقه‌بندی‌کننده‌های ضعیف، طبقه‌بندی‌کننده قوی به دست می‌آید. نمونه‌های متفاوت از طریق هر طبقه‌بندی‌کننده به منزله نمونه مثبت (نمونه حاوی شیء) یا نمونه منفی (نمونه بدون شیء) برچسب‌گذاری می‌شود. اگر خروجی منفی باشد، فرایند متوقف می‌شود و در غیر این صورت، فرایند وارد مرحله بعدی می‌شود؛ بنابراین خطا در هر مرحله باید حداقل باشد زیرا در

تصویر، بسیار زیاد است؛ به همین دلیل، استفاده عملی از این روش‌ها، به‌ویژه در کارهای آنی بسیار ناممکن به نظر می‌رسد. از دیگر روش‌ها Fast R-CNN (Ren et al., 2015) و Girshick, (2015) هستند که از شبکه کانولوشنی برای تولید نواحی بالقوه در تصویر، به منظور افزایش سرعت R-CNN بهره برده‌اند. سرعت این الگوریتم‌ها به ترتیب ۰/۵ و ۷ فریم بر ثانیه، روی واحد پردازنده گرافیکی است که مناسب کاربردهای دقیق و آنی در پرنده‌های بدون سرنشین نیست.

در حوزه تشخیص اشیا در تصاویر پهپاد، با استفاده از یادگیری عمیق، برخی محققان از شبکه‌های کانولوشنی عمیق استفاده کرده‌اند. در مقاله د اولیویرا و ورمایستر (۲۰۱۶)، شبکه‌های کانولوشنی عمیق برای تشخیص افراد در تصاویر پهپاد استفاده شده است. به منظور کاهش جست‌وجوی مناطق منتخب، از پردازش تصویر حرارتی استفاده شد. در بهترین حالت وقتی دقت و زمان اجرا مهم باشد، با ترکیب شبکه‌های کانولوشنی عمیق و پردازش تصویر حرارتی، امکان اجرای این روش با ۱/۰۸ فریم بر ثانیه وجود دارد. لی<sup>۱</sup> و همکاران (۲۰۱۷) از پردازش ابری، با هدف افزایش سرعت تشخیص اشیا با استفاده از شبکه‌های کانولوشنی و الگوریتم R-CNN، استفاده کردند. با توجه به سنگین بودن محاسبات، پردازش‌های انجام‌شده به رایانش ابری انتقال یافت. زمان اجرای گزارش‌شده، برای واحد پردازنده گرافیکی لپ‌تاپ، حدود ۰/۶ ثانیه و در محاسبات ابری، نزدیک ۰/۱ ثانیه بود. سانگ هان<sup>۲</sup> و همکاران (۲۰۱۵) شبکه‌های کانولوشنی را در تشخیص و ردیابی به کار بردند. آنها از روش KCF (Henriques et al., 2015) که سرعت اجرای بالایی دارد، برای ردیابی اشیا استفاده کردند؛ در عین حال زمانی که دقت ردیابی از حد آستانه‌ای کمتر باشد، از روش Faster-R-CNN برای تشخیص بهره می‌برند که زمان اجرای بسیار پایینی دارد. سرعت تشخیص و ردیابی با استفاده از Nvidia ۹۸۰ GTX، به ترتیب ۱۷۰ و ۵/۵ میلی‌ثانیه بود.

معماری معرفی‌شده از سوی ردمان<sup>۳</sup> و همکاران

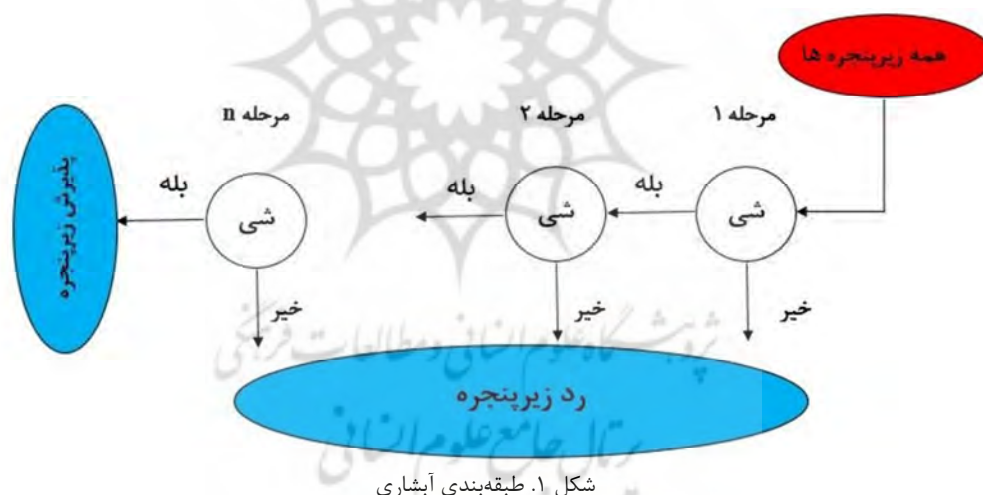
1. Lee
2. Song Han
3. Redmon
4. Farhadi

مقیاس‌های متفاوت انجام می‌شود. طبقه‌بندی‌کننده‌های پایه، در این زمینه، درخت‌های تصمیم‌گیری<sup>۱</sup> با دست‌کم دو برگ به‌شمار می‌روند. ویژگی‌های Haar ورودی این طبقه‌بندی‌کننده‌هاست و مطابق شکل ۲ محاسبه می‌شوند. ویژگی استفاده‌شده در طبقه‌بندی‌کننده‌ای خاص با استفاده از شکل، موقعیت و مقیاس آن مشخص خواهد شد. هر مرحله با تکنیکی به نام boosting آموزش داده می‌شود. در مرحله تشخیص، پنجره‌ای متناسب با ابعاد شیء روی تصویر حرکت می‌کند. در هر ناحیه، ویژگی‌های Haar محاسبه می‌شود و با محاسبه تفاضل و مقایسه آن با حد آستانه آموزش‌دیده، نواحی شامل شیء از غیرشیء تشخیص داده می‌شود (Viola & Jones, 2001). این ویژگی‌ها در دسته‌بندی قطعات گوناگون در تصویر به‌کار می‌روند.

مرحله بعدی تصحیح نمی‌شود (De Oliveira & Wehrmeister, 2016). شکل ۱ ساختار این طبقه‌بندی‌کننده‌ها را نشان می‌دهد.

ویژگی‌های Haar شامل مستطیل‌هایی سیاه و سفید می‌شود. در شکل ۲، این ویژگی‌ها نواحی همسایه در یک قطعه از تصویر را در نظر می‌گیرند و با محاسبه تفاضل جمع شدت پیکسل‌ها در هر ناحیه استخراج می‌شوند؛ به این ترتیب می‌توان تغییر شدت در مناطق متفاوت تصویر را محاسبه کرد و در نهایت، به الگویی معنی‌دار دست یافت.

طبقه‌بندی‌کننده‌ها به‌گونه‌ای طراحی می‌شوند که ابعادشان به‌سادگی تغییر یابد تا بتوانند اشیای دارای ابعاد متفاوت را شناسایی کنند؛ بنابراین برای پیدا کردن اشیای دارای ابعاد متفاوت، فرایند استنتاج در



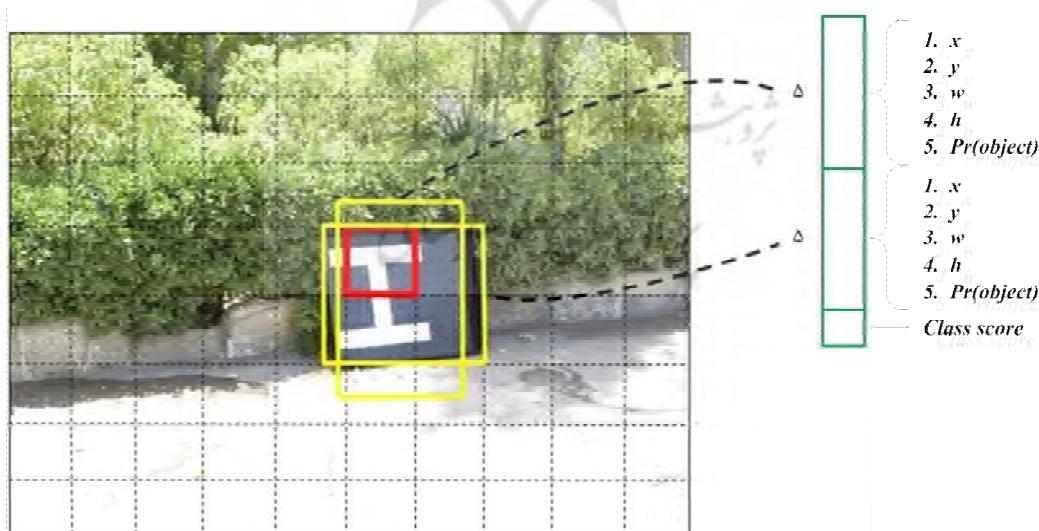
شکل ۲. ویژگی‌های استفاده‌شده در طبقه‌بندی‌کننده آبخاری

## 1. Decision Tree

### ۲-۳- Tiny-Yolo روش

الگوریتم‌های معرفی شده در پژوهش‌های ردمان و فرهادی (۲۰۱۶) و روساکوفسکی<sup>۱</sup> و همکاران (۲۰۱۵)، با عنوان YOLO و YOLOv2 بر دیگر الگوریتم‌های تشخیص شیء مزیت‌هایی دارند؛ نخست اینکه، در این الگوریتم‌ها، به تولید نواحی بالقوه در تصویر نیاز نیست. در روش‌های معمولی تشخیص شیء، فرایند تولید نواحی بالقوه با طبقه‌بندی، پالایش و حذف تکراری این نواحی دنبال می‌شود. در نهایت، همه این نواحی با توجه به دیگر اشیای موجود در تصویر، نمره‌دهی می‌شوند. این کار در همه نواحی و مقیاس‌های گوناگون تصویر اعمال و نواحی دارای امتیاز بالا به‌منزله شیء در نظر گرفته می‌شود. فرایند تولید نواحی بالقوه سبب افزایش محاسبات بسیار زیادی می‌شود که در الگوریتم YOLO، این نیاز برطرف شده است. فرایند به‌کاررفته برای تشخیص شیء در YOLO (Redmon & Farhadi, 2016) به‌صورت مسئله رگرسیون<sup>۲</sup> از ورودی تصویر تا مختصات مستطیل دربردارنده شیء و احتمال کلاس انجام می‌شود. شبکه کانولوشنی طراحی شده در YOLO الهام‌گرفته از شبکه

GoogleNet (Szegedy et al., 2015) است. شبکه YOLO شامل ۲۴ لایه کانولوشنی و دو لایه تمام‌اتصال می‌شود. لایه آخر این شبکه کلاس و مختصات مستطیل دربردارنده شیء را پیش‌بینی می‌کند. یکی دیگر از مزیت‌های این الگوریتم ارزیابی کل تصویر، به‌جای نواحی بالقوه است که پتانسیل این الگوریتم را افزایش می‌دهد. در فرایند تشخیص، تصویر به‌صورت قطعات مساوی  $S \times S$  ( $S=7$ ) گریدهای می‌شود. هر یک از این قطعات  $B$  ( $B=2$ ) مستطیل را پیش‌بینی می‌کند. هر یک از مستطیل‌ها این اطلاعات را دربر دارد: مختصات مرکز مستطیل  $x$  و  $y$ ، طول  $w$  و عرض  $h$  مستطیل پیش‌بینی شده و احتمال حضور شیء در آن مستطیل  $Pr(object)$ . در آخر، خروجی  $S \times S \times (B \times 5 + C)$  تنسور است که در آن،  $C$  تعداد کل کلاس‌های موجود شمرده می‌شود. آموزش شبکه با الگوریتم پس‌انتشار خطا انجام شد. چنین ساختاری زمان اجرا را به‌شدت کاهش می‌دهد و این الگوریتم را مناسب پردازش‌های آنی می‌سازد. شکل ۳ یک تنسور را نشان می‌دهد.



شکل ۳. عملکرد YOLO

1. Russakovsky
2. Regression

جدول ۱. شبکه کانولوشنی Tiny-YOLO

ابعاد فیلتر	تعداد فیلترهای کانولوشن	لایه
۲۸۸×۲۸۸ تصویر ورودی		
۳×۳	۱۶	Conv-maxpool
۳×۳	۳۲	Conv-maxpool
۳×۳	۶۴	Conv-maxpool
۳×۳	۱۲۸	Conv-maxpool
۳×۳	۲۵۶	Conv-maxpool
۳×۳	۵۱۲	Conv-maxpool
۳×۳	۱۰۲۴	Conv
۳×۳	۵۱۲	Conv
۱×۱	۴۲۵	Conv
۱۱×۷×۷	-	تنسور

#### ۴- اجرا و ارزیابی نتایج

در این بخش به تجهیزات مورد استفاده، جزئیات اجرا و داده‌های به کاررفته در بخش‌های متفاوت اشاره می‌شود.

#### ۴-۱- تجهیزات مورد استفاده

پرنده مورد استفاده در تست الگوریتم Parrot AR.Drone 2.0 بوده که پلتفرمی کم‌هزینه، سبک و کوچک است (ابعاد آن حدود ۵۰×۵۰ و وزن آن ۴۲۰ گرم است). پرنده Parrot AR.Drone 2.0 به دو دوربین جلو و پایین، IMU شامل ژيروسکوپ سه‌محوره، شتاب‌سنج سه‌محوره، مغناطیس‌سنج سه‌محوره و حسگرهای فشارسنج و اولتراسونیک مجهز است. ابعاد تصویر دوربین جلو ۱۲۸۰×۷۲۰، با قابلیت ارسال سی فریم بر ثانیه و دوربین زیر ۳۲۰×۲۴۰، با قابلیت ارسال شصت فریم بر ثانیه است. پرنده مورد نظر دارای CPU ۱ گیگاهرتز Cortex-A8 و مجهز به سیستم‌عامل لینوکس است. سرعت افقی با استفاده از دوربین زیر محاسبه می‌شود. ارتفاع و وضعیت پرنده

1. Batch Normalization
2. Ren
3. Overfitting

در معماری جدید YOLO، با نام YOLOv2 [5]، تغییراتی با هدف بهبود عملکرد و سرعت YOLO ایجاد شده است. در این روش، از تکنیک نرمال‌سازی دسته‌ای<sup>۱</sup> [w] پس از هر لایه کانولوشن استفاده شد. ورودی تصویر، در مرحله طبقه‌بندی‌کننده در YOLOv2، از ۲۲۴×۲۲۴ به ۴۴۸×۴۴۸ افزایش یافت و این دو تکنیک ۶٪ معیار mAP را در قیاس با YOLO، ۶٪ افزایش داد. به جای معماری لایه تمام‌اتصال، Anchor boxes معرفی شده در پژوهش رن<sup>۲</sup> و همکاران (۲۰۱۵) در شبکه جای‌گذاری شد. مدل ارائه‌شده از اطلاعات خوشه‌بندی K-means در طول و عرض جعبه داده‌های زمینی بهره برد. در نهایت، ابعاد گرید مطابق رابطه (۱) محاسبه شد.

$$\text{Output Size} = (N_{\text{class}} + N_{\text{coords}} + 1) \times N_{\text{anchor}} \quad \text{رابطه (۱)}$$

$N_{\text{class}}$  تعداد کلاس‌ها،  $N_{\text{coords}}$  تعداد مختصات خروجی،  $N_{\text{anchor}}$  تعداد مستطیل‌های anchor است. شبکه YOLOv2، با قابلیت اجرای بسیار سریع و دقت بالا، می‌تواند اشیای موجود در تصویر را شناسایی کند. در دیگر ساختار پیشنهادی شبکه Tiny-YOLO به جای ۲۴ لایه کانولوشن، از ۹ لایه استفاده می‌کند که عملکرد آن با تعداد کمتر پارامترها و زمان اجرای بسیار سریع (زمان اجرای ۱۵۵ فریم بر ثانیه روی GPU) در قیاس با YOLO و YOLOv2 است. اطلاعات هر لایه در جدول ۱ آمده است. بعد از هر لایه کانولوشن، مطابق روش YOLOv2 از نرمال‌سازی دسته‌ای استفاده شده است. این تکنیک می‌تواند، علاوه بر تسریع زمان همگرایی شبکه و کاهش نیاز به تکنیک‌های تنظیم‌سازی مانند بیش‌برازش<sup>۳</sup>، دقت شبکه را نیز بهبود ببخشد. استفاده از لایه ۳×۳ به منظور حفظ اطلاعات ورودی انجام می‌شود زیرا فیلتر با ابعاد زیاد سبب کاهش و ازدست‌رفتن اطلاعات ورودی می‌شود. مطابق YOLOv2، استفاده از تابع فعال خطی leaky rectified پس از این لایه باعث غیرخطی شدن سامانه خواهد شد. فرایند آموزش این شبکه از طریق الگوریتم پس‌انتشار خطا انجام شد.



را از آن به‌دست آورد. شکل محل فرود می‌تواند به‌صورت دایره، مربع، «H»، «T» و یا ترکیبی از این شکل‌ها باشد. زمانی که محل فرود شناسایی شد، اطلاعات موقعیتی پرنده را می‌توان، با تشکیل معادلات طبق مختصات نقاط کلیدی استخراج‌شده از محل فرود، محاسبه کرد (Chen & Liu, 2016). برای آموزش شبکه‌کانولوشنی در حدود هزار عکس از محل فرود، در فاصله‌های متفاوت با پیش‌زمینه‌های متنوع، گرفته شد. محل فرود دربردارنده شکل «H» سفید، با پیش‌زمینه سیاه دارای ابعاد  $70 \times 80$  سانتی‌متر بود که این ابعاد زیاد با توجه به امکان دید کافی در ارتفاع بالا و ساده‌کردن فرایند تشخیص، طراحی شد. از دیگر سو، برای آموزش طبقه‌بندی‌کننده آبخاری، ۲۰۰۰ تصویر دربردارنده شکل محل فرود و ۵۰۰۰ تصویر غیرمرتبط جمع‌آوری شد. تصاویر غیرمرتبط به‌صورت تصادفی از داده‌های اینترنتی گرد آمد. شکل ۵ داده‌های آموزشی حاوی شیء را نشان می‌دهد. برای ارزیابی مدل، نزدیک به ۴۰۰ تصویر دربردارنده شکل «H» و ۲۰۰ تصویر فاقد محل فرود گردآوری شد.

شامل زاویه‌های yaw, pitch, roll با استفاده سنسورهای موجود تخمین زده می‌شود. پردازشگر مورد استفاده روی لپ‌تاپ نیز اینتل COREi7 و ۲/۴ گیگاهرتز و پردازنده گرافیکی آن Nvidia 610m است. پیکربندی Parrot AR Drone 2.0 به‌گونه‌ای است که به پایگاهی زمینی، برای کنترل و دریافت اطلاعات مانند ارسال تصاویر، دریافت و پردازش آن نیاز دارد. پس از انتقال تصویر ورودی به پردازنده لپ‌تاپ، فرایند تشخیص انجام می‌شود؛ سپس موقعیت پرنده نسبت به محل فرود، با الگوریتم پیشنهادی، به‌دست می‌آید و در نهایت، فرمان‌های لازم برای فرود به پرنده فرستاده خواهد شد. Parrot AR Drone 2.0 در شکل ۴ نشان داده شده است.

#### ۴-۲- داده‌های آموزشی و ارزیابی

طراحی شکل محل فرود باید طوری باشد که بتوان به‌راحتی آن را از پس‌زمینه تفکیک کرد و ویژگی‌های لازم به‌منظور تعیین موقعیت پرنده نسبت به محل فرود



شکل ۴. پرنده Parrot AR Drone 2.0



شکل ۵. داده‌های آموزشی برای آموزش شبکه‌های کانولوشنی و طبقه‌بندی‌کننده آبخاری

#### ۳-۴- محل اجرای پژوهش

محل دریافت تصاویر و اجرای الگوریتم‌های اشاره شده محوطه دانشکده مهندسی عمران و نقشه‌برداری دانشگاه صنعتی خواجه نصیرالدین طوسی، واقع در تقاطع خیابان ولیعصر و خیابان میرداماد، در شهر تهران بود. شکل ۶ این محل را روی نقشه هوایی نشان می‌دهد.

#### ۴-۴- اجرا

فرایند آموزش و استنتاج شبکه کانولوشنی در محیط لینوکس، با استفاده از کتابخانه (Tensorflow Abadi et al., 2016) و OpenCV انجام شد. کتابخانه Tensorflow کتابخانه‌ای متن‌باز برای محاسبات عددی، یادگیری ماشینی و یادگیری عمیق است که لایه‌ها و توابع گوناگون در شبکه‌های کانولوشنی را در اختیار می‌گذارد. OpenCV کتابخانه‌ای متن‌باز، دربردارنده الگوریتم‌ها و توابع گوناگون برای پردازش تصویر و یادگیری ماشینی، و بسیار مناسب کارهای آنی است که در این مورد، به منظور آموزش طبقه‌بندی کننده آبخاری به کار رفت. کتابخانه استفاده شده برای هدایت پرنده PS-Drone تحت زبان برنامه‌نویسی پایتون بود. PS-Drone یک رابط برنامه‌نویسی کاربردی است که با هدف هدایت Parrot AR Drone 2.0 ایجاد شده و

امکان و شرایط کار با داده‌های سنسورهای گوناگون، از جمله دوربین را فراهم می‌آورد.

#### ۴-۵- آموزش

در هر شبکه کانولوشنی، لازم است پارامترهایی به منظور آموزش شبکه تعیین شود؛ از جمله تعداد دسته، مومنتوم، مقدار یادگیری، تعداد تکرار، حد آستانه (برای تشخیص شیء). تعداد دسته تعداد تصاویری است که در یک مرحله پیش رو یا پس‌انتشار، وارد شبکه می‌شود و با توجه به امکان پردازش سخت‌افزاری تعیین می‌شود. مقدار یادگیری روند یادگیری در طول فرایند آموزش را کنترل می‌کند. ثابت مومنتوم نیز بهبود میزان یادگیری را تنظیم می‌کند. منظور از اپک ارائه همه تصاویر به شبکه، در طول فرایند یادگیری است. در این آزمایش تعداد دسته، مقدار یادگیری، ثابت مومنتوم، تعداد اپک و حد آستانه در فرایند استنتاج، به ترتیب ۰.۳، ۰.۰۱، ۰.۰۹ و ۱۰۰ و ۰/۳ تنظیم شد. طبق مطالب بخش‌های پیشین، استفاده از تکنیک نرمال‌سازی دسته‌ای نیاز به الگوریتم‌های تنظیم‌سازی، مانند dropout را کاهش می‌دهد؛ بنابراین در اجراء تکنیک dropout به کار نرفت.



شکل ۶. محل اجرای پژوهش

مجموع، ۱۲۵۰ تصویر دریافت شد و در ۱۱۰۰ تصویر، امکان رؤیت محل فرود وجود داشت. روش شبکه کانولوشنی و طبقه‌بندی‌کننده آبخاری، به‌منظور بررسی سرعت اجرا و دقت، روی این تصاویر اجرا شدند. نتایج نشان داد که روش Tiny-YOLO امکان شناسایی محل فرود را با دقت ۹۹/۱ (تعداد ۱۰۹۰ از ۱۱۰۰) دارد؛ در صورتی که ده مورد به اشتباه تشخیص داده شد (نه مورد شناسایی نشد و یک مورد اشتباه شناسایی شد). طبق شکل ۷، نور شدید و تاحدی پنهان‌شدگی در تصویر وسطی و سمت چپ سبب تشخیص داده‌نشدن الگوریتم Tiny-YOLO و نیز طبقه‌بندی‌کننده آبخاری شده است. بیان این نکته لازم است که کیفیت بسیار اندک دوربین زیر پهپاد Parrot AR.Drone 2.0 نیز در این مسئله بی‌تأثیر نیست. علاوه‌براین جنس محل فرود نیز، با توجه به ویژگی‌اش، امکان بازتاب نور بسیار بالایی دارد؛ برای بهبود نتایج، می‌توان از مواد جایگزین بهتری استفاده کرد. از سوی دیگر، در تصویر سمت راست نیز، طبقه‌بندی‌کننده آبخاری محل فرود را اشتباه تشخیص داده است که به‌نظر می‌رسد، هنگام وجود سایه یا نور شدید، با مشکل روبه‌رو شود. معیارهای دقت FNr، TPr، PPV و FDr مطابق معادله‌های زیر محاسبه می‌شود:

$$\begin{aligned} FNr &= \frac{FN}{FN+TP} & PPV &= \frac{TP}{FP+TP} \quad \text{رابطه (۲)} \\ TPr &= \frac{TP}{FN+TP} & FDr &= \frac{FP}{FP+TP} \end{aligned}$$

شایان ذکر است که به‌منظور افزایش داده، می‌توان از راهکارهای تقویت داده استفاده کرد. در این مقاله این کار با انتقال، دوران، تغییرات شدت نور و دیگر موارد روی داده‌های اصلی اجرا شد.

کل مراحل یادگیری آبخاری طی پانزده مرحله انجام شد. در این ارزیابی، از طبقه‌بندی‌کننده بوستینگ و ویژگی‌های Haar استفاده شد. ابعاد نمونه‌های مورد استفاده ۲۴×۲۴، ۰/۹۹ minimal hit rate و maximal و ۰/۵ false alarm برای آموزش آن در نظر گرفته شد.

#### ۴-۶- ارزیابی

معیارهای مناسب ارزیابی فرود اتوماتیک شامل دو بخش می‌شود. بخش نخست دربرگیرنده دقت محل تشخیص است؛ این هدف با ساختار پژوهش حاضر تناسب دارد. بخش دوم دقت پارامترهای فرود را دربر می‌گیرد؛ این هدف با توجه به سخت‌افزار پرنده مورد استفاده ارزیابی می‌شود. معیارهای مناسب برای صحت استخراج محل فرود را می‌توان با معیارهای عملکرد طبقه‌بندی‌ها فراهم آورد چون تصاویر خروجی شناسایی محل فرود، در واقع، به دو کلاس صفر و یک تقسیم‌بندی می‌شوند. کلاس یک نشان‌دهنده وجود محل فرود و کلاس صفر بیانگر نبود آن است. برای ارزیابی الگوریتم از فاصله‌ها و زوایای متفاوت، از محل فرود فیلم‌برداری شد؛ با استخراج فریم‌های آن، در



شکل ۷. وجودداشتن امکان تشخیص در الگوریتم Tiny-YOLO

برای ارزیابی در این بخش به مرحله تشخیص محل فرود در تصویر بازمی‌گردد. این روش‌ها از لحاظ سرعت نیز مقایسه شدند (جدول ۳). زمان اجرای شبکه کانولوشنی در حدود ۱۲۰ میلی‌ثانیه (هشت فریم بر ثانیه) است که به نسبت طبقه‌بندی‌کننده آبخاری، زیاد است اما به نظر می‌رسد بتوان از آن در پردازش‌های آبی استفاده کرد.

شکل ۸ منحنی عملکرد طبقه‌بندی‌کننده‌های گوناگون موسوم به ROC را نشان می‌دهد. نمودار ROC میزان تشخیص درست، برحسب میزان تشخیص اشتباه را طبق حد آستانه‌های متفاوت بررسی می‌کند. در این نمودار، دو نوع طبقه‌بندی‌کننده ارزیابی شده است. از این نمودار، می‌توان دریافت در حالت کلی، الگوریتم Tiny-YOLO از طبقه‌بندی‌کننده آبخاری مناسب‌تر است.

مساحت زیر نمودار ROC با شاخص یا عبارت AUC مشخص می‌شود که نشان‌دهنده برتری طبقه‌بندی‌کننده است (هرچه مساحت بیشتر باشد، طبقه‌بندی‌کننده مناسب‌تر است). در جدول ۴، شاخص مورد نظر در این الگوریتم‌ها محاسبه شده است که نشان می‌دهد الگوریتم Tiny-YOLO بیشترین عملکرد را در تشخیص شیء دارد.

در این معادله، TP زمانی است که شیء در تصویر باشد و در پایگاه داده نیز وجود داشته باشد. FN در حالتی رخ می‌دهد که الگوریتم شیئی را در تصویر پیدا نکند؛ در صورتی که شیء در داده‌های زمینی وجود دارد. FP نیز متعلق به زمانی است که شیئی در تصویر شناسایی شده اما در پایگاه داده، چنین شیئی وجود نداشته باشد. نتایج این چهار معیار در جدول ۲ آمده است. روش Tiny-YOLO می‌تواند، در ۹۹/۱٪ موارد، شیء را در تصویر تشخیص دهد که در حال حاضر، بهترین دقت گزارش شده در کارهای اخیر است. از طرفی، روش طبقه‌بندی‌کننده آبخاری نیز دقت ۹۶/۵٪ را دارد.

یکی از مهم‌ترین شاخص‌ها به منظور ارزیابی الگوریتم‌ها، در حوزه پهپادها، سرعت اجرای برنامه‌ها روی سخت‌افزارهای آنهاست. با توجه به ساختار پرنده Parrot AR Drone 2.0، تمامی محاسبات شامل تشخیص محل فرود روی پردازنده زمینی انجام می‌شود. کل زمان لازم برای تشخیص و اجرا در پردازنده زمان ارسال تصویر، فرایند تشخیص و ارسال دوباره دستورها به پرنده است. زمان ارسال تصاویر از دوربین پایین در حدود شصت فریم بر ثانیه است که برای کاربردهای آبی، بسیار مناسب است؛ بنابراین کل زمان محاسباتی

جدول ۲. مقایسه روش‌ها از لحاظ دقت

روش	معیار	FNR (%)	PPV (%)	TPr (%)	FDr (%)
Tiny-YOLO		۰/۸	۱۰۰	۹۹/۱	۰/۰۹
طبقه‌بندی‌کننده آبخاری		۲/۴۷	۹۹	۹۷/۵	۱/۰۲

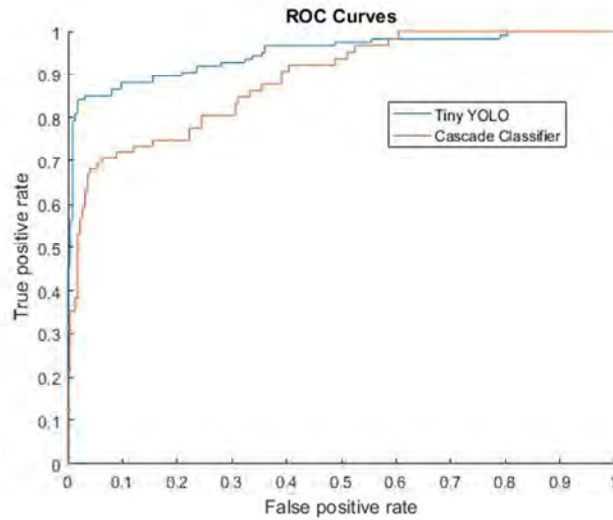
جدول ۳. مقایسه روش‌ها از لحاظ سرعت

روش	معیار	سرعت اجرا (میلی‌ثانیه)
Tiny-YOLO		۱۲۰ (GPU)
طبقه‌بندی‌کننده آبخاری		۲۵ (CPU)

جدول ۴. مقایسه روش‌ها از لحاظ شاخص AUC

روش	معیار	شاخص AUC
Tiny-YOLO		۰/۹۸۱
طبقه‌بندی‌کننده آبخاری		۰/۹۶۵

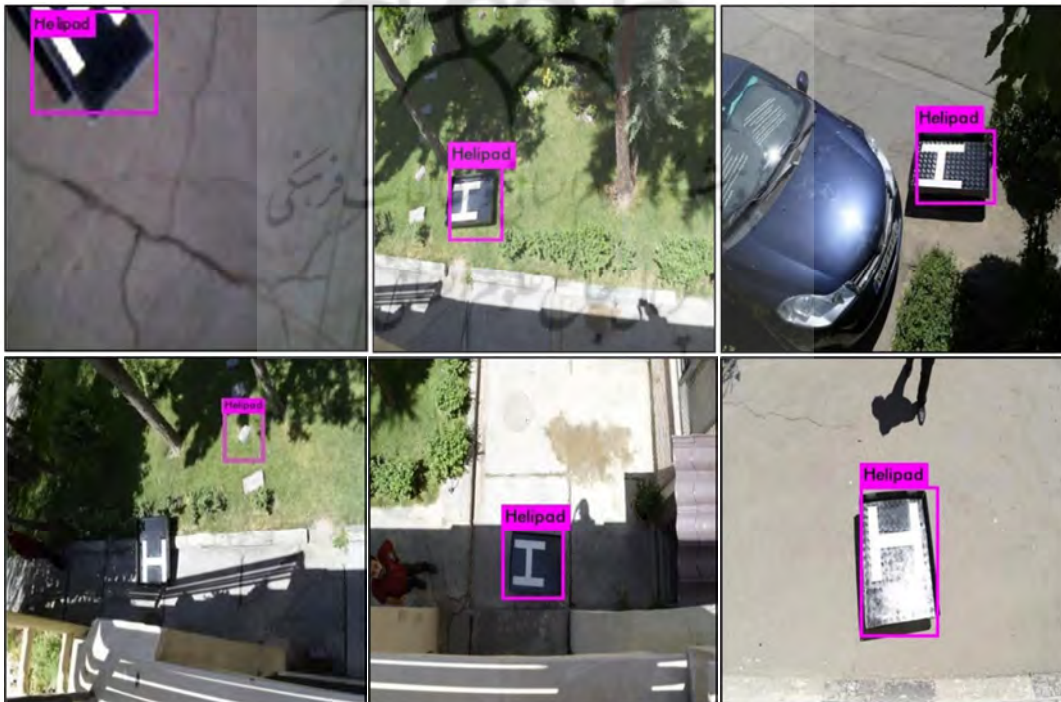
مقایسه روش‌های یادگیری عمیق و طبقه‌بندی کننده آشناری در ...



شکل ۸. نمودار ROC برای الگوریتم‌های متفاوت

یادگیری عمیق تا حد بسیاری کاهش یافته است. البته در این جا، با به‌کارگیری تکنیک‌های پس‌پردازش، می‌توان این دقت را بهبود داد. این کار با محاسبه تعداد نواحی سیاه برای هر مستطیل پیش‌بینی شده و انتخاب حد آستانه‌ای مناسب انجام شد.

در شکل ۹، نمونه‌هایی از نتیجه فرایند تشخیص نشان داده شده است. پس از انجام شدن فرایند و محاسبه موقعیت نسبی، فرمان لازم به‌منظور فرود برای پرنده فرستاده خواهد شد. از ضعف‌های اساسی الگوریتم‌های معمول، مانند طبقه‌بندی کننده آشناری، میزان اشتباه زیاد آنهاست. این ضعف در الگوریتم‌های مبتنی بر

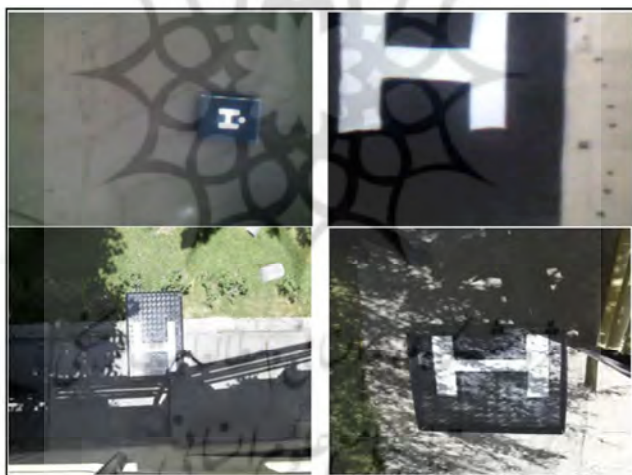


شکل ۹. نتایج فرایند تشخیص در روش‌های متفاوت. سطر اول تشخیص با استفاده از Tiny-YOLO و سطر دوم روش طبقه‌بندی کننده آشناری است

#### ۷-۴- مشکلات و محدودیت‌ها

از دیگر مشکلات در این زمینه، پنهان‌شدگی است. با توجه به ساختار پهپادها و کنترلگر آنها، این پرنده‌ها امکان جابه‌جایی در هر لحظه را دارند و در واقع، ثابت‌ماندن پهپادهایی مانند AR Drone 2.0 در فضا تقریباً بسیار دشوار است. با توجه به جابه‌جایی‌های افقی این پرنده لزوم شناسایی محل فرود، هنگامی که بخشی از محل فرود دیده خواهد شد، ضروری است. از دیگر سو نیز، زاویه دید دوربین زیر پرنده AR Drone 2.0 دشواری این کار را دوچندان می‌کند. این مسئله یکی از چالش‌های اصلی، هم برای روش پیشنهادی و هم برای روش‌های فعلی است و از محدودیت‌های جدی در فرود به‌شمار می‌آید. شکل ۱۱ این چالش را در ارتفاع اندک نشان می‌دهد.

همان‌طور که پیش‌تر نیز بیان شد، مقیاس یکی از مهم‌ترین چالش‌های مطرح‌شده در تشخیص محل فرود است. این نکته زمانی اهمیت می‌یابد که پرنده، حین فرود، با ابعاد گوناگون و متفاوت شکل «H» مواجه شود (شکل ۱۰). مطابق شکل ۱۰، تشخیص شکل محل فرود در ارتفاعات گوناگون مهم‌ترین چالش الگوریتم‌های فرود اتوماتیک شمرده می‌شود. استفاده از ویژگی‌های سلسله‌مراتبی (به‌کارگیری ویژگی‌های استخراجی در همه لایه‌ها به‌منظور تصمیم‌گیری) در روش Tiny-YOLO می‌تواند این نقص را تا حد بسیاری جبران کند؛ درحالی‌که مقیاس از ضعف‌های اساسی روش‌های مبتنی بر طبقه‌بندی‌کننده آبخاری است (De Oliveira & Wehrmeister, 2016).



شکل ۱۰. مقایسه چالش ارتفاع به‌منظور تشخیص محل فرود



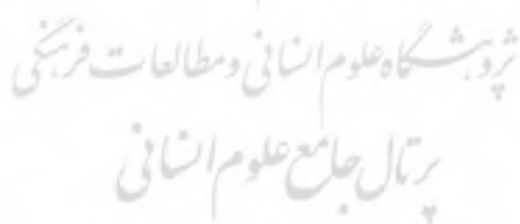
شکل ۱۱. چالش پنهان‌شدگی

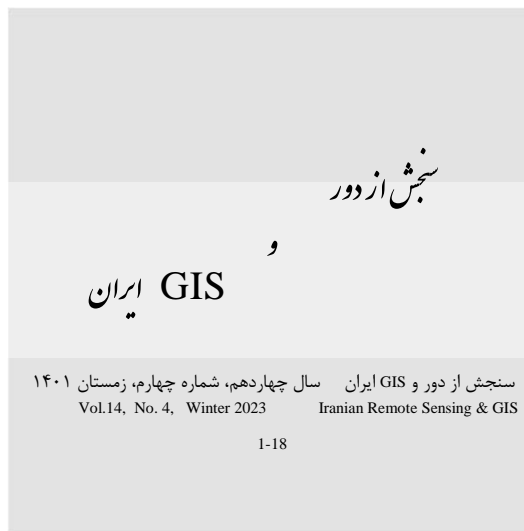
- ۵- بحث و نتیجه‌گیری**
- در این مقاله، سامانه‌ای آنی مبتنی بر تصویر، براساس شبکه‌های عصبی کانولوشنی، به منظور فرود پرنده‌های بدون سرنشین مطرح شد. شبکه‌های کانولوشنی عمیق معمولاً بار محاسبات زیادی دارند اما، طبق آنچه در این پژوهش بیان شد، می‌توان از این شبکه‌ها برای تشخیص شیء در تصاویر پرنده‌های بدون سرنشین، در مسائل دنیای واقعی، استفاده کرد. روش بیان‌شده در این مقاله با سایر روش‌های موجود، از نظر سرعت و دقت، مقایسه شد و نتیجه آن بود که دقت این شیوه بسیار مناسب است و در عین حال، از نظر سرعت نیز، می‌توان آن را با روش‌های مورد اشاره مقایسه کرد. طی نتایج، مشاهده شد که شبکه‌های کانولوشنی تا حد بسیار زیادی در برابر نور، مقیاس، پنهان‌شدگی پایدارند؛ بنابراین می‌توان از آنها به صورت الگوریتم‌هایی کارآمد در حوزه تشخیص اشیاء، در مورد پرنده‌های بدون سرنشین، استفاده کرد. در گام بعدی تحقیق، قصد داریم شبکه‌های کانولوشنی را برای تشخیص اشیاء در حوزه پرنده‌های بدون سرنشین، روی سخت‌افزارهای سبک مانند Raspberry-pi و با هدف کاربردهای واقعی، مانند تشخیص بازماندگان حوادث، به کار بگیریم.
- ۶- منابع**
- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S. et al., 2016, **Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems**, arXiv preprint arXiv: 1603.04467.
- Chen, Y. & Liu, H.- L., 2016, **Overview of Landmarks for Autonomous, Vision-Based Landing of Unmanned Helicopters**, IEEE Aerospace and Electronic Systems Magazine, 31(5), PP. 14-27.
- Chi, H., Yu, C. & Liu, L., 2013, **Vision-Based Attitude Estimation and Control for Unmanned Helicopter Landing**, Control Conference (CCC), 32nd Chinese.
- De Oliveira, D.C., & Wehrmeister, M.A., 2016, **Towards Real-Time People Recognition on Aerial Imagery Using Convolutional**
- Neural Networks, , 19th International Symposium on Real-Time Distributed Computing (ISORC), 17-20 May.
- De Oliveira, C.S., Anvar, A., Silva, M.C., Neto, A. & Mozelli, L., 2015, **Comparison of Cascade Classifiers for Automatic Landing Pad Detection in Digital Images**.
- Fan, Y., Haiqing, S. & Hong, W., 2008, **A Vision-Based Algorithm for Landing Unmanned Aerial Vehicles**, International Conference on Computer Science and Software Engineering, 12-14 Dec.
- Fucen, Z., Haiqing, S. & Hong, W., 2009, **The Object Recognition and Adaptive Threshold Selection in the Vision System for Landing an Unmanned Aerial Vehicle**, 2009 International Conference on Information and Automation, 22-24 June.
- Gautam, A., Sujit, P.B. & Saripalli, S., 2014, **A Survey of Autonomous Landing Techniques for UAVs**, 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 27-30 May.
- Girshick, R., 2015, **Fast R-CNN**, 2015 IEEE International Conference on Computer Vision (ICCV), 7-13 Dec.
- Girshick, R., Donahue, J., Darrell, T. & Malik, J., 2014, **Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation**, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 23-28 June.
- Guennouni, S., Ahaitouf, A. & Mansouri, A., 2015, **A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms**, Modelling and Simulation in Engineering, 2015 (ID 948960).
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. & Lew, M.S., 2016, **Deep Learning for Visual Understanding: A Review**, Neurocomputing, 187(2016), PP. 27-48.
- He, K., Zhang, X., Ren, S. & Sun, J., 2015, **Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification**, Proceedings of the IEEE International Conference on Computer Vision.

- Henriques, J.F., Caseiro, R., Martins, P. & Batista, J., 2015, **High-Speed Tracking with Kernelized Correlation Filters**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(3), PP. 583-596.
- IOFFE, S. & Szegedy, C., 2015, **Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift**, Proceedings of the 32nd International Conference on International Conference on Machine Learning, Vol, 37 July 2015, PP. 448-456.
- Kong, W., Zhou, D., Zhang, D. & Zhang, J., 2014, **Vision-Based Autonomous Landing System for Unmanned Aerial Vehicle: A Survey**, 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 29 December.
- Konoplich, G.V., Putin, E.O. & Filchenkovgeniy, A.A., 2016, **Application of Deep Learning to the Problem of Vehicle Detection in Uav Images**, 2016 XIX IEEE International Conference on Soft Computing and Measurements (SCM), 25-27 May.
- Lee, J., Wang, J., Crandall, D., Šabanović, S. & Fox, G., 2017, **Real-Time Object Detection for Unmanned Aerial Vehicles Based on Cloud-Based Convolutional Neural Networks**, First IEEE International Conference on Robotic Computing (IRC) Taichung, Taiwan, 10-12 April. Lin, min, qiang chen, and shuicheng yan. "Network in network." arxiv preprint arxiv: 1312.4400 (2013).
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A.C., 2016, **SSD: Single Shot Multibox Detector**, European Conference on Computer Vision, Springer, Cham.
- Miller, A., Shah, M. & Harper, D., 2008, **Landing a UAV on a Runway Using Image Registration**, 2008 IEEE International Conference on Robotics and Automation (ICRA), 19-23 May.
- Pan, X., Ma, De-q., Jin, Li-l. & Jiang, Z-s., 2008, **Vision-Based Approach Angle and Height Estimation for UAV Landing**, 2008 Congress on Image and Signal Processing, 27-30 May.
- Redmon, J. & Farhadi, A., 2016, **Yolo9000: Better, Faster, Stronger**, Conference: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016, **You Only Look Once: Unified, Real-Time Object Detection**, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June.
- Ren, S., He, K., Girshick, R. & Sun, J., 2015, **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**, Advances in Neural Information Processing Systems 28 (NIPS 2015).
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. & Fei-Fei, L., 2015, **Imagenet Large Scale Visual Recognition Challenge**, International Journal of Computer Vision, 115(3), PP. 211-252.
- Saripalli, S., Montgomery, J.F. & Sukhatme, G.S., 2003, **Visually Guided Landing of an Unmanned Aerial Vehicle**, IEEE transactions on Robotics and Automation, 19(3), PP. 371-380.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. & LeCun, Y., 2013, **Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks**, 2nd International Conference on Learning Representations, ICLR 2014 - Banff, Canada, 14-16 Apr.
- Han, S., Shen, W. & Liu, Z., 2015, **Deep Drone Object Detection and Tracking for Smart Drones on Embedded System**, Stanford University.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguel, D., Erhan, D., Vanhoucke, V. & Rabinovich, A., 2015, **Going Deeper with Convolutions**, proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2015.



- Tsai, A.C., Gibbens, P.W. & Stone, R.H., 2006, **Terminal Phase Vision-Based Target Recognition and 3d Pose Estimation for a Tail-Sitter, Vertical Takeoff and Landing Unmanned Air Vehicle**, Pacific-Rim Symposium on Image and Video Technology, Springer, Berlin, Heidelberg, 2006.
- Viola, P. & Jones, M., 2001, **Rapid Object Detection Using a Boosted Cascade of Simple Features**, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001.
- Verbandt, M., Theys, B. & De Schutter, J., 2014, **Robust Marker-Tracking System for Vision-Based Autonomous Landing of Vtol Uavs**, IMAV 2014: International Micro Air Vehicle Conference and Competition 2014.
- Wang, R., Zhang, G-J. & Yan, P., 2007, **A Hierarchical Method Based on Optical Flow for Planar 3d Motion Estimation**, Optical Technique, 1(2007), P. 027.





## Comparison of Deep Learning and Cascade Classifiers Approaches for Object Detection in Autonomous UAV Landing

Moradi B.\*<sup>1</sup>, Mehraban A.A.<sup>2</sup>, Mohammadi M.<sup>2</sup>

1. M.Sc. of Photogrammetry, Faculty of Geodesy and Geomatics Engineering, K.N. Toosi University of Technology
2. Ph.D., Faculty of Aerospace, Ferdowsi University, Mashhad

### Abstract

Autonomous landing is a key challenging in the domain of UAV navigation systems. Developing an autonomous landing system requires a precise estimation of the UAV pose relative to landing marker, particularly in vision systems this involves precise Helipad recognition. It seems that traditional approaches including cascade classifiers, image matching and segmentation techniques to have major challenges in different weather conditions and scales. On the other hand, convolutional neural networks (CNNs) have been introduced as a powerful tool in the visual recognition systems in the recent years but the high computational cost of this techniques, limited their performance in the low cost and light weight UAVs. The aim of this paper is to compare the convolutional neural networks and cascade classifier for helipad detection. The results show that CNNs are invariant under translation, rotation, scaling and occlusion. The detection accuracy of this method is 99.1 % which is 3 % more than cascade classifier while its running time is suitable for real time UAV applications.

**Keywords:** Deep learning, Autonomous landing, UAV.

\* Correspondence Address: Faculty of Geomatics Engineering, K.N. Toosi University of Technology, No. 1346, Vali-Asr Street, Mirdamad Cross, Tehran, Postal Code: 1996715433. Tel: (+98 21) 88770218, Fax: (+9821)88786213  
Email: B.msurveyy@gmail.com