

## طراحی پهپاد مسیریاب هوشمند با استفاده از یادگیری عمیق و منطق فازی

فریبرز مولایی، عبدالمجید موسوی و محمد باقر دولتشاهی\*

گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه لرستان، خرم آباد، ایران

تاریخ پذیرش: ۱۴۰۳/۰۶/۲۱

تاریخ دریافت: ۱۴۰۳/۰۲/۱۰

نوع مقاله: علمی-پژوهشی

چکیده. پهپادها ربات‌های پرنده‌ای هستند که با ساختارهای مختلف قابلیت پرواز بدون سرنشین را دارند و می‌توانند تا حدی از هوشمندی برای تصمیم‌گیری مناسب در موقعیت‌های جدید بهره‌مند شوند. همچنین، با پیشرفت هوش مصنوعی در سال‌های اخیر، کاربرد آن در اکثر شاخه‌های دیگر قابل مشاهده است. این مقاله یک پهپاد هوشمند را ارائه می‌دهد که قادر به مانور صحیح هنگام پرواز بر روی جاده‌ها است. به این منظور از یک شبکه یادگیری عمیق استفاده شده که برای آموزش آن، مجموعه داده بزرگی از تصاویر طبیعت و جاده‌های شهری جمع‌آوری شده است و از منطق فازی برای به دست آوردن بهترین زاویه و سرعتی که پهپاد را در مسیر نگه می‌دارد، استفاده شده است. سیستم منطق فازی به عنوان یک لایه واسط بین جزء یادگیری عمیق و سیستم کنترل پرواز پهپاد عمل می‌کند. این سیستم بر اساس داده‌های سنسور بلادرنگ و خروجی شبکه یادگیری عمیق، مسیر پرواز و کنترل سرعت را اصلاح می‌کند. دقت شبکه یادگیری عمیق استفاده شده در این مقاله نسبت به دقت شبکه‌های یادگیری عمیق مورد استفاده در موارد مشابه بهبود یافته است. (ادامه دارد)

2010 Mathematics Subject Classification. 68T27; 03B52

\* Corresponding author

E-mails: mollaie.fa@fe.lu.ac.ir, mousavi.m@lu.ac.ir and dowlatshahi.mb@lu.ac.ir.

عبارات و کلمات کلیدی. وسیله نقلیه بدون سرنشین، یادگیری عمیق، منطق فازی، پرواز خودکار، پهپاد مسیر یاب.

هدف این است که با گسترش مجموعه داده ها و شبکه نسبت به کارهای قبلی، دقت یادگیری را بهبود بخشیده و با استفاده از منطق فازی پیچیدگی را کاهش دهیم. سیستم طراحی شده مذکور، بر روی یک هگزاکوپتر اجرا گردیده و در جاده های خارج از شهر و در ارتفاع پایین مورد بررسی قرار گرفت. این دقت برابر ۱۸.۹۳ درصد است که این نتیجه حاکی از عملکرد قابل قبول آن بود. با این وجود می توان جهت افزایش دقت و کمک به تصمیم گیری بهتر منطق فازی از سنسورهای فاصله سنج در اطراف پهپاد و یا از دوربین های استریو بر روی پهپاد استفاده کرد.

## ۱. سرآغاز

وسایل نقلیه هوایی بدون سرنشین<sup>۱</sup>، که به طور عام به عنوان کوادکوپتر نیز شناخته می شوند، به سرعت در حال تبدیل شدن به یک ابزار رایج در دنیای ما هستند. این ربات ها، از نسخه های کوچک و سرگرم کننده برای افراد عادی گرفته تا ماشین های تجاری بزرگ برای کاربردهای صنعتی، در حال دگرگونی صنایع مختلف هستند [۲، ۶، ۲۲]. یکی از حوزه های کلیدی توسعه در رباتیک، حرکت خودکار است، جایی که ربات ها بدون کنترل مستقیم انسان در محیط اطراف خود حرکت می کنند. این فناوری به طور فعال توسط بخش های خودروسازی و نظامی دنبال می شود و بر استفاده از زیرساخت های جاده ای و حسگرهای متعدد بر روی خودروها برای سفر ایمن تمرکز دارد. با این حال، هنگام تلاش برای اعمال این روش ها در محیط های طبیعی، یک چالش بزرگ به وجود می آید. برخلاف جاده های کاملاً مشخص با الگوهای واضح، مسیرهای جنگلی آسفالت نشده یا پوشش گیاهی انبوه موانع مهمی را برای روبات هایی که صرفاً بر تشخیص مسیر تکیه می کنند، ایجاد می کند [۴، ۷، ۸]. محیط های طبیعی ذاتاً در مقایسه با سازه های ساخته شده توسط انسان غیرقابل پیش بینی تر و پیچیده تر هستند و چالش مهمی را برای روبات های هوایی مانند پهپادها ایجاد می کنند [۲۵]. عواملی مانند زمین ناهموار، شرایط آب و هوایی غیرقابل پیش بینی، و نشانه های بصری مبهم می توانند به طور قابل توجهی مانع از توانایی پهپاد در جهت یابی مستقل شوند. با وجود این مشکلات، پیشرفت در هوش مصنوعی راه حل های امیدوارکننده ای را ارائه می دهد. این مقاله روش جدیدی را پیشنهاد می کند که از قدرت یادگیری عمیق استفاده می کند. الگوریتم های یادگیری عمیق در استخراج الگوهای معنی دار از داده های پیچیده [۹، ۱۷، ۱۹، ۲۰] برتری دارند و آنها را قادر می سازد تا محیط اطراف خود را تفسیر و درک کنند [۵، ۲۱، ۲۳، ۲۸، ۲۹]. هدف از مطالعه حاضر، آموزش یک پهپاد برای پیدا کردن مسیر و حرکت در محیط های طبیعی با

<sup>۱</sup>UAV

استفاده از تکنیک‌های یادگیری عمیق است. همچنین در این مقاله از منطق فازی برای افزایش قابلیت‌های تصمیم‌گیری پهپاد استفاده می‌شود. منطق فازی در مدیریت داده‌های نادقیق برتری دارد و به پهپاد اجازه می‌دهد تا بر اساس عوامل مختلف، از جمله خروجی سیستم یادگیری عمیق، سرعت، شتاب و... انتخاب‌های آگاهانه‌تری داشته باشد. این رویکرد ترکیبی پهپاد را قادر می‌سازد تا سرعت و مسیر پرواز خود را در زمان واقعی بر اساس ماهیت پویا و غیرقابل پیش بینی محیط‌های طبیعی تنظیم کند [۱۴، ۱۸]. در اصل، این مقاله رویکرد جدیدی را بررسی می‌کند که قابلیت‌های استخراج ویژگی یادگیری عمیق را با قدرت تصمیم‌گیری در زمان اجرای منطق فازی ادغام می‌کند. ترکیب یادگیری عمیق و منطق فازی می‌تواند موجب غلبه بر چالش‌های ناوبری هواپیماهای بدون سرنشین در محیط‌های طبیعی باشد و راه را برای طیف وسیع‌تری از کاربردها برای پهپادهای خودران هموار کند [۱۱].

## ۲. پیشینه و کارهای مرتبط

یادگیری عمیق به عنوان ابزاری قدرتمند برای استخراج ویژگی‌ها و بهبود کنترل پهپاد در سناریوهای مختلف پرواز ظاهر شده است [۱۵، ۲۶، ۲۷]. در اینجا نگاهی دقیق‌تر به نحوه متحول کردن ناوبری پهپاد توسط یادگیری عمیق خواهیم انداخت.

ساختارهای یادگیری عمیق، با توانایی خود در یادگیری الگوهای پیچیده از داده‌ها [۱۳، ۲۴]، در استخراج ویژگی‌های حیاتی از ورودی‌های حسگر مانند دوربین و لیدار برتری دارند. این امر به پهپاد اجازه می‌دهد تا به طور موثر محیط اطراف خود را درک کند، موانع، نقاط دیدنی و مسیرهای پرواز را با دقت بالا شناسایی کند. تحقیقات قبلی بر این مزیت تاکید کرده و اثربخشی یادگیری عمیق را در کارهایی مانند مثال‌های زیر نشان داده است.

دو همکاران با موفقیت از یادگیری عمیق برای ایجاد مسیرهای بصری برای کوادروتورهای خودمختار در محیط‌های داخلی استفاده کردند. با تجزیه و تحلیل یک نقشه بصری به عنوان یک نمودار از تصاویر مرتبط، سیستم می‌تواند پهپاد را در فضای داخلی هدایت کند [۱۰]. در دنیای پرهیجان مسابقات سرعت پهپادها، جونگ و همکاران قدرت یادگیری عمیق را برای تشخیص و عبور دقیق از دروازه‌ها نشان دادند. سیستم آنها از یک neural convolutional network برای تخمین دقیق مرکز دروازه‌ها استفاده می‌کرد و به پهپاد اجازه می‌داد تا به طور موثر و بدون برخورد از آنها عبور کند. تخمین دقیق وضعیت برای یک پرواز ایمن و پایدار پهپاد ضروری است. رویکردهای یادگیری عمیق ثابت کرده‌اند که در مقابله با این چالش مهارت دارند [۱۵]. آل شارمن و همکاران یک چارچوب یادگیری عمیق را توسعه دادند که عملکرد

ثبات پهپاد را به طور قابل توجهی بهبود بخشید و در نتیجه پایداری کلی پرواز پهپاد را افزایش داد [۳].

یادگیری عمیق نه تنها به اجتناب از موانع و برنامه ریزی مسیر کمک می کند، بلکه به کنترل خودکار پهپاد نیز کمک می کند. گوتسی و همکاران یک رویکرد مبتنی بر شبکه عصبی را برای کنترل یک کوادروتور برای ناوبری مستقل در مناطق جنگلی پیشنهاد کردند. این امر پتانسیل یادگیری عمیق را برای فعال کردن ماموریت‌های پیچیده پهپاد در دنیای واقعی بدون دخالت گسترده انسان برجسته می کند [۱۲]. علاوه بر این، یادگیری عمیق را می توان برای تقلید تصمیمات خلبانان ماهر پهپاد به کار گرفت. کی کیم و چن از یک مدل یادگیری عمیق برای یادگیری استراتژی‌های کنترل بر اساس اقدامات خلبانان متخصص در فضاهای راهرو محصور استفاده کردند. این رویکردی را برای یادگیری پهپادها باز می کند تا در سناریوهای مختلف پرواز آموزش داده و سازگار شوند، حتی مانورهایی را که نیاز به تخصص انسانی دارند تکرار کنند [۱۶].

منطق فازی برای مسیریابی ربات هایی که در محیط های ناشناخته عمل می کنند می تواند مفید باشد پژوهشی که سمیه مغازی و همکاران با به کارگیری رباتی در مسیر های پویا برای کمینه سازی زمان رسیدن از مبدا به مقصد انجام دادند [۱].

به طور خلاصه، یادگیری عمیق به یک عامل تغییر دهنده بازی در ناوبری پهپاد تبدیل شده است. یادگیری عمیق با تسهیل استخراج ویژگی، امکان تخمین دقیق و حتی خودکار کردن کنترل، راه را برای آینده ای هموار می کند که در آن پهپادها بتوانند به صورت خودکار، با دقت و کارایی بیشتر در محیط های متنوع عمل کنند.

### ۳. روش کار

قبل از شرح جزئیات این رویکرد، نگاهی به سخت افزار و نرم افزار راه اندازی شده در این تحقیق می اندازیم:

هگزاکوپتر دست ساز: از این ربات برای تست و ارزیابی روش کار استفاده شده و تمام تجهیزات روی آن نصب شده است. (شکل ۱)

Pixhawk Flight Controller : این کنترل کننده پرواز متن باز محبوب، پایداری پرواز را فراهم می کند و قابلیت اتصال به سنسورهای مختلف را دارد.

GPS و سنسور Lidarlight : این سنسورها اطلاعات حیاتی را برای کنترل کننده پرواز جهت حفظ موقعیت و ارتفاع ارائه می دهند.



شکل ۱: پهپاد استفاده شده در پژوهش

کامپیوتر تک‌بردی XU۴ (واحد پردازش): کامپیوتر قرار داده شده بر روی پهپاد، داده‌های تصویری گرفته شده توسط دوربین را پردازش می‌کند و با مدل یادگیری عمیق و سیستم منطق فازی فرمانهای مناسب برای فلات کنترل را ارسال می‌نماید.

دوربین: OV۲۷۱۰ این دوربین تصاویر محیط را برای تجزیه و تحلیل توسط مدل یادگیری عمیق ثبت می‌کند.

Pixhawk Flight Controller که روی این پهپاد نصب شده، از رام PX۴ استفاده می‌کند و از طریق پروتکل Mavlink به رایانه تک‌بردی odroid-XU۴ متصل می‌شود. سیستم عامل اوبونتو ۱۶.۰۴، بسته نرم‌افزاری FlytOS و ROS Kinetic روی odroid-XU۴ نصب شده‌اند تا دستورات لازم را به فلات کنترل ارسال کنند. برای عملیات یادگیری از کتابخانه keras به عنوان واسط TensorFlow استفاده شده است.

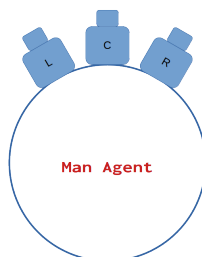
تمام نرم‌افزارها و کتابخانه‌هایی که در odroid-xu۴ استفاده می‌شوند، برای فاز آموزش روی یک رایانه‌ی قدرتمندتر نیز نصب شده‌اند.

### ۱.۳. جمع‌آوری و توسعه مجموعه داده.<sup>۱</sup>

در این فاز، حجم زیادی از داده‌ها شامل تصاویر گرفته شده از محیط برای ورود به مدل یادگیری عمیق جمع‌آوری می‌شود. این مجموعه داده شامل عکس‌هایی است که از ۳ دوربین تعبیه شده بر روی سر عامل انسانی به دست آمده است این دوربین‌ها در زوایای خاص بر روی سر عامل قرار داده شده‌اند و عامل انسانی با عبور از جاده‌های داخل و یا خارج شهر از محیط

<sup>۱</sup>Extending the dataset

فیلم برداری می کند و سپس فیلم تبدیل به مجموعه ای از عکس ها می شود که با توجه به زاویه دوربین در کلاس مورد نظر قرار داده می شود. (شکل ۲).



شکل ۲: نحوه نصب دوربین ها بر روی سر

عکس های گرفته شده توسط دوربین چپ در پوشه TL ، عکس های دوربین راست در پوشه TR و عکس های دوربین وسط در پوشه TS قرار گرفتند. این عکس های جمع آوری شده به دلیل مشکلات خاصی مانند زوایای نادرست و تاریکی، اصلاح و بهبود یافتند و در نهایت مجموعه ای از ۴۰،۹۸۱ عکس ایجاد شد. سپس عکس های دسته چپ به صورت افقی برگردانده شدند و به دسته راست اضافه شدند. عکس های دسته راست نیز به صورت افقی برگردانده شده و در دسته چپ قرار گرفتند. (شکل ۳)



شکل ۳: گسترش مجموعه داده با کپی کردن معکوس تصاویر کلاس سمت چپ به کلاس سمت راست و بالعکس.

به همین ترتیب، عکس های کلاس وسط به صورت افقی برعکس شده و در همان کلاس قرار گرفت. (شکل ۴)

کل مجموعه داده شامل ۸۱،۹۶۲ عکس می شود. تعداد عکس ها در هر کلاس در زیر فهرست شده است:



شکل ۴: گسترش مجموعه داده با کپی کردن تصاویر معکوس شده‌ی کلاس وسط به خود آن کلاس.

TR:۲۸۸۸۱

TL:۲۸۸۸۱

TS:۲۴۲۰۰

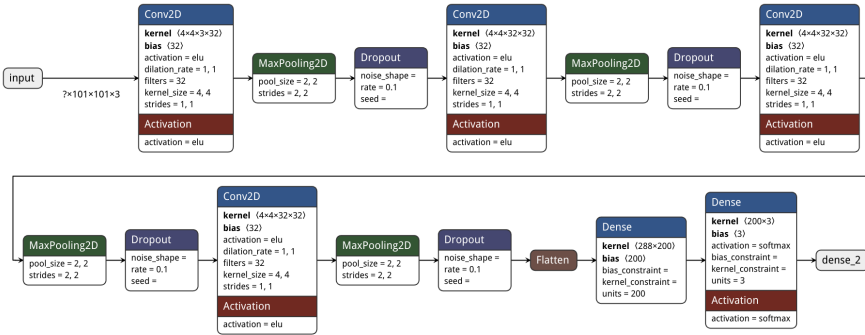
۳۰ درصد عکس‌ها برای تست و ۷۰ درصد آن‌ها برای آموزش استفاده شده است.

### ۲.۳. ساخت مدل شبکه عصبی عمیق.

مدل ساخته شده، یک مدل شبکه عصبی عمیق ۱۶ لایه است که اولین لایه آن لایه ورودی  $۱۶ \times ۱۶ \times ۳$  نود است و برای تزریق تصاویر به شبکه عصبی استفاده می‌شود. این لایه یک Flatten Layer است که در مجموع ۷۶۸ دارد. لایه بعدی یک Convolutional Layer است که از فیلترهای ۴ در ۴ استفاده می‌کند. سومین لایه، لایه Max Polling است که با کاهش ابعاد تصاویر، پیچیدگی شبکه را کاهش می‌دهد. در این لایه از فیلترهای ۲ در ۲ استفاده شده است. علاوه بر این، استفاده از یک لایه Dropout به جلوگیری از Overfitting شبکه کمک می‌کند. لایه‌های انتهایی شامل لایه Flatten، Dense و لایه خروجی است که لایه خروجی دارای سه نورون است که نشان‌دهنده سه کلاس چپ، راست و وسط می‌باشد (شکل ۵). در این مدل به دلیل اینکه خروجی شبکه بازه‌ای بین ۰ تا ۱ است از تابع ELU استفاده شده است که نسبت به توابعی همچون Tanh پیچیدگی کمتری دارد.

### ۳.۳. آموزش مدل.

مدل شبکه عصبی با استفاده از مجموعه داده آماده شده آموزش داده می‌شود. همانطور که ذکر شد، تعداد عکس‌ها در پایگاه داده ۸۱۹۶۲ عکس است که به طور کلی حجم و ابعاد زیادی دارند و باعث بار زیادی در مدل شبکه عصبی می‌شوند، بنابراین ابعاد عکس‌ها به  $۱۰۱ \times ۱۰۱ \times ۳$  تغییر داده می‌شود که در نتیجه باعث کاهش چشمگیر تعداد نورون‌ها در لایه اول می‌شود. همچنین شبکه ساخته شده با مجموعه داده بزرگتر که با روشهای افزایش داده



شکل ۵: مدل ارائه شده در این مقاله.

۱ گسترش پیدا کرده ارزیابی گردیده که در این روش از چرخش، بزرگنمایی شیفت عکس ها به صورت افقی یا عمودی و تغییر مقیاس استفاده شده است (شکل ۶) که در نهایت اندازه مجموعه داده به ۳۲۲۵۶۰ عکس رسیده که ۲۵۸۰۴۸ عکس برای آموزش و ۶۴۵۱۲ عکس برای تست در نظر گرفته شده است [۶]. رایانه‌ای که در فاز آموزش استفاده می‌شود به کارت



شکل ۶: گسترش مجموعه داده با روشهای افزایش داده.

<sup>1</sup>Data Augmentation

گرافیک Nvidia و ۱۲ گیگابایت رم مجهز شده است که برای انجام عملیات یادگیری توسط پردازنده گرافیکی به کار می‌رود. با توجه به محدودیت های سخت افزاری که در مرحله یادگیری با آن مواجه هستیم، بارگذاری کل مجموعه داده در حافظه به دلیل اینکه حداکثر حافظه سیستم ۱۲ گیگابایت است در حالی که حجم کل فایل‌ها تقریباً به هفت گیگابایت می‌رسد، غیرممکن است. از آنجایی که برخی از این فایل‌ها برای درج متغیرها و تغییر فرمت‌ها چندین بار فراخوانده می‌شوند، به حداقل ۳۰ گیگابایت رم نیاز دارند. بنابراین، راه‌حل بهتر استفاده از یک تابع تولیدکننده سفارشی برای قرار دادن داده‌ها در رم به صورت قسمت‌های جداگانه و مستقل است. تابع تولید کننده سفارشی روشی برای تغذیه مجموعه داده‌ها به مدل شبکه عصبی در حین آموزش است. این امر به ویژه هنگام کار با مجموعه داده‌های بزرگ که نمی‌توانند در حافظه جا شوند یا زمانی که نیاز به انجام پیش‌پردازش یا افزایش داده‌های پیچیده دارید، می‌تواند مفید باشد. تولید کننده داده سفارشی مورد استفاده در این پژوهش در زیر آورده شده است:

```
#Custom Generator Function
class My_Custom_Generator(keras.utils.Sequence) :
    def __init__(self, file_names, labels, batch_size) :
        self.file_names = file_names
        self.labels = labels
        self.batch_size = batch_size
    def __len__(self) :
        return (np.ceil(len(self.file_names) / float(self.batch_size))).astype(np.int)
    def __getitem__(self, idx) :
        batch_x = self.file_names[idx * self.batch_size : (idx+1) * self.batch_size]
        batch_y = self.labels[idx * self.batch_size : (idx+1) * self.batch_size]
        return np.array([resize(imread('dataset_location'+ str(file_name))), (101, 101, 3))
                        for file_name in batch_x])0.255/, np.array(batch_y)

#import Data
file_names_counter = 0
labels_counter = -1
for subdir, dirs, files in os.walk(train_dir):
    for file in files:
        file_names.append(file)
        labels[file_names_counter, 0] = labels_counter
        file_names_counter = file_names_counter + 1
        labels_counter = labels_counter+1

#categorying the photos
```

```
y_labels_one_hot = to_categorical(labels)
```

```
#shuffling the Data
```

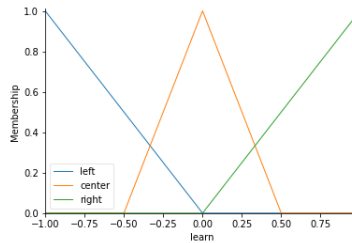
```
filenames_shuffled, label_shuffled = shuffle(filenames, y_labels_one_hot)
```

برای یک مدل شبکه عصبی عمیق به صورت کلی نمی توان یک بهینه ساز خاص را پیشنهاد داد. در حالی که بهینه ساز Adam مزایایی همچون تنظیم خودکار پارامترهای یادگیری، استفاده در شبکه‌های یادگیری مختلف و یا داشتن برخی از مزایای بهینه‌سازهای RMSprop و Momentum، اما بهینه سازی همچون SGD از سرعت بیشتری برخوردار است و نسبت به نوسان حساسیت کمتری دارد. این مدل با بهینه‌سازهای Adam و SGD مورد آزمایش قرار گرفته است که با توجه به نتایج حاصله در نهایت، Adam به عنوان بهینه‌ساز اصلی انتخاب شد. تابع هزینه استفاده شده برای شبکه Categorical Crossentropy است. Categorical Crossentropy یک تابع هزینه است که برای مسائل طبقه‌بندی چند کلاسه استفاده می‌شود. این تابع اختلاف بین توزیع احتمال پیش‌بینی شده توسط مدل و توزیع احتمال واقعی را اندازه‌گیری می‌کند و مزایایی از جمله سادگی و کارایی تفسیر پذیری دارد که گزینه مناسبی برای مسائل چند کلاسه از جمله این پژوهش مناسب می‌سازد.

### ۴.۳. حرکت پهپاد.

مدل کامپایل شده در فاز قبلی به عنوان یک فایل حاوی نتیجه فاز یادگیری ذخیره شده و سپس به رایانه تک برد روی پهپاد منتقل می‌شود. ارتفاع پهپاد همان ارتفاع عامل انسانی است و این ارتفاع ۸۰.۱ می‌باشد. هر بار تصویری توسط دوربین تعبیه شده در پهپاد گرفته می‌شود، دوربین موجود بر روی پهپاد یک دوربین هشت مگاپیکسل است که رزولیشن عکسهای آن به مراتب بزرگتر از عکسهایی است که به شبکه عصبی تزریق شود بنابراین قبل از تزریق این عکسها اندازه آنها به سایز ۱۰۱\*۱۰۱ تغییر داده می‌شود و بدون هیچ تغییر دیگری عکسها برای کلاس بندی به شبکه عصبی وارد می‌شوند. خروجی شبکه عصبی عمیق عددی اعشاری بین صفر و یک است، در صورتی که این عدد مربوط به کلاس چپ باشد میزانی از وابستگی به کلاس وسط را نیز دارد، این عدد به عددی اعشاری بین منفی یک تا صفر تبدیل می‌شود. در صورتی که خروجی شبکه عصبی کلاس وسط باشد عدد تبدیل به عددی بین -۰.۵ تا ۰.۵ می‌شود که در این صورت با توجه به مقدار آن ممکن است میزانی از وابستگی به کلاس چپ یا راست داشته باشد. در صورتی که خروجی شبکه عصبی کلاس راست را نشان دهد خروجی

به عددی بین صفر تا یک تبدیل می‌شود که ممکن است میزان تداخلی با کلاس وسط داشته باشد. این عدد به عنوان متغیر Learn در نظر گرفته می‌شود که متغیرهای فازی آن شامل نام کلاس‌های شبکه DNN است. (شکل ۷)



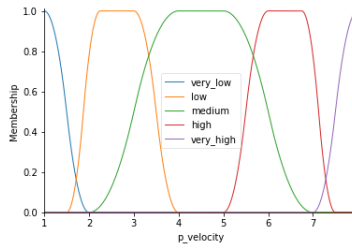
شکل ۷: تابع عضویت learn

در مسیرهای مستقیم، سرعت پهپاد افزایش می‌یابد و بسته به میزان وابستگی تصویر به کلاس، پهپاد با زاویه بسیار کم به سمت چپ یا راست یا به طور مستقیم حرکت می‌کند. در پیچ جاده‌ها، سرعت پهپاد کاهش می‌یابد و با زاویه بیشتری به سمت چپ یا راست حرکت می‌کند. همچنین سرعت ربات نباید از حد خاصی بالاتر یا پایینتر شود و در یک محدوده کنترل شده قرار گیرد، به همین منظور پنج محدوده برای سرعت ربات در نظر گرفته می‌شود که در قسمت‌های که سرعت ربات خیلی زیاد یا خیلی کم است میزان تغییرات (افزایش یا کاهش) سرعت ربات باید خیلی کمتر شود و یا به صفر میل کند. کمترین سرعت ربات در پیچ‌ها تقریباً یک و بیشترین سرعت برابر هشت در نظر گرفته شده است.  $P\_velocity$  سرعت فعلی ربات است که به عنوان ورودی به سیستم فازی داده می‌شود، از تابع عضویت  $\pi$  membership function استفاده شده است تا سرعت در هنگام تغییر وضعیت سریعتر تغییر کند. (شکل ۸)

خروجی سیستم فازی شامل ۲ متغیر Speed\_Adaptation و متغیر Rotate است. (شکل ۹)

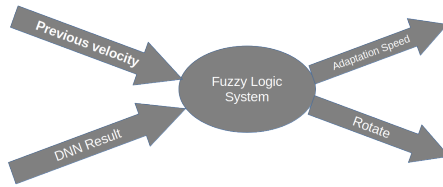
با استفاده از سرعت فعلی ربات و کلاس تشخیص داده شده با شبکه DNN مقداری به عنوان سرعت سازگاری به دست می‌آید که با سرعت فعلی ربات جمع می‌گردد، این متغیر شامل سه محدوده Increase Speed very Low change Decrease Speed است. (شکل ۱۰)

خروجی Rotate فقط از خروجی شبکه DNN استفاده می‌کند و زاویه چرخش جدید را محاسبه می‌کند. (شکل ۱۱)

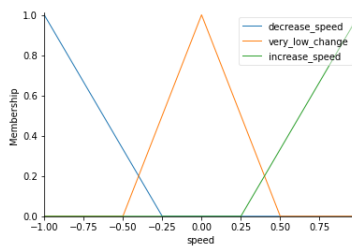


شکل ۸: تابع عضویت p\_velocity

Fuzzy Logic System Output And Input

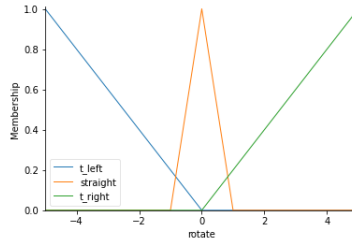


شکل ۹: ورودی و خروجی سیستم فازی.



شکل ۱۰: تابع عضویت Speed Adaptation

پس از تعریف متغیرها و تعریف توابع عضویت قواعد فازی سیستم مشخص می‌شود. (شکل



شکل ۱۱: تابع عضویت rotate

Fuzzy Rules				
RULE	IF		THEN	
RULE	Learn	p_velocity	Adaptation Speed	Rotate
1	Left	Very_Low	very_low_change	-----
2	Left	-----	-----	t_left
3	Left	Low	decrease_speed	-----
4	Left	Medium	decrease_speed	-----
5	Left	Hight	decrease_speed	-----
6	Left	Very_High	decrease_speed	-----
7	Center	Very_Low	increase_speed	-----
8	Center	-----	-----	straight
9	Center	Low	increase_speed	-----
10	Center	Medium	increase_speed	-----
11	Center	Hight	increase_speed	-----
12	Center	Very_High	very_low_change	-----
13	Right	Very_Low	very_low_change	-----
14	Right	-----	-----	t_right
15	Right	Low	decrease_speed	-----
16	Right	Medium	decrease_speed	-----
17	Right	Hight	decrease_speed	-----
18	Right	Very_High	decrease_speed	-----

شکل ۱۲: قواعد فازی

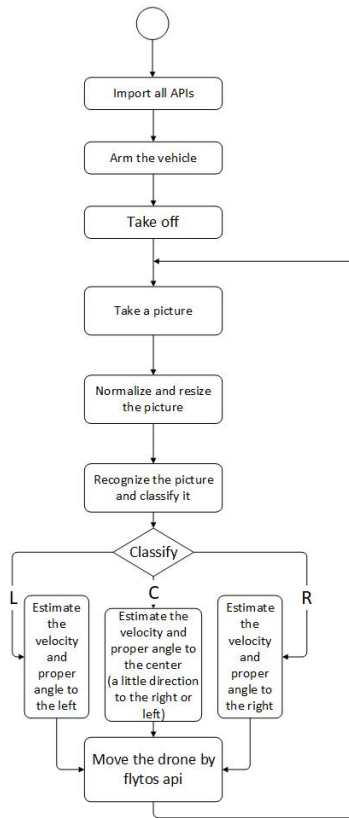
بعد از اینکه سیستم فازی زاویه و سرعت را تعیین کرد، مقادیر به دست آمده برای تولید دستورات مورد نیاز برای کنترل پرواز به فلایت کنترل داده می‌شود. (شکل ۱۳)

#### ۴. نتایج تجربی

##### ۱.۴. فاز آموزش.

در بخش یادگیری عمیق، از دو بهینه‌ساز adam و sgd و دو تابع فعال‌سازی tanh و elu برای آموزش مدل استفاده شده است. ما ابتدا مدل را با بهینه‌ساز SGD و ۸۱۹۶۲ داده آموزش دادیم تا دقت و خطای مدل را اندازه‌گیری کنیم، که در جدول ۱ نشان می‌دهد که overfitting

All Process Flowchart



شکل ۱۳: شمای سیستم.

رخ داده است. بنابراین، برای جلوگیری از overfitting از لایه‌های dropout بعد از لایه‌های maxpooling استفاده کردیم. (جدول ۲)

جدول ۱: نتایج به دست آمده با استفاده از SGD

تعداد داده	خطای ارزیابی	دقت ارزیابی	خطا	دقت	تعداد اپوک
۸۱۹۶۲	۱۷۰۶.۰	۸۲۹۴.۰	۰۰۲.۰	۹۹۸.۰	۹۰

جدول ۲: نتایج به دست آمده با استفاده از SGD و  $\text{DROPOUT}=0.1$

تعداد اپوک	دقت	خطا	دقت ارزیابی	خطای ارزیابی	تعداد داده
۹۰	۹۳۳.۰	۰.۶۷.۰	۸۴۳۲.۰	۱۵۶۸.۰	۸۱۹۶۲

علاوه بر این، مدل با استفاده از بهینه‌ساز Adam و ۸۱۹۶۲ داده آموزش داده شده است. همچنین از لایه‌های Dropout استفاده شده است (جدول ۳).

جدول ۳: نتایج به دست آمده با ۸۱۹۶۲ عکس و استفاده از adam و  $\text{DROPOUT}=0.1$

تعداد اپوک	دقت	خطا	دقت ارزیابی	خطای ارزیابی	تعداد داده
۹۰	۹۳۴.۰	۰.۶۶.۰	۹۳۹۳.۰	۰.۶۰۷.۰	۸۱۹۶۲

با توجه به افزایش اندازه مجموعه داده و محدودیت سخت‌افزار، بارگذاری همه تصاویر به طور همزمان امکان‌پذیر نیست، بنابراین آنها را به صورت دسته‌ای در حافظه قرار می‌دهد، اما باعث افزایش زمان یادگیری می‌شود. در نهایت، حجم مجموعه داده با استفاده از بزرگنمایی به ۳۲۲۵۶۰ افزایش یافته است که دقت یادگیری را به ۲۹۰.۹۴ افزایش می‌دهد. (جدول ۴)

جدول ۴: نتایج به دست آمده با ۳۲۲۵۶۰ عکس و استفاده از adam و  $\text{DROPOUT}=0.1$

تعداد اپوک	دقت	خطا	دقت ارزیابی	خطای ارزیابی	تعداد داده
۹۵	۹۴۲.۰	۰.۵۸.۰	۹۶۲۸.۰	۰.۳۷۲.۰	۳۲۲۵۶۰

به منظور ارزیابی نهایی مدل آموزش دیده، دقت مدل با مجموعه‌ای از تصاویر خارج از مجموعه داده سنجیده شد، دقت بدست آمده ۱۸.۹۳ است. (جدول ۵)

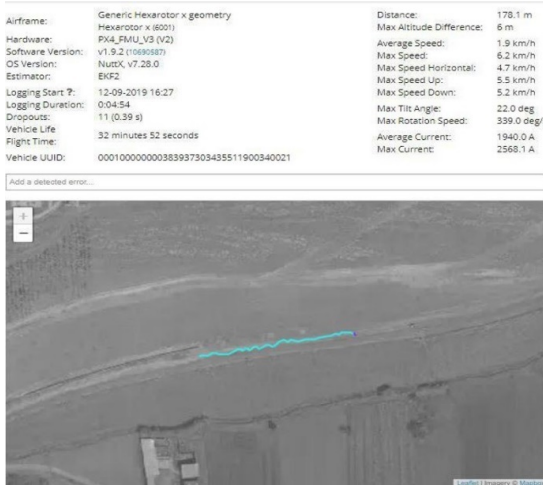
#### ۲.۴. حرکت پهپاد در محیط‌های طبیعی.

ربات طبق شرایط ذکر شده در یک جاده کم رفت و آمد در حومه شهر مورد آزمایش قرار گرفت که از آن در حال پرواز و حرکت فیلمبرداری شد. (شکل ۱۰)

در جدول زیر مقایسه‌ای با برخی از روش‌های کار شده در پژوهش‌های دیگر انجام شده است:

جدول ۵: نتایج به دست آمده برای تست بر روی عکس های خارج از مجموعه داده

موضوع	چپ	وسط	راست	مجموع
تعداد عکس ها	۱۶	۹	۱۹	۴۴
تعداد تشخیص درست	۱۶	۷	۱۸	۴۱
تعداد تشخیص نادرست	۰	۲	۱	۳



شکل ۱۴: مسیر حرکت پهپاد.

جدول ۶: مقایسه روش فعلی با روش های مطرح شده در سایر پژوهش ها

دقت این پژوهش	[۱۲]	Saliency	[۲۶]	
دقت	۱۸.۹۳	۲.۸۵	۳.۵۲	۵.۳۶

تمامی روش های بالا با یک مجموعه داده ارزیابی شده اند. روش Saliency در پردازش تصویر و بینایی ماشین، روشی است که برای شناسایی مهم ترین و جلب توجه ترین نواحی یک تصویر استفاده می شود. این نواحی معمولاً مناطقی هستند که در مقایسه با سایر بخش های تصویر، از نظر بصری برجسته تر بوده و حاوی اطلاعات مهم تری هستند.

## ۵. نتیجه گیری

این مقاله روشی نوین برای ناوبری خودکار پهپاد در محیط‌های طبیعی را پیشنهاد می‌کند که از نقاط قوت یادگیری عمیق و منطق فازی استفاده می‌کند. یادگیری عمیق در استخراج ویژگی‌ها از داده‌های پیچیده برتری دارد و به پهپاد اجازه می‌دهد تا به طور موثر محیط اطراف خود را درک کند. از طرف دیگر، منطق فازی در مدیریت اطلاعات نادرست مهارت دارد و به پهپاد اجازه می‌دهد تا بر اساس عوامل مختلف تصمیمات آگاهانه بگیرد. با ترکیب این تکنیک‌ها، روش پیشنهادی بر چالش‌های ناوبری در محیط‌های طبیعی غلبه می‌کند و راه را برای کاربردهای گسترده تر پهپادهای خودکار هموار می‌سازد.

تحقیقات ارائه شده در این مقاله، دقت نسبتاً مناسبی را برای خودکار سازی این نوع از ربات‌ها ارائه می‌کند. همچنین برخی از زمینه‌های کلیدی برای کاوش بیشتر آورده شده است: استفاده از حسگرهای اضافی علاوه بر دوربین مانند LiDAR یا رادار می‌تواند داده‌های غنی تری را برای مدل یادگیری عمیق فراهم کند که منجر به ناوبری قوی تر در محیط‌های متنوع می‌شود.

توسعه مجموعه داده آموزشی با طیف وسیعی از صحنه‌های طبیعی و شرایط آب و هوایی، توانایی مدل را برای تعمیم به سناریوهای دیده نشده تقویت می‌کند. ادغام الگوریتم‌های پیچیده‌تر با این روش می‌تواند به پهپاد اجازه دهد تا مانورهای پیچیده‌تری انجام داده و موانع را به طور موثری مدیریت کند. توسعه تکنیک‌های یادگیری آنلاین به مدل یادگیری عمیق اجازه می‌دهد تا بر اساس داده‌های پرواز بلادرنگ به طور مداوم عملکرد خود را تطبیق دهد و بهبود بخشد. بررسی الگوریتم‌هایی برای ارتباط چندین پهپاد مجهز به این سیستم می‌تواند آنها را قادر سازد تا به طور مشترک در محیط‌ها حرکت و کاوش کنند.

## مراجع

- [۱] مغاری، فتحعلی، جعفر، براتی. (۲۰۲۱). الگوریتم مسیریابی دو هدفه مبتنی بر منطق فازی ربات‌های متحرک در محیط‌های پویای ناشناخته خلوت. سیستم‌های فازی و کاربردها، شماره ۱، صص ۱ تا ۲۱.
- [2] Ahmed, T., Rahman, T., Roy, B. B., & Uddin, J. (2021). Drone Detection by Neural Network Using GLCM and SURF. *Journal of Information Systems and Telecommunication*, 9(33), 15-24.
- [3] Al-Sharman, M. K., Zweiri, Y., Jaradat, M. A. K., Al-Husari, R., Gan, D., & Seneviratne, L. D. (2019). Deep-learning-based neural network training for state estimation enhancement: Application to attitude estimation. *IEEE Transactions on Instrumentation and Measurement*, 69(1), 24-34.

- [4] Bezas, K., Tsoumanis, G., Angelis, C. T., & Oikonomou, K. (2022). Coverage path planning and point-of-interest detection using autonomous drone swarms. *Sensors*, 22(19), 7551.
- [5] Brownlee, J. (2019). How to configure image data augmentation in Keras. *Machine Learning Mastery*, 5.
- [6] Chamayou, G. (2015). *A Theory of the Drone*. New Press, The.
- [7] Chen, Q., Chen, J., & Huang, W. (2022). Pathfinding method for an indoor drone based on a BIM-semantic model. *Advanced Engineering Informatics*, 53, 101686.
- [8] Diller, J., & Han, Q. (2023). Energy-Aware Drone Path Finding with a Fixed-Trajectory Ground Vehicle.
- [9] Dorrani, Z., Farsi, H., & Mohammadzadeh, S. (2022). Edge detection and identification using deep learning to identify vehicles. *Journal of Information Systems and Telecommunication (JIST)*, 3(39), 201.
- [10] Do, T., Carrillo-Arce, L. C., & Roumeliotis, S. I. (2019). High-speed autonomous quadrotor navigation through visual and inertial paths. *The International Journal of Robotics Research*, 38(4), 486-504.
- [11] ermaugan, U., Yildiz, B., & Salman, F. S. (2022). A learning based algorithm for drone routing. *Computers & Operations Research*, 137, 105524.
- [12] Giusti, A., Guzzi, J., Ciresan, D. C., He, F. L., Rodríguez, J. P., Fontana, F., ... & Gambardella, L. M. (2015). A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2), 661-667.
- [13] Hafez-Kolahi, H., & Kasaei, S. (2019). Information bottleneck and its applications in deep learning. *arXiv preprint arXiv:1904.03743*.
- [14] Jindal, K., & Aron, R. (2022). A hybrid machine learning approach for sentiment analysis of beauty products reviews. *Journal of Information Systems and Telecommunication (JIST)*, 1(37), 1.
- [15] Jung, S., Hwang, S., Shin, H., & Shim, D. H. (2018). Perception, guidance, and navigation for indoor autonomous drone racing using deep learning. *IEEE Robotics and Automation Letters*, 3(3), 2539-2544.
- [16] Kim, D. K., & Chen, T. (2015). Deep neural network for real-time autonomous indoor navigation. *arXiv preprint arXiv:1511.04668*.
- [17] Kulkarni, P. V., Rai, S., Sachdeo, R., & Kale, R. (2023). Deep Learning-based Educational User Profile and User Rating Recommendation System for E-Learning. *Journal of Information Systems and Telecommunication (JIST)*, 3(43), 185.
- [18] Kilic, K. I., & Mostarda, L. (2021). Heuristic drone pathfinding over optimized charging station grid. *IEEE Access*, 9, 164070-164089.

- [19] MJ, C. M. B. (2023). Convolutional Neural Networks for Medical Image Segmentation and Classification: A Review. *Journal of Information Systems and Telecommunication (JIST)*, 4(44), 347.
- [20] Mozaffari, F., Raeesi Vanani, I., Mahmoudian, P., & Sohrabi, B. (2023). Application of Machine Learning in the Telecommunications Industry: Partial Churn Prediction by using a Hybrid Feature Selection Approach. *Journal of Information Systems and Telecommunication (JIST)*, 4(44), 331.
- [21] Ricco, J. (2017). What is max pooling in convolutional neural networks.
- [22] Reséndiz, V. M. A., & Rivas-Araiza, E. A. (2016). System Identification of a Quad-rotor in X Configuration from Experimental Data. *Res. Comput. Sci.*, 118, 77-86.
- [23] Sanchez-Lopez, J. L., Sampedro, C., Cazzato, D., & Voos, H. (2019, June). Deep learning based semantic situation awareness system for multirotor aerial robots using LIDAR. In 2019 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 899-908). IEEE.
- [24] Sandooghdar, A., & Yaghmaee, F. (2022). Deep learning approach for cardiac MRI images. *Journal of Information Systems and Telecommunication (JIST)*, 1(37), 61.
- [25] Schedl, D. C., Kurmi, I., & Bimber, O. (2021). An autonomous drone for search and rescue in forests using airborne optical sectioning. *Science Robotics*, 6(55), eabg1188.
- [26] Santana, P., Correia, L., Mendonça, R., Alves, N., & Barata, J. (2013). Tracking natural trails with swarm-based visual saliency. *Journal of Field Robotics*, 30(1), 64-86.
- [27] Tch, A. (2017). The mostly complete chart of Neural Networks, explained. Online Medium.
- [28] Vo, A. (2018). Deep Learning-Computer Vision and Convolutional Neural Networks,.”
- [29] Yiu, T. (2019). Understanding neural networks. *Towards Data Science*, 2.
- [30] Zadeh, L. A. (1965), Fuzzy sets. *Information and Control*, 8, 338–353.