



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر

گزارش پروژه کارشناسی

هدایت پهپاد با علائم دست مبتنی بر بینایی  
ماشین

نگارش  
سارا تاجرنیا

استاد راهنما

دکتر مهدی جوانمردی

خرداد ۱۴۰۳

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر

گزارش پروژه کارشناسی

هدایت پهپاد با علائم دست مبتنی بر بینایی  
ماشین

نگارش  
سارا تاجرنیا

استاد راهنما

دکتر مهدی جوانمردی

خرداد ۱۴۰۳

# صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تایید و تصویب پایان نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- را قرار دهید.

## نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.



دانشگاه صنعتی امیرکبیر  
(پلی‌تکنیک تهران)

به نام خدا

تاریخ: خرداد ۱۴۰۳

## تعهدنامه اصالت اثر

اینجانب سارا تاجرنیا متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظرارت و راهنمایی استادی دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

سارا تاجرنیا

امضا

این پیان نامه را تقدیم می کنم به محبا ترین همراهان زندگیم، پدر، مادر، برادران  
غزینزم که حضور شان همیشه گرما بخش روح من بوده است.

# سپاس‌گزاری

زندگی دفتری از خاطره هاست، یک نفر در دل شب، یک نفر در دل خاک، یک نفر همدم خوشبختی هاست، یک نفر همسفر سختی هاست، چشم تا باز کنیم، عمرمان می‌گذرد ما همه رهگذریم، آنچه باقیست فقط خوبی‌هاست.

تشکر می‌کنم از تمامی عزیزانی که در تمامی مراحل زندگی همراه من بوده‌اند.  
و همچنین از استاد گرامی جناب آقای دکتر مهدی جوانمردی که در انتخاب و پیشبرد این پروژه به عنوان استاد پروژه، کمک‌های فراوانی به این جانب داشتند، کمال تشکر را دارم.

سارا تاجرانی  
خرداد ۱۴۰۳

## چکیده

پهپادهای تجاری که به عنوان هواپیماهای بدون سرنشین<sup>۱</sup> نیز شناخته می‌شوند، به سرعت در حال رایج شدن هستند. این هواپیماها در بسیاری از کاربردهای مختلف مانند نظارت برای رویدادهای ورزشی، حمل و نقل تجهیزات و کالاهای اضطراری، فیلمبرداری، عکاسی هوایی و بسیاری از فعالیت‌های دیگر مورد استفاده قرار می‌گیرند.

هدف این پژوهه توسعه سیستمی است که از حرکات دست به عنوان روشی برای کنترل پرواز پهپاد استفاده شود. بدین صورت که با استفاده از روش‌های بینایی ماشین<sup>۲</sup>، روشی بصری برای ارتباط بدون عامل بین پهپاد و اپراتور آن ایجاد شود. روش‌های مبتنی بر بینایی ماشین بر توانایی دوربین هواپیماهای بدون سرنشین متکی هستند. بدین صورت که تصاویر اطراف را گرفته و با استفاده از ترجمه تصاویر و تشخیص الگوی دست، اطلاعات معناداری را استخراج می‌کنند. ساختار این پژوهه از دو مأذول اصلی تشکیل شده است: تشخیص حرکت دست<sup>۳</sup> و دستور به هواپیمای بدون سرنشین. برای مأذول اول از یک روش یادگیری عمیق<sup>۴</sup> استفاده شده است. الگوریتم‌ها و تکنیک‌های پردازش تصویر به عنوان روشی پویا برای شناسایی ژست‌ها و حرکات دست معرفی شده‌اند. مأذول دوم وظیفه ارتباط با پهپاد را بر عهده دارد. بدین صورت که پیام‌های بین سیستم پیشنهادی و پهپاد متصل به سیستم را ارسال و دریافت می‌کند و طبق آن پیام‌ها عملیات مورد نظر را اجرا می‌کند.

## واژه‌های کلیدی:

پهپاد، هواپیمای بدون سرنشین، ژست دست، بینایی ماشین، شبکه عصبی پیچشی<sup>۵</sup>، حافظه طولانی

<sup>1</sup>Unmanned Aerial Vehicles

<sup>2</sup>Computer Vision

<sup>3</sup>Hand Detection

<sup>4</sup>Deep Learning

<sup>5</sup>Convolutional Neural Network(CNN)

<sup>6</sup>Long Short-Term Memory(LSTM)

<sup>7</sup>Recurrent Neural Network(RNN)

<sup>8</sup>Multilayer Perceptron(MLP)

<sup>9</sup>Machine Learning

کوتاه مدت<sup>۶</sup>، شبکه عصبی بازگشتی<sup>۷</sup>، شبکه پرسپترون چندلایه<sup>۸</sup>، یادگیری ماشین<sup>۹</sup>، رابط انسان و پهپاد<sup>۱۰</sup>

---

<sup>10</sup>Human–Drone Interface

# فهرست مطالب

صفحه

عنوان

۱	.....	۱	مقدمه
۲	.....	۱-۱	مقدمه
۳	.....	۲-۱	چالش‌های استفاده از پهپاد
۳	.....	۳-۱	اهمیت استفاده از بینایی ماشین در پهپاد
۴	.....	۴-۱	تعریف مسئله
۵	.....	۱-۴-۱	چالش‌های اجرای پروژه
۵	.....	۱-۱	مراحل انجام پروژه
۶	.....	۶-۱	جمع‌بندی
۷	.....	۲	کارهای مشابه
۸	.....	۱-۲	مقدمه
۹	.....	۲-۲	مقالات مربوط به ویژگی‌های تصویر
۹	... Hand Gesture Controlled Drones: An Open Source Library	۱-۲-۲	مقاله
۱۱	.... A Real-Time Hand Gesture Recognition Method	۲-۲-۲	مقاله
۱۲	.... مقالات مربوط به ورودی تصویر دست به مدل	۳-۲	۳-۲
۱۲	.... Hand Gestures For Drone Control Using Deep Learning	۱-۳-۲	مقاله
۱۳	UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition	۲-۳-۲	مقاله
۱۵	.... مقالات مربوط به نقاط کلیدی دست	۴-۲	۴-۲
۱۵	. Hand Gesture Recognition System for Real-Time Application	۱-۴-۲	مقاله
	An Improved Hand Gesture Recognition System Using Key-	۲-۴-۲	۴-۲
۱۶	.... points and Hand Bounding Boxes		
۱۷	.... Visual Gesture Recognition Based On Hand Key Points	۳-۴-۲	۳-۴-۲
۱۸	.... MediaPipe Hands: On-Device Real-time Hand Tracking	۴-۴-۲	۴-۴-۲
۱۹	.... مقالات مربوط به آشنایی با پهپاد و اجرای مدل‌های بینایی کامپیوتر روی آن	۵-۲	۵-۲
	Modeling Relation Among Implementing AI-Based Drones	۱-۵-۲	۱-۵-۲
۱۹	.... and Sustainable Construction Project Success		

۲-۵-۲ مقاله Use of A DJI Tello Drone as An Educational Platform in the

۲۰	Field of Control Engineering
۲۱	۶-۲ جمع‌بندی
۲۲	۳ روش انجام پروژه
۲۴	۱-۳ مقدمه
۲۵	۲-۳ انتخاب ژست‌های دست متناسب با حرکت پهپاد
۲۶	۳-۳ دیتاست
۲۶	۴-۳ اهمیت ژست دست
۲۷	۵-۳ کنترل پهپاد
۲۸	۶-۳ ابزارها و نرم افزارهای مورد استفاده
۲۸	۱-۶-۳ کتابخانه TensorFlow
۲۸	۲-۶-۳ کتابخانه Scikit-learn
۲۹	۳-۶-۳ رابط برنامه‌نویسی Keras
۲۹	۴-۶-۳ کتابخانه‌های MediaPipe
۳۰	۵-۶-۳ کتابخانه NumPy
۳۰	۶-۶-۳ کتابخانه Matplotlib
۳۱	۷-۶-۳ کتابخانه OpenCV
۳۱	۸-۶-۳ پهپاد DJI Tello
۳۲	۷-۳ فلوچارت اجرای پروژه
۳۳	۱-۷-۳ تشخیص دست <sup>۱۱</sup>
۳۴	۲-۷-۳ تشخیص نقاط عطف دست <sup>۱۲</sup>
۳۶	۳-۷-۳ پیش‌بینی ژست دست
۳۷	۴-۷-۳ محدود کننده جریان زمان واقعی <sup>۱۳</sup>
۳۷	۵-۷-۳ تشخیص به مستطیل <sup>۱۴</sup>

<sup>11</sup>Hand Detection

<sup>12</sup>Hand Landmark

<sup>13</sup>Real Time Flow Limiter

<sup>14</sup>Detection To Rectangle

<sup>15</sup>Image Cropping

۳۸	.....	۶-۷-۳ برش تصویر <sup>۱۵</sup>
۳۸	.....	۷-۷-۳ نقاط عطف یه مستطیل <sup>۱۶</sup>
۳۹	.....	۸-۷-۳ ارائه کننده حاشیه نویسی <sup>۱۷</sup>
۳۹	.....	۹-۷-۳ مدل های کلاس بندی برای تعیین ژست دست
۳۹	.....	۱۰-۷-۳ شبکه پرسپترون چند لایه
۴۰	.....	۱۱-۷-۳ شبکه عصبی کانولوشنال
۴۱	.....	۱۲-۷-۳ شبکه عصبی بازگشتی
۴۱	.....	۱۳-۷-۳ شبکه های حافظه کوتاه مدت بلند مدت
۴۲	.....	۸-۳ پیش پردازش
۴۴	.....	۹-۳ پس پردازش
۴۴	.....	۱۰-۳ جمع بندی
۴۵	.....	<b>۴ نتایج و ارزیابی</b>
۴۶	.....	۱-۴ مقدمه
۴۶	.....	۲-۴ ارزیابی عملکرد مدل ها
۴۶	.....	۱-۲-۴ دقت
۴۷	.....	۲-۲-۴ صحت
۴۸	.....	۳-۲-۴ فراخوانی
۴۸	.....	۴-۲-۴ امتیاز F1
۴۸	.....	۵-۲-۴ گزارش معیارهای ارزیابی در مدل ها
۴۹	.....	۶-۲-۴ ماتریس درهم ریختگی <sup>۱۸</sup>
۴۹	.....	۳-۴ نمودارهای دقت و خطاب بر حسب دوره
۵۱	.....	۴-۴ سرعت اجرای برنامه
۵۲	.....	۵-۴ سخت افزار مورد نیاز
۵۲	.....	۶-۴ مقایسه دقت پروژه ما با کارهای مشابه
۵۲	.....	۷-۴ جمع بندی

<sup>16</sup>Landmarks To Rectangle

<sup>17</sup>Annotation Renderer

<sup>18</sup>Confusion Matrix

۵۵	.....	۵	نتیجه‌گیری و پیشنهادات
۵۶	.....	۱-۵	۱- مقدمه
۵۶	.....	۲-۵	۲- نتیجه‌گیری
۵۶	.....	۳-۵	۳- پیشنهادات
۵۶	.....	۱-۳-۵	۱-۳- محدودیت وجود پهپادهای منبع باز
۵۷	.....	۲-۳-۵	۲-۳- وجود چندین دست در تصویر
۵۸	.....	۳-۳-۵	۳-۳- شناسایی یک دست در دو مستطیل در نتیجه دو خروجی دست
۵۸	.....	۴-۳-۵	۴-۳- اجرا روی کارت گرافیکی <sup>۱۹</sup>
۵۹	.....	۵-۳-۵	۵-۳- اجرای بروزه با برنامه C++
۶۰	.....	۶-۳-۵	۶-۳- استفاده از مدل‌های موازی برای تشخیص ژست دست
۶۰	.....	۷-۳-۵	۷-۳- توسعه برنامه با استفاده از سایر قسمت‌های بدن انسان
۶۱	.....	۴-۵	۴- جمع‌بندی
۶۳	.....		کتابنامه

---

<sup>19</sup>Graphics Processing Units

# فهرست تصاویر

صفحه

شکل

۱-۲	ویژگی های Haar برای استفاده از آستانه رنگ پوست برای تشخیص دست . . . . .	۱۰
۲-۲	چارچوب کنترل پهپاد مبتنی بر ژست دست با کمک ویژگی های Haar . . . . .	۱۰
۳-۲	تشخیص کف دست و انگشتان . . . . .	۱۲
۴-۲	معماری VGG-16 . . . . .	۱۳
۵-۲	تشخیص نقطه کلید و تطبیق توسط SIFT . . . . .	۱۵
۶-۲	معماری ساختار شبکه های عصبی دو خط لوله . . . . .	۱۷
۷-۲	ورودی ها و خروجی های مدل تشخیص نقاط کلیدی دست . . . . .	۱۸
۸-۲	ارتباط برنامه پایتون با پهپاد DJI Tello . . . . .	۲۱
۹-۳	تشخیص ژست دست با کمک نقاط کلیدی دست . . . . .	۲۴
۱۰-۳	نمونه ای از ژست های انتخاب شده در مجموعه داده ها . . . . .	۲۵
۱۱-۳	برخی کاربردهای کتابخانه MediaPipe . . . . .	۳۰
۱۲-۳	اطلاعات پهپاد DJI Tello . . . . .	۳۲
۱۳-۳	فلوچارت پروژه . . . . .	۳۳
۱۴-۳	خط لوله تشخیص دست . . . . .	۳۴
۱۵-۳	معماری مدل آشکارساز کف دست . . . . .	۳۵
۱۶-۳	معماری مدل نقطه عطف دست. این مدل دارای سه خروجی است که یک استخراج کننده ویژگی را به اشتراک می گذاردند. هر سر توسط مجموعه داده های مربوطه که با همان رنگ مشخص شده اند آموزش داده می شود. . . . .	۳۶
۱۷-۳	موقعیت ۲۱ نقطه کلیدی در ناحیه دست . . . . .	۳۶
۱۸-۳	پیدا کردن مستطیل حاوی دست . . . . .	۳۸
۱۹-۳	نمونه ای از پرسپترون های چند لایه دارای دو لایه پنهان . . . . .	۴۰
۲۰-۳	نمونه معماری شبکه عصبی کانولوشنال . . . . .	۴۱
۲۱-۳	نمونه معماری شبکه عصبی بازگشتی . . . . .	۴۲
۲۲-۳	نمونه معماری شبکه حافظه کوتاه مدت بلند مدت . . . . .	۴۳
۲۳-۴	معیارهای ارزیابی برای تشخیص ژست دست در مدل MLP . . . . .	۴۹

۵۰	۲-۴ ماتریس درهم ریختگی در مدل MLP ..
۵۱	۳-۴ منحنی‌های کم‌برازش، بهینه و بیش‌برازش ..
۵۱	۴-۴ نمودار روند دقت و خطای بر حسب دوره در داده‌های آموزش و تست در مدل CNN ..
۵۸	۱-۵ شناسایی یک دست دو مرتبه

## فهرست جداول

صفحه

جدول

۴۷	۱-۴	جدول ارزیابی دقت مدل‌ها
۵۱	۲-۴	جدول ارزیابی زمان پاسخگویی مدل‌ها
۵۲	۳-۴	جدول ارزیابی سخت‌افزار موردنیاز مدل‌ها
۵۳	۴-۴	جدول مقایسه پروژه با کارهای مشابه

# فصل اول

## مقدمه

## ۱-۱ مقدمه

پهپادها یا به عبارتی هواپیماهای بدون سرنشین امروزه در صنایع مختلف به عنوان یک فناوری بسیار گسترده و کارآمد مورد استفاده قرار می‌گیرند. هواپیماهای بدون سرنشین اساساً به عنوان ربات‌های پرنده‌ای دیده می‌شوند که عملکردهای متعددی مانند جمع‌آوری داده‌ها و سنجش از محیط اطراف را بر عهده دارند [۱]. از جمله این صنایع می‌توان به کشاورزی، ساخت و ساز، خدمات حمل و نقل و نقشه‌برداری اشاره کرد. یکی از دلایل اصلی افزایش کاربرد این هواپیماهای بدون سرنشین، کارایی بالای آنها است. این فناوری نه تنها به دلیل سرعت بالا در پوشش دهی مساحت‌های گسترده، بلکه به دلیل قابلیت برنامه‌ریزی و استفاده در صنایع مختلف مورد توجه قرار می‌گیرد. همچنین، صرفه‌جویی در هزینه‌های مالی و جانی و افزایش امنیت نیز از جمله عوامل مهمی است که اهمیت پهپادها را بیشتر می‌کند [۲].

در حال حاضر، ربات‌های پرنده در مشاغل مختلف مانند سیستم‌های تحویل بسته استفاده می‌شوند [۳]. به عنوان مثال، شرکت‌هایی مانند آمازون و UPS از پهپادهای چند روتور برای تحویل بسته‌های خود استفاده می‌کنند [۴]. در پی این موضوع، بسیاری از شرکت‌های تولید کننده پهپاد تشویق شدند تا انواع مختلفی از ویژگی‌های نرم‌افزاری و سخت‌افزاری مانند حسگرها را به پهپادها اضافه کنند، که ابتدایی ترین آنها دوربین است. دوربین بصری یک حسگر ضروری برای پهپادهای فعلی است که آنها را به پهپادهای کاربردی و متعدد در بازار تبدیل می‌کند [۵]. در ادامه زمینه مطالعاتی جدیدی به نام رابط هواپیماهای بدون سرنشین و انسان گشوده شد تا تعامل بین پهپاد و انسان را پیشرفت دهد، که این تعامل مجموعه دستگاه‌های سنتی مانند کنترلر رادیویی<sup>۱</sup> تا کنترل پهپادها با استفاده از وضعیت بدن و دست انسان را شامل می‌شد [۶].

یکی از رویکردهای مورد استفاده برای افزایش کاربرد و دسترسی به پهپادها، استفاده از بینایی ماشین است. این ویژگی معمولاً از طریق پردازش تصویر و با استفاده از شبکه‌های عصبی به کار می‌رود. پهپادهایی که با مدل‌های بینایی ماشین آموزش می‌بینند، توانایی تحلیل تصاویر و ویدئوهایی که از محیط اطراف دریافت می‌کنند را دارا هستند. این قابلیت به پهپاد این امکان را می‌دهد که بدون نیاز به تداخل انسانی، وظایفی همچون امنیت، ارسال کالا، پست و این چنین موارد را انجام دهد [۷]. می‌توان گفت هدف اصلی استفاده از بینایی ماشین در پهپادها برای به حداقل رساندن دخالت انسان به صورت مستقیم است. این امر پهپاد را قادر می‌سازد تا تشخیص اشیاء، تشخیص چهره، تحلیل تصاویر، شناسایی الگوهای مختلف و

<sup>۱</sup>Radio Controller

مواردی از این دست را به صورت خودکار انجام دهنده [۸].

## ۲-۱ چالش‌های استفاده از پهپاد

استفاده از پهپادها، با چالش‌های متعددی همراه است. یکی از این چالش‌ها، محدودیت زمان پرواز است که پس از مدتی نیاز به شارژ مجدد دارند. همچنین، محدودیت‌های محیطی نیز می‌تواند به چالش‌هایی بدل شوند؛ زیرا پهپادها به شرایط محیطی مانند آب و هوا، یا وزن و ارتفاع حساس هستند و این موارد می‌توانند در طراحی آنها تأثیر بهسزایی داشته باشد. در ادامه باید به میزان اهمیت امنیت اطلاعات هم اشاره کرد، زیرا پهپادها به دلیل استفاده از سیستم‌های موقعیت‌یاب و ارتباطات بی‌سیم ممکن است در برابر حملات سایبری آسیب‌پذیر باشند و اطلاعات مهمی که توسط آنها مخابره می‌شود، در معرض خطر قرار گیرد.

همچنین می‌توان به برخی چالش‌هایی که ما هم در این پروژه به صورتی با آنها سر و کار داریم و در تلاشیم آنها را از بین ببریم یا کمتر کنیم اشاره کرد. مانند انتقال اطلاعات، زیرا برای ارتباط با پهپادها از شبکه‌های بی‌سیم استفاده می‌شود و در شرایطی مانند اشباع شبکه یا فاصله بین پهپاد و کنترل‌کننده، ممکن است این ارتباط دچار مشکل شود. علاوه بر این، محدودیت محاسباتی پهپاد نیز با توجه به اهدافی که برای آن در نظر گرفته شده می‌تواند چالش برانگیز باشد؛ زیرا پهپادها به دلیل محدودیت‌های سخت‌افزاری و نرم‌افزاری، دارای پردازشگرها و حافظه‌های محدودی هستند [۹]. قابل ذکر است که با ادامه پیشرفت فناوری پهپاد، می‌توان انتظار داشت که ویژگی‌های جدید و نوآورانه‌ای برای از بین بردن این محدودیت‌ها و چالش‌ها به پهپادهای آینده اضافه شود.

## ۳-۱ اهمیت استفاده از بینایی ماشین در پهپاد

طبق اعلام پیش‌بینی اداره هوانوردی فدرال، بازار هوایی‌های بدون سرنشین تا سال ۲۰۲۵ به ۱۷ میلیارد خواهد رسید و ۷ میلیون هوایی‌مای بدون سرنشین به آسمان پرواز خواهند کرد. پهپادهای کنترل از راه دور به تدریج به دستگاههای نیمه خودکار یا کاملاً خودکار تبدیل می‌شوند که از پیاده‌سازی مبتنی بر هوش مصنوعی بهره می‌برند. در این پروژه هدف ما هدایت پهپاد با استفاده از علائم دست مبتنی بر بینایی ماشین است که یک حوزه پژوهشی مهم در ترکیب هوش مصنوعی و رباتیک است. استفاده از حرکات دست در کنترل هوایی‌های بدون سرنشین در حال تبدیل شدن به یک روش محبوب برای

تعامل کاربر و پهپاد است.

## ۴-۱ تعریف مسئله

این پایان نامه یک سیستم کامل برای کنترل هواپیماهای بدون سرنشین با استفاده از حرکات دست پیشنهاد می‌کند. سیستم پیشنهادی باید در زمان واقعی<sup>۲</sup> کار کند و دقت<sup>۳</sup> بالایی داشته باشد تا بتواند به بهترین نحو ممکن پهپاد را کنترل کند [۶].

در این روش، از سیستم بینایی ماشین به منظور تشخیص و تحلیل حرکات دست از روی تصاویر ویدئویی پهپاد استفاده می‌شود. با استفاده از الگوریتم‌های یادگیری عمیق و شبکه‌های عصبی، سیستم قادر است علائم و حرکات دست را تشخیص داده و به تفسیر آنها بپردازد. سپس، براساس تحلیل این حرکات، دستورات مربوطه برای حرکت و کنترل پهپاد را صادر کند. بدین صورت این روش نه تنها از دقت بالا برای تشخیص و تفسیر حرکات دست برخوردار است، بلکه قابلیت ارائه یک رابط کاربری بین انسان و پهپاد را نیز فراهم می‌کند. به طوری که با استفاده از حرکات دست کاربر قادر است به راحتی و بدون نیاز به دستگاه‌های کنترل خارجی، پهپاد را هدایت کند [۱۰].

استفاده از حرکات دست برای کنترل پهپاد مزایای زیادی دارد. ابتدا باید گفت که حرکات دست یک شکل طبیعی ارتباطی هستند و استفاده از آنها برای کنترل پهپاد یک روش شهودی و طبیعی برای تعامل با فناوری است. این امر باعث می‌شود که کاربران بتوانند به راحتی و با کمترین تلاش پهپاد را کنترل کنند. استفاده از حرکات دست به کاربر اجازه می‌دهد پهپاد را با سرعت و دقت بیشتری کنترل کند و محدودیت‌های مرتبط با دستگاه‌های کنترل سنتی را کاهش دهد. همچنین، این روش، حرکت و دنبال کردن پهپاد را آسان‌تر می‌کند و امکان جابجایی پهپاد در فضا را بهبود می‌بخشد.

استفاده از علائم دست سبب کاهش نیاز به دستگاه‌های کنترل پیچیده می‌شود و به این ترتیب، پهپاد را برای طیف وسیع‌تری از کاربران قابل دسترس می‌کند. این امر به کاربرانی که با دستگاه‌های کنترل سنتی آشنایی ندارند، امکان استفاده آسان از پهپاد را می‌دهد. همچنین، با توجه به چالش‌هایی که از قبل بیان شده است، این روش خطرات مرتبط با اتصالات بی‌سیم بین کنترلر و پهپاد را کاهش می‌دهد و دقت در کنترل پهپاد در محیط‌های پرتلاطم و مختلف را افزایش می‌دهد.

برای پیاده‌سازی این پروژه می‌توان از شبکه‌های عصبی عمیق<sup>۴</sup>، مانند شبکه عصبی کانولوشن،

<sup>2</sup>Real-Time

<sup>3</sup>Accuracy

<sup>4</sup>Deep Neural Network

استفاده کرد. دلیل استفاده از این معماری قابلیت استخراج خودکار ویژگی‌ها با توجه به الگوریتم طبقه‌بندی تصاویر<sup>۵</sup> است. عملکرد شبکه عصبی کانولوشن به این گونه است که ویژگی‌ها را با توجه به لایه‌های پنهان می‌آموزد، همچنین می‌تواند تعداد پارامترها را بدون به خطر انداختن دقیق مدل تغییر دهد.

## ۱-۴-۱ چالش‌های اجرای پروژه

وجود سخت‌افزاری مناسب برای اجرای این پروژه الزامی است. پهپاد انتخاب شده در ابتدا باید شامل یک دوربین با رزوولوشن نسبتاً بالا باشد تا ژست دست تا فاصله سه متری از پهپاد به وضوح گرفته شود. در ادامه از آنجایی که زمان واقعی در این پروژه از اهمیت بالایی برخوردار است پهپاد باید پردازنده نسبتاً قوی داشته باشد تا بتواند در کمترین زمان ممکن عکس را از دوربین دریافت کرده و انتقال دهد و پس از اجرای مدل دستور را اجرا کند. از دیدگاهی دیگر، از آنجایی که این ارتباطات در کنار حرکت پهپاد انرژی زیادی می‌طلبند، لذا باید پهپادی را انتخاب کرد که از شامل باطری بادوام و باکیفیت باشد تا به مرور زمان برای استفاده کننده آزاردهنده نباشد.

همچنین پهپاد مدد نظر ما برای این پروژه باید توانایی ارتباط با برنامه پایتون را نیز داشته باشد تا بتوان دستورهای پیش‌بینی شده توسط شبکه‌های هوش مصنوعی پیاده‌سازی شده با این زبان را روی آن اجرا کرد.

## ۱-۵ مراحل انجام پروژه

۱. انتخاب ژست‌های مناسب و مفهومی برای کنترل پهپاد

۲. پیاده‌سازی کد برای جمع‌آوری مجموعه داده

۳. پیاده‌سازی شبکه مربوط به پیدا کردن کف دست

۴. پیاده‌سازی شبکه پیدا کردن نقاط کلیدی دست

۵. پیاده‌سازی شبکه‌هایی برای تعیین ژست دست

۶. آموزش به مدل‌ها

<sup>۵</sup>Image Classification

۷. بهینه‌سازی مدل‌ها

۸. تست مدل‌ها و انتخاب بهترین مدل

۹. پیاده سازی رأی‌گیری پنجره‌ای<sup>۶</sup>

۱۰. اجرای مدل روی پهپاد

## ۱-۶ جمع بندی

هدف این پروژه پیاده‌سازی برنامه‌ای کاربردی بر روی پهپاد است تا بتوان ۹ ژست دست از پیش تعیین شده را شناسایی و با توجه به آنها دستور به پهپاد داده‌شود. در این پروژه موارد زیر از اهمیت بالایی برخوردار هستند:

- دقت بالای مدل زیرا در صورت انجام نادرست دستورات امکان برآورد هزینه مالی وجود دارد.
- پیچیدگی کم و سرعت بالای تشخیص ژست دست، زیرا زمان واقعی اجرای دستورات مهم است و باید در کمترین زمان ممکن رخ دهد تا مورد پسند کاربر باشد.

---

<sup>6</sup>Window Voting

# فصل دوم

## کارهای مشابه

## ۱-۲ مقدمه

در این فصل هدف ما بررسی کارهای مشابه است تا بتوان از آنها در روند بهبود پروژه کمک گرفت. همچنین با توجه به نتایج و ارزیابی پروژه‌های دیگر می‌توان بستری فراهم کرد تا نتیجه پروژه را با دیگر کارهای مشابه مقایسه کرد.

به صورت کلی پروژه‌هایی با هدف کنترل پهپاد با ژست دست در ۳ دسته قرار می‌گیرند.

- کنترل پهپاد با کمک بینایی ماشین که شامل شبکه‌هایی برای پردازش تصویر است.
- کنترل پهپاد با دستکش‌های سنسور دار از جمله سنسور IMU که نیازمند سخت‌افزار خاص برای پیدا کردن موقعیت نقاط کلیدی دست است. با استفاده از یک شبکه عصبی و موقعیت این نقاط می‌توان ژست دست را تشخیص داد. مانند پروژه‌های [۱۰] و [۱۱].
- با کمک دستگاه کنترل کننده حرکت جهشی<sup>۱</sup> که با ردیابی دست، ویژگی‌های آن از جمله موقعیت کف دست و انگشتان، جهت دست و انگشتان، طول و عرض انگشتان، موقعیت مفصل و استخوان‌ها را با دقت بالا اندازه‌گیری کرده و با کمک شبکه‌های عصبی ژست دست تشخیص داده می‌شود. پروژه‌ی [۱۲] و [۱۳] نمونه‌ای از این جمله پروژه‌ها هستند.

از بین این موارد پروژه ما مربوط به اولین گزینه است که تنها سخت‌افزار مورد نیاز به جز پهپاد دوربین نصب شده روی آن است. پروژه‌های مشابه با کار ما که با کمک پردازش تصویر پهپاد را کنترل می‌کنند به ۳ دسته کلی تفکیک می‌شوند.

۱. استخراج ویژگی‌های تصویر در هر فریم که با توجه به نیازهای مسئله می‌تواند متفاوت باشد، و پیش‌بینی ژست دست براساس آن ویژگی‌ها.

۲. تشخیص دست<sup>۲</sup> برای پیدا کردن موقعیت دست در هر فریم تصویر و ورودی پیکسل‌های RGB آن به مدل و در نهایت کلاس‌بندی ژست دست.

۳. استخراج نقاط کلیدی<sup>۳</sup> دست در تصویر و ورودی آنها به مدل برای کلاس‌بندی و تشخیص ژست دست.

<sup>1</sup>Leap Motion Controller

<sup>2</sup>Hand Detection

<sup>3</sup>Key Point

در ادامه این فصل به بررسی مقالات در این سه زمینه و مقالاتی که بر روی پهپاد و پیاده‌سازی مدل‌های هوش مصنوعی تمرکز دارند می‌پردازیم.

## ۲-۲ مقالات مربوط به ویژگی‌های تصویر

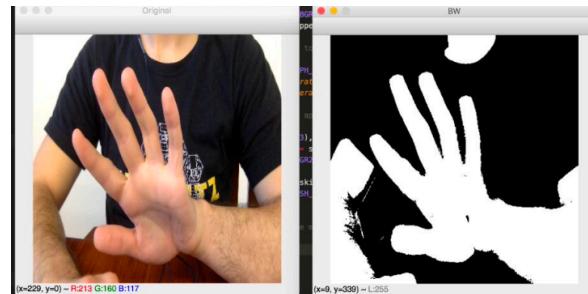
مقالات به کار برده شده در این قسمت بر چگونگی تعیین ژست دست با توجه به تصویر داده شده تمرکز دارند. برخی از این مقالات چکونگی ارتباط با پهپاد را نیز پوشش می‌دهند، اما نکته قابل بررسی برای ما چگونگی استخراج ویژگی‌های تصویر و استفاده از آنها برای تعیین ژست دست است.

### ۱-۲-۲ مقاله Hand Gesture Controlled Drones: An Open Source Library

در این پروژه، تمرکز بر پیاده‌سازی یک سیستم کنترل برای هواپیماهای بدون سرنشین با استفاده از حرکات دست است که مشابه رویکرد مورد بحث در مقاله می‌باشد. هدف اصلی این پروژه استفاده از شبکه‌های عصبی یادگیری عمیق برای تشخیص لحظه‌های حرکات دست پویا برای کنترل پرواز پهپاد است. این تشخیص بر اساس ویژگی‌های Haar که با توجه به سایه‌ها و رنگ‌های درون تصویر تعیین می‌شوند، انجام می‌شود [۵].

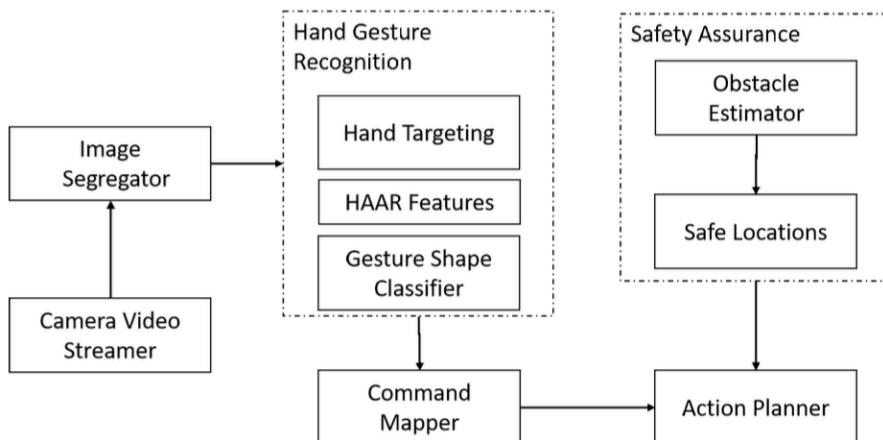
#### روش‌شناسی

در این پروژه، ابتدا از یک شبکه عصبی برای تشخیص موقعیت دست استفاده می‌شود که به عنوان یک مازول پیش‌پردازش عمل می‌کند و با دقت بالا موقعیت دست را تشخیص می‌دهد. پس از تشخیص موقعیت دست، ویژگی‌های Haar از تصویر استخراج می‌شوند. آنها مجموعه‌ای از الگوریتم‌های تشخیص ویژگی هستند که از تصاویر استفاده می‌کنند تا ویژگی‌های مدنظر را شناسایی کنند. ویژگی‌های Haar بر اساس تغییرات گرادیان در تصویر تعیین می‌شوند و به عنوان الگوهای محلی برای تشخیص حرکات دست استفاده می‌شوند. در ادامه، از شبکه‌های عصبی یادگیری عمیق برای تشخیص حرکات دست پویا برای کنترل پرواز پهپاد استفاده می‌شود. این شبکه‌ها عملکرد پیچیده‌ای دارند و با استفاده از ویژگی‌های Haar به عنوان داده‌ی ورودی می‌آموزند تا حرکات دست را تشخیص دهند و بر اساس آنها دستورات برای حرکت پهپاد را تعیین کنند.



شکل ۱-۲ ویژگی های Haar برای استفاده از آستانه رنگ پوست برای تشخیص دست

این سیستم شامل مراحل پیش‌پردازش داده‌ها، انتخاب ویژگی، مازول شبکه عصبی برای تشخیص ژست و مازول کنترل پهپاد برای ترجمه ژست‌های شناسایی شده به دستورات حرکت پهپاد می‌باشد. علاوه بر این، از مدل ماشین بردار پشتیبان<sup>۴</sup> برای کلاس‌بندی و تشخیص حرکات دست استفاده شده است. ماشین بردار پشتیبان به عنوان یک ماشین یادگیری است که برای مسائل دسته‌بندی و رگرسیون استفاده می‌شود و در این پروژه برای تشخیص حرکات دست و ترجمه آنها به دستورات حرکت پهپاد مورد استفاده قرار گرفته است. این روش امکان کنترل دقیق و پویا برای پهپاد را فراهم می‌کند و از قابلیت‌های پیشرفتی یادگیری عمیق برای تشخیص حرکات دست بهره می‌برد.



شکل ۲-۲ چارچوب کنترل پهپاد مبتنی بر ژست دست با کمک ویژگی‌های Haar

### نتیجه بدهست آمده

این پروژه دقیق بالایی در تشخیص ژست دست و کنترل پرواز پهپاد دارد. پنج حالت دست در این پروژه مدنظر قرار گرفته‌اند و دقیق متوسط این مدل، برابر ۹۷.۴۷۱ درصد است که نشان‌دهنده عملکردی بسیار

<sup>4</sup>Support Vector Machine

عالی است. اما لازم به ذکر است که این دقت در پس زمینه های بهم ریخته و شرایط نوری مختلف بسیار متغیر است زیرا ویژگی Haar به سایه و رنگ های درون تصویر بسیار حساس است.

## A Real-Time Hand Gesture Recognition Method ۲-۲-۲ مقاله

این مقاله در زمینه پردازش تصویر و تشخیص ژست های دست، از روش های مبتنی بر مدل ظاهری<sup>۵</sup> به عنوان یک رویکرد موثر استفاده کرده است. این روش از ویژگی های تصویری و حرکتی دست برای تشخیص و تعیین ژست های دست استفاده می کند. در این مقاله، روش تشخیص ژست های دست به صورت زمان واقعی و قابل اعتماد است که از تشخیص دست، ردیابی دست<sup>۶</sup>، تقسیم بندی دست<sup>۷</sup> و تشخیص ویژگی های مقیاس-فضا<sup>۸</sup> برای تشخیص ژست ها استفاده می کند [۱۴].

### روش شناسی

در مقاله از ترکیب متنوعی از روش ها و ویژگی های تصویری استفاده می شود. این ویژگی ها به ترتیب به صورت زیر عمل می کنند.

۱. استفاده از روش Adaboost برای تشخیص دست، که یک روش معتبر برای تشخیص اشیاء در تصویر است.

۲. پیگیری دست با استفاده از تشخیص حرکت و رنگ، که از ترکیب تکنیک های جریان نوری و نشانه رنگ برای ردیابی دست در تصاویر استفاده می شود.

۳. تقسیم بندی دست با استفاده از اطلاعات حرکت و رنگ برای تمایز دست از پس زمینه و اشیاء دیگر.

۴. تشخیص ویژگی های مقیاس-فضا برای تشخیص ژست های دست، که برای شناسایی ساختارهای شبیه به کف دست و انگشتان استفاده می شود تا نوع ژست دست توسط ترکیب این ساختارها تعیین شود.

در نهایت این روش ها و ویژگی ها با هم ترکیب می شوند تا یک سیستم تشخیص دست پایدار و دقیق

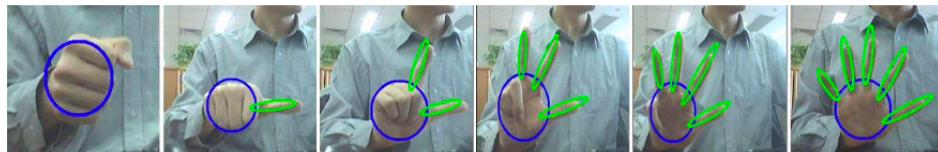
<sup>5</sup>Appearance Model

<sup>6</sup>Hand Tracking

<sup>7</sup>Hand Segmentation

<sup>8</sup>Scale-Space Feature Detection

برای استفاده در رابط کاربری تعاملی و تشخیص ژستهای دست در زمینه‌های مختلف ارائه شود.



شکل ۲-۲ تشخیص کف دست و انگشتان

### نتیجه بدست آمده

اعمال این روش‌ها نتایج قابل قبولی را به همراه داشته است. دقیق مدل در تشخیص ژستهای دست به صورت میانگین ۹۳.۸ درصد بوده و از جمله نتایج مهم آزمایشات می‌توان به تشخیص صحیح ۲۴۳۶ فریم از کل ۲۵۹۶ فریم ضبط شده اشاره کرد. این نتایج نشان می‌دهند که روش ارائه شده در این مقاله عملکرد قابل قبولی در تشخیص ژستهای دست دارد و می‌تواند به عنوان یک روش موثر برای تعاملات زمان واقعی استفاده شود.

## ۳-۲ مقالات مربوط به ورودی تصویر دست به مدل

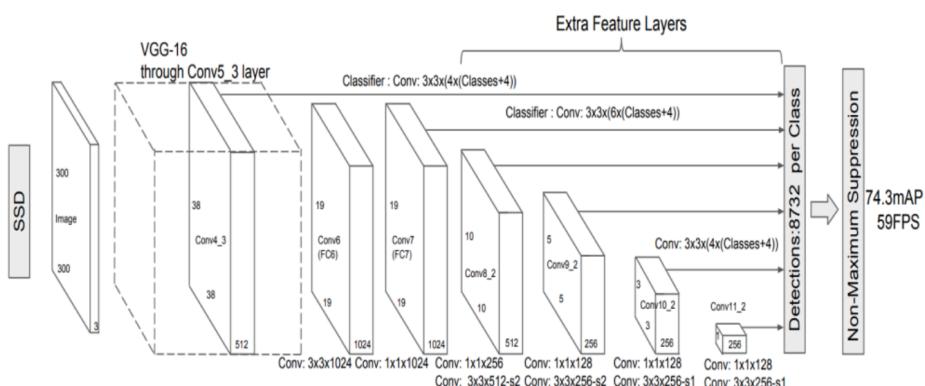
مقالات این دسته از جمله پژوههایی هستند که بیشتر از آنکه بر روی پیش پردازش کار کنند باید بر روی معماری خود شبکه تمرکز کنند. در این مقالات تمام یا بخشی از تصویر گرفته شده به صورت یک ماتریس از تصویر با پیکسل‌های متعدد به مدل داده می‌شود. رویکردی که در این نوع پروژه‌ها در پیش گرفته می‌شود، تشخیص موقعیت دست است تا بتوان تنها قسمتی از تصویر را به ورودی شبکه داد که دست در آن وجود دارد تا در حد ممکن اندازه ورودی شبکه و دقیق آن افزایش یابد. در اینگونه مقالات باید توجه داشت که معماری شبکه را به گونه‌ای برگزید تا در پردازش تصویر بتوان ویژگی‌های تصویر مورد نیاز را استخراج کرد.

## ۱-۳-۲ مقاله Hand Gestures For Drone Control Using Deep Learning

این پروژه نیز با هدف کنترل پهپاد با استفاده از حرکات دست و شبکه‌های عمیق یادگیری انجام شده است تا بتوان ۹ حالت مختلف دست را شناسایی و به پهپاد، دستور موردنظر کاربر را ارسال کرد [۶].

## روش‌شناسی

در این تحقیق، از معماری شبکه عصبی عمیق VGG-16 برای تشخیص و تعیین حرکات دست و کنترل پهپاد استفاده شده است. شبکه VGG-16 یکی از معروف‌ترین و پرکاربردترین شبکه‌های عمیق در زمینه بینایی ماشین است که شامل ۱۶ لایه عصبی با لایه‌های کانولوشن و ادغام<sup>۹</sup> می‌باشد. این شبکه برای استخراج ویژگی‌های مهم از تصاویر استفاده می‌شود. ورودی این شبکه تصاویری است که از دوربین متصل به پهپاد گرفته می‌شود و سپس این تصاویر به شبکه وارد می‌شوند. خروجی این شبکه شامل تشخیص حرکات دست مانند حرکات مختلف انگشتان و دست‌ها است. سپس این اطلاعات برای ارسال دستورات کنترلی به پهپاد فرستاده می‌شوند.



شکل ۴-۲ معماری VGG-16

## نتیجه بدست آمده

در این پژوهه ۹ حالت دست مدنظر قرار گرفته شده و دقت آن برابر ۸۳.۳ درصد است. که در بهترین حالت ممکن با پس‌زمینه‌ی مناسب بدست آمده و باید در نظر گرفت که دقت بالایی برای کنترل پهپاد به حساب نمی‌آید.

## ۲-۳-۲ مقاله UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition

این مقاله به منظور کنترل پهپاد یا خلبان خودکار با استفاده از حرکت دست پیاده‌سازی شده است. به عنوان مثال، حرکت دست از چپ به راست نشان‌دهنده حرکت پهپاد به راست می‌باشد. برای اجرای این

<sup>9</sup>Pooling

برنامه، شبکه P-CNN طراحی شده است تا بتواند مفهوم تصاویر را ترجمه کند [۱۵].

### روش‌شناسی

در این مقاله، از شبکه P-CNN<sup>۱۰</sup> برای تشخیص حرکات دست استفاده شده است. این شبکه از اطلاعات ژست و حرکت دست و بخش‌هایی از بدن استفاده می‌کند. بدین صورت که ابتدا موقعیت دست را با استفاده از جعبه مرزی<sup>۱۱</sup> مشخص می‌کند و سپس تصویر دست را با استفاده از فیلترهای مناسب وارد شبکه P-CNN می‌کند تا بتواند هدف کاربر از حرکت دست را پیش‌بینی کند.

در خروجی این مدل، ۱۳ نوع حرکت مختلف وجود دارد که توسط مدل مفهوم آنها پیش‌بینی می‌شود. این حرکات شامل کل دست از شانه تا انگشتان و حرکات آنها است. این مدل در پروژه برای دستور دادن به هواپیماهای بزرگ بدون سرنشین در فرودگاه‌ها استفاده می‌شود.

اطلاعات ظاهر و حرکت را از بخش‌های مختلف بدن استخراج و از دو شبکه پیش‌آموزش داده شده برای محاسبه ویژگی‌های CNN استفاده می‌کند. در این پروژه برای بخش‌های ژست ظاهری، از شبکه VGG-f استفاده می‌شود، و برای بخش‌های حرکتی از شبکه Action Tube کمک گرفته می‌شود. در نهایت با ترکیب این دو شبکه با استفاده از روش‌های تجمعی مینیمم و ماکسیمم در هر بعد از تصویر و سپس کنار هم گذاشتن آنها می‌توان حرکت دست را با دقت بالا تشخیص داد.

### نتیجه بدست آمده

این مقاله با استفاده از از شبکه P-CNN روشی برای کنترل پهپاد با استفاده از حرکت و ژست دست پیاده‌سازی کرده است. نتایج نشان داده‌اند که این روش با دقت ۹۱.۹ درصد، قابلیت اجرا در پروژه‌های واقعی را دارد. این پروژه نیازمندی‌های پیچیده‌ای برای پیاده‌سازی به همراه دارد که اباید در نظر گرفته شود. این شامل نیاز به سخت‌افزار و نرم‌افزار پیشرفته برای پردازش داده‌های حجمی و پیچیده، تنظیمات محیطی مانند نورپردازی و شرایط جوی، تنظیمات دقیق برای تشخیص حرکات و ژست‌ها، و مسائل امنیتی مرتبط با کنترل پهپاد است. این چالش‌ها نشان از پیچیدگی‌هایی است که برای اجرای موفق این روش در پروژه‌های واقعی باید در نظر گرفته شوند.

<sup>10</sup>Pose-Based Convolutional Neural Network

<sup>11</sup>Bounding Box

## ۴-۲ مقالات مربوط به نقاط کلیدی دست

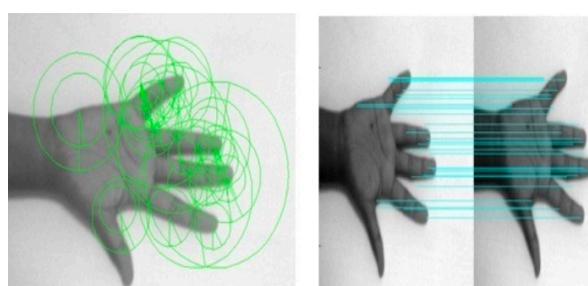
در این چنین مقالات ورودی شبکه بینایی ماشین برای تشخیص ژست، مختصات نقاط کلیدی دست هستند، که به نوعی یک ویژگی تصویر نیز تلقی می‌شوند. بدین صورت که در ابتدای کار دست کاربرد شناسایی شده و سپس نقاط کلیدی آن استخراج می‌شوند تا بتوان حجم داده ورودی به مدل را تا حد امکان ساده‌تر و در عین حال مفیدتر کرد.

### ۱-۴-۲ مقاله Hand Gesture Recognition System for Real-Time Application

این مقاله به بررسی سیستم تشخیص حرکات دست در زمان واقعی می‌پردازد. در این سیستم، از الگوریتم SIFT برای استخراج ویژگی‌ها از تصاویر حرکتی استفاده شده است. سپس، از مدل Bag of Features ماشین بردار پشتیبانی برای تشخیص دقیق حرکات دست در زمان واقعی بهره گرفته شده است [۱۶].

#### روش‌شناسی

همان‌طور که گفته شد در این مقاله از الگوریتم SIFT برای استخراج نقاط کلیدی از تصاویر حرکتی دست استفاده شده است. الگوریتم SIFT یک الگوریتم معروف برای استخراج ویژگی‌های برجسته و تمایزدهنده از تصاویر است که از مقیاس، جهت و از تغییرات نوری برای استخراج این ویژگی‌ها استفاده می‌کند. این ویژگی‌ها به طور قابل توجهی مستقل از مقیاس و جهت تصویر هستند و می‌توانند برای کاربردهای گوناگونی از جمله، تطبیق قابل اعتماد بین تصاویر مختلف یک شیء مورد استفاده قرار گیرند.



شکل ۲-۵ تشخیص نقطه کلید و تطبیق توسط SIFT

#### نتیجه بدست آمده

الگوریتم SIFT به عنوان یک ابزار قدرتمند برای استخراج ویژگی‌های برجسته و تمایزدهنده از تصاویر شناخته شده است. این پروژه توانایی تشخیص حرکات دست را با دقت ۹۰.۸ درصد دارد. در این مقاله

از روشی نوین و متفاوت برای رسیدن به ژست دست استفاده شده است اما قابل ذکر است که این دقت برای پروژه ما می‌تواند خطای زیادی داشته باشد و در روند پروژه استفاده از این الگوریتم کمک‌کننده نیست.

## ۲-۴-۲ مقاله An Improved Hand Gesture Recognition System Using Keypoints and Hand Bounding Boxes

این مقاله یک سیستم بهبود یافته تشخیص حرکات دست با استفاده از نقاط کلیدی و جعبه‌های محدود کننده دست را معرفی می‌کند [۱۷].

### روش‌شناسی

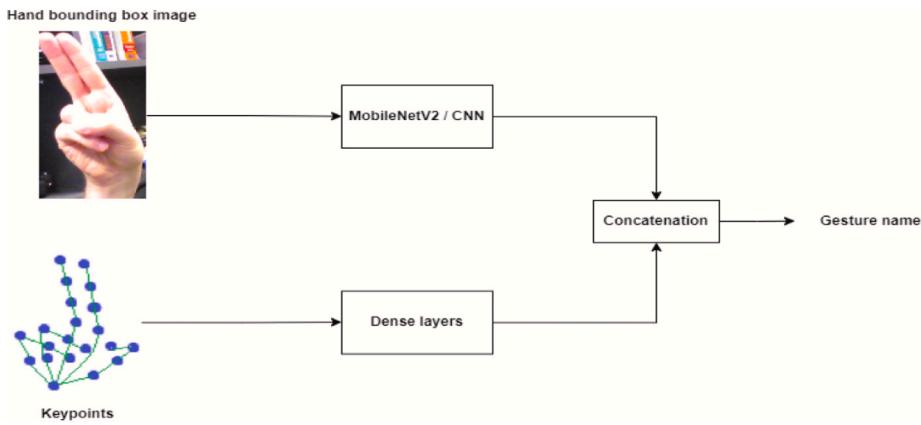
این پروژه از دو لوله موازی به نام‌های "MobileNetV2 + FC" و "CNN + FC" تشکیل شده است که تصاویر جعبه محدود کننده دست و ویژگی‌های استخراج شده از نقاط کلیدی را ترکیب می‌کند و از این طریق ژست دست را پیش‌بینی می‌کند.

در مدل MobileNetV2 + FC، از یک معماری سبک به نام MobileNetV2 برای استخراج ویژگی‌ها از تصاویر جعبه محدود کننده دست استفاده می‌شود. سپس این ویژگی‌ها به یک شبکه عصبی کاملاً متصل (FC) داده می‌شوند تا حرکات دست تشخیص داده شوند.

در مدل CNN + FC، در ابتدا نقاط کلیدی دست که اطلاعات مهمی درباره حرکات دست را شامل می‌شوند پیدا کرده و به یک شبکه عصبی کاملاً متصل وارد می‌شوند تا حرکات دست تشخیص داده شوند. این مدل از لایه‌های کانولوشنی برای کاهش تعداد پارامترها استفاده می‌کند و سپس از لایه‌های کاملاً متصل برای تشخیص حرکات دست استفاده می‌کند.

### نتیجه بدست آمده

در صورتی که دو مدل خروجی‌های متفاوتی را پیش‌بینی کنند، از روش‌های ترکیبی مانند ترکیب احتمالاتی استفاده می‌شود تا بهترین تصمیم برای تشخیص ژست دست گرفته شود. دقت به دست آمده برای تشخیص ۶ ژست دست مختلف در این مقاله در سه دیتاست متفاوت به ترتیب برابر ۹۱، ۹۴ و ۹۶ درصد است.



شکل ۲-۶ معماری ساختار شبکه‌های عصبی دو خط لوله

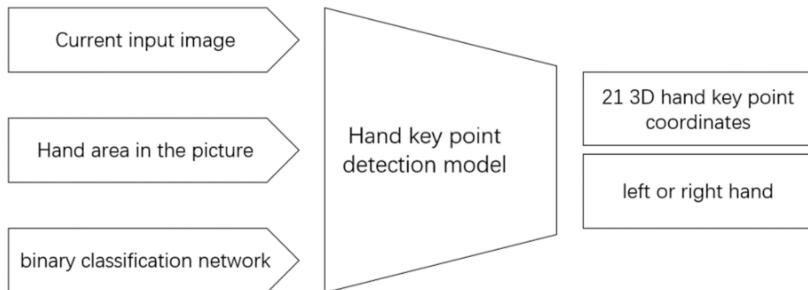
### مقاله ۳-۴-۳ Visual Gesture Recognition Based On Hand Key Points

این مقاله یک روش تشخیص حرکات ژستی بصری بر اساس نقاط کلید دست ارائه می‌دهد. این روش ابتدا نقاط کلید دست را در تصویر ورودی فعلی تشخیص می‌دهد و سپس حرکات تعریف شده را تشخیص می‌دهد [۱۸].

#### روش‌شناسی

مدل ارائه شده در این مقاله شامل دو بخش اصلی است:

- مدل تشخیص کف دست که بر اساس ویژگی‌های سخت دست طراحی شده است. برای تشخیص حضور دست در تصویر، از یک مدل SSD استفاده شده است که به صورت زمان واقعی تشخیص انجام می‌دهد. و نتیجه آن به صورت یک مستطیل نشان داده می‌شود.
- مدل تشخیص نقاط کلیدی دست که پس از تشخیص حضور دست در تصویر، از این مدل برای تشخیص ۲۱ مختصات نقطه کلید سه‌بعدی دست استفاده می‌شود. این مدل از الگوریتم نرمال‌سازی برای محاسبه مختصات افقی و عمودی هر نقطه دست استفاده می‌کند. یک سیستم مختصات فضایی برای به دست آوردن مختصات عمق هر نقطه دست نسبت به مبدأ مختصات تعیین شده است. در نهایت، معنای حرکات در تصویر ورودی بر اساس رابطه مکانی بین مختصات نقاط کلید تعیین می‌شود.



شکل ۷-۲ ورودی ها و خروجی های مدل تشخیص نقاط کلیدی دست

### نتیجه بدست آمده

این روش دارای دقت بالا و عملکرد مناسبی است. دقت متوسط مدل برابر ۸۵.۴ درصد تا ۹۸.۵ درصد است که با توجه به جزئیات مدل و پیاده‌سازی پیش‌پردازش می‌تواند انعطاف بالایی داشته باشد در نتیجه می‌تواند راهکار مناسبی تلقی شود.

## ۴-۴-۲ مقاله MediaPipe Hands: On-Device Real-time Hand Tracking

در این مقاله، از کتابخانه MediaPipe برای پیش‌بینی ۲۱ نقطه کلیدی دست استفاده شده است. این پروژه در موارد گوناگونی از جمله تشخیص ژست دست و افکت‌های واقعیت افزوده <sup>۱۲</sup> کاربرد دارد [۱۹].

### روش‌شناسی

برای پیش‌بینی ژست دست در این پروژه، از دو شبکه کانولوشن استفاده شده است. شبکه اول برای پیدا کردن کف دست در تصویر استفاده می‌شود و شبکه دوم ورودی موقعیت عکس دست پیدا شده را دریافت و مختصات ۲۱ نقطه عطف را موقعیت‌یابی می‌کند. به کمک این دو شبکه، می‌توان به طور همزمان موقعیت دست‌ها را تشخیص داد و نقاط عطف آن‌ها را پیش‌بینی کرد تا برای تشخیص ژست دست در پروژه‌های واقعیت افزوده و واقعیت مجازی استفاده شود.

### نتیجه بدست آمده

مدل‌های طراحی شده در این مقاله برای تشخیص نقاط عطف دست از دقت ۹۵.۷ درصد برخوردار هستند که دقت بسیار بالایی محاسبه می‌شود. این مدل به نور و تصویر پس‌زمینه وابسته نیست و دقت متوسط آن در زمینه‌های مختلف اندازه‌گیری شده، لذا مدل را کاربردی‌تر می‌کند.

<sup>12</sup>Augmented Reality

## ۵-۲ مقالات مربوط به آشنایی با پهپاد و اجرای مدل‌های بینایی

### کامپیوتر روی آن

#### ۱-۵-۲ مقاله Modeling Relation Among Implementing AI-Based Drones and Sustainable Construction Project Success

این مقاله به بررسی ارتباط بین استفاده از پهپادهای مبتنی بر هوش مصنوعی و موفقیت پروژه‌های ساخت‌وساز پایدار می‌پردازد. این تحقیق تأثیرات مثبت ادغام این دو عنصر را در بهبود کارایی و پایداری پروژه‌های ساختمانی بررسی می‌کند. با توجه به افزایش سریع استفاده از پهپادها در صنعت این مقاله به شناخت دقیقی از ارتباط بین این دو از جمله موانع و موفقیت‌ها می‌پردازد. این شناخت به شناسایی بهترین روش‌ها برای از بین بردن موانع و تضمین موفقیت پهپادها در ترکیب هوش مصنوعی کمک می‌کند [۲۰].

#### روش‌شناسی

در این تحقیق، از مدل معادلات ساختاری هوشمند برای بررسی موانع و موفقیت اجرای پهپادهای مبتنی بر هوش مصنوعی در صنعت استفاده شده است. این مدل شامل یک مجموعه جامع از عوامل موفقیت شامل کیفیت، ایمنی و عوامل محیطی است. پیاده‌سازی مدل‌ها بر روی پهپادها چالش‌هایی ایجاد می‌کند که باید مورد توجه قرار گیرد. این چالش‌ها شامل موارد زیر می‌شود:

- پیچیدگی فنی: پیاده‌سازی مدل‌های هوش مصنوعی بر روی پهپادها نیازمند دانش و تخصص فنی بالا است. این امر نیازمند همکاری بین متخصصان مختلف از حوزه‌های مختلف می‌باشد.
- محدودیت‌های سخت‌افزاری: پهپادها ممکن است دارای محدودیت‌های سخت‌افزاری مانند ظرفیت پردازشی و حافظه باشند که ممکن است موانعی برای پیاده‌سازی مدل‌های پیچیده ایجاد کنند.
- امنیت و حریم خصوصی: استفاده از هوش مصنوعی در پهپادها نیازمند رعایت استانداردهای امنیتی و حفظ حریم خصوصی است. این امر می‌تواند یک چالش مهم برای پیاده‌سازی مدل‌های هوش مصنوعی باشد.

- آموزش و توسعه مدل‌ها: پیاده‌سازی مدل‌های هوش مصنوعی بر روی پهپادها نیازمند آموزش و توسعه مدل‌های مناسب برای محیط و وظایف خاص پهپادها است.

این چالش‌ها نشان‌دهنده اهمیت اصلی پیاده‌سازی مدل‌های هوش مصنوعی بر روی پهپادها در صنعت است. با غلبه بر این چالش‌ها، می‌توان بهبود قابل توجهی در کارایی، ایمنی و پایداری این گونه پروژه‌ها داشت.

### نتیجه بدبست آمده

این مقاله نه تنها به شناخت بهتر موانع استفاده از پهپادهای مبتنی بر هوش مصنوعی کمک می‌کند، بلکه راهکارهایی برای غلبه بر این موانع و افزایش موفقیت این تکنولوژی در صنعت ساختمان ارائه می‌دهد. از آنجا که پهپادها می‌توانند به صورت خودکار و هوشمند وظایف مختلفی را انجام دهند، این تکنولوژی می‌تواند به بهبود مدیریت پروژه، کاهش هزینه‌ها و زمان اجرا، افزایش کیفیت و ایمنی کارها کمک کند.

## ۲-۵-۲ مقاله Use of A DJI Tello Drone as An Educational Platform in the Field of Control Engineering

این مقاله یک رویکرد نوآورانه را برای استفاده از پهپاد DJI Tello به عنوان یک پلتفرم آموزشی در زمینه مهندسی کنترل ارائه می‌دهد. این مقاله به بررسی نحوه استفاده از ویژگی‌های پهپاد برای آموزش مفاهیم کنترل به صورت عملی و جذاب و همچنین چگونگی استفاده از پهپاد DJI Tello به عنوان یک پلتفرم آموزشی برای مفاهیم کنترل می‌پردازد [۲۱].

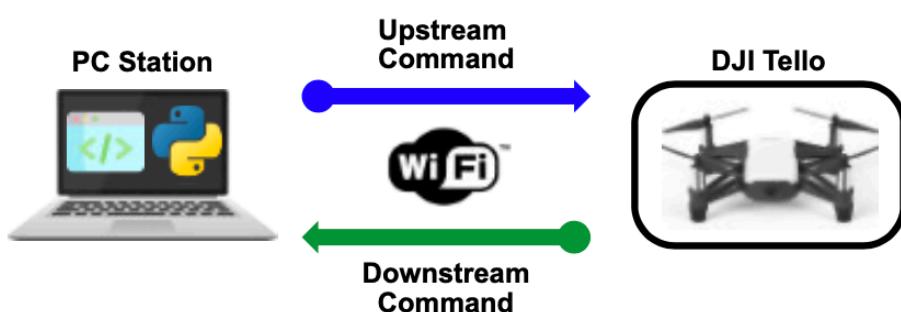
### روش‌شناسی

در این مقاله، از پهپاد DJI Tello به عنوان یک ابزار آموزشی و تحقیقاتی استفاده شده است. این پهپاد به دلیل داشتن حسگرهای متنوع و امکان برنامه‌نویسی با زبان پایتون، به عنوان یک ابزار ایده‌آل و انعطاف‌پذیر برای اهداف آموزشی و تحقیقاتی شناخته می‌شود. مقاله به بررسی ارتباط با پهپاد DJI Tello از طریق زبان برنامه‌نویسی پایتون، امکانات SDK رسمی ارائه شده توسط پهپاد، و نحوه ارتباط با پهپاد از طریق وای‌فای و پورت UDP می‌پردازد.

در این مقاله، انتخاب پهپاد DJI Tello برای نمایش مفاهیم ابتدایی مربوط به حوزه کنترل به دلایل گوناگونی انجام شده است. از جمله دلایل انتخاب این پهپاد، محبوبیت رو به افزایش آن در بین عموم

مردم، ویژگی‌های چندگانه آن شامل حسگرها، کنترل بازخورد و الگوریتم‌های تخمین برای انجام وظایف پیچیده، و همچنین قیمت مقرن به صرفه آن نسبت به سایر تجهیزات آموزشی است.

برای پیاده‌سازی مدل‌های هوش مصنوعی بر روی پهپاد DJI Tello، از زبان برنامه‌نویسی پایتون و کتابخانه‌های مختلفی که برای ارتباط با حسگرها و سیستم‌های کنترل پروازی پهپاد ارائه شده استفاده می‌شود. این چنین موارداین امکان را می‌دهد که مدل‌های هوش مصنوعی توانایی پیاده‌سازی بر روی پهپاد را داشته و عملکرد آن را تحلیل کنند. به عنوان مثال، کاربران می‌توانند از شبکه‌های عصبی یا منطق فازی برای کنترل حرکت پهپاد استفاده کنند.



شکل ۲-۸- ارتباط برنامه پایتون با پهپاد DJI Tello

### نتیجه بدست آمده

نتیجه این مقاله نشان می‌دهد که استفاده از هوش مصنوعی بر روی پهپاد DJI Tello به خوبی عمل می‌کند و این ابزار آموزشی و تحقیقاتی می‌تواند به افراد کمک کند تا مفاهیم پیچیده کنترل و هوش مصنوعی را به صورت عملی و جذاب فرا بگیرند.

## ۶-۲ جمع‌بندی

در این فصل به بررسی ۴ گونه مقالات پرداخته شد

- مقالاتی که با استفاده از استخراج ویژگی‌های تصویر از جمله نور، جهت، موقعیت و این چنین موارد توانستند ژست دست را تشخیص دهند. این مقالات برای کلاس‌بندی تعداد ژست‌های محدود و گوناگون از یکدیگر به خوبی عمل می‌کنند. در زمانی که تعداد ژست‌ها زیاد شده و حالات دست نزدیک به هم باشند، این دقت به صورت چشم‌گیری کاهش پیدا می‌کند. از آنجایی که

تمرکز پژوه ما تعیین ژست دست بدون محدودیت تعداد و حالت است این راهکار به خوبی عمل نمی‌کند.

• مقالاتی مربوط به ورودی تصویر دست، نه تنها دقت بالایی از خود نشان نمی‌دهند، بلکه زمان اجرای آنها نسبت به دیگر راهکارها بسیار بیشتر است و می‌تواند پژوه را از زمان واقعی بودن آن که یکی از بزرگ‌ترین اهداف است دور کند.

• مقالات مربوط به تعیین ژست دست با استفاده از نقاط کلیدی عملکرد بسیار خوب و دقت بالایی را از خود نشان داد. این مقالات با توجه به اینکه از چندین مدل پی در پی برای تعیین ژست کمک می‌گیرند، اما مدل‌ها حجم پایینی دارند لذا در زمان کوتاهی می‌توان ژست دست را پیش‌بینی کرد. این مقالات راهکارهای مناسبی را برای کمک پیاده‌سازی پژوه ما ارائه دادند.

• مقالات مربوط به آشنایی با پهپاد و اجرای مدل‌های بینایی کامپیوتر روی آن دیدگاه جامعی در مورد پهپاد، مشخصات مورد نیاز آن و چالش‌ها برای اجرای این پژوه به ما نشان دادند.

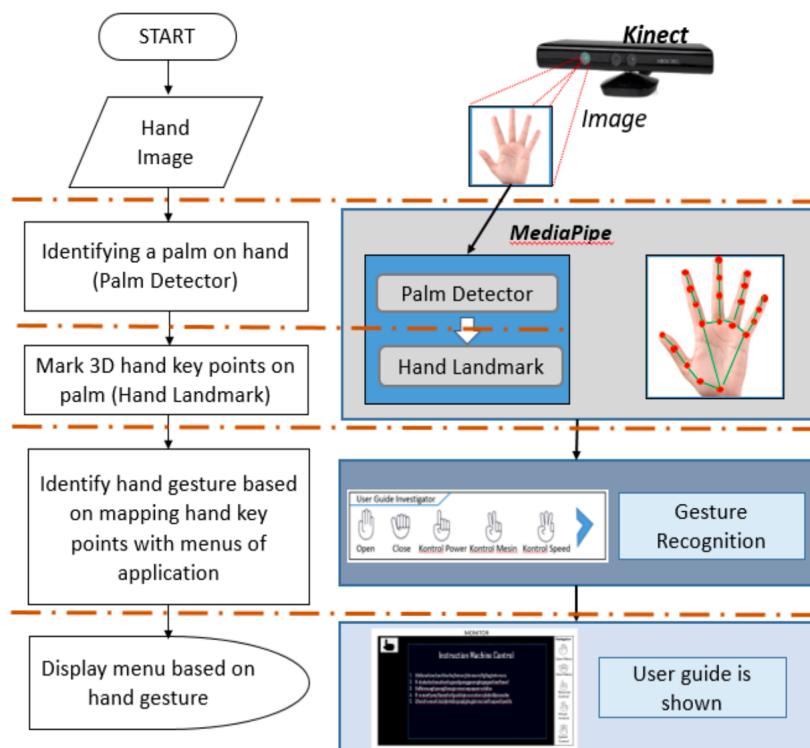
در تمام مقالات بررسی شده، پژوه‌هایی با هدف تعیین ژست دست به گونه‌ای پیاده‌سازی شده‌اند تا حرکات را به گونه‌ای کلاسیبندی کنند که در خروجی حتماً یکی از ژست‌های درنظر گرفته‌شده انتخاب شود. لذا زمانی که دست در حالتی غیر از آنها قرار دارد، مدل طراحی شده حتماً یکی از ژست‌هایی را که به آن شبیه‌تر است انتخاب می‌کند که این امر می‌تواند برای پیاده‌سازی روی پهپاد واقعی مشکل‌زا باشد و حتی هزینه مالی به ارمغان آورد. برای پیاده‌سازی پژوه ما باید مسیری را در پیش بگیریم تا بتوانیم این مشکل را برطرف کرده و در عین حال دقت بالایی داشته باشیم.

# فصل سوم

## روش انجام پروژه

### ۱-۳ مقدمه

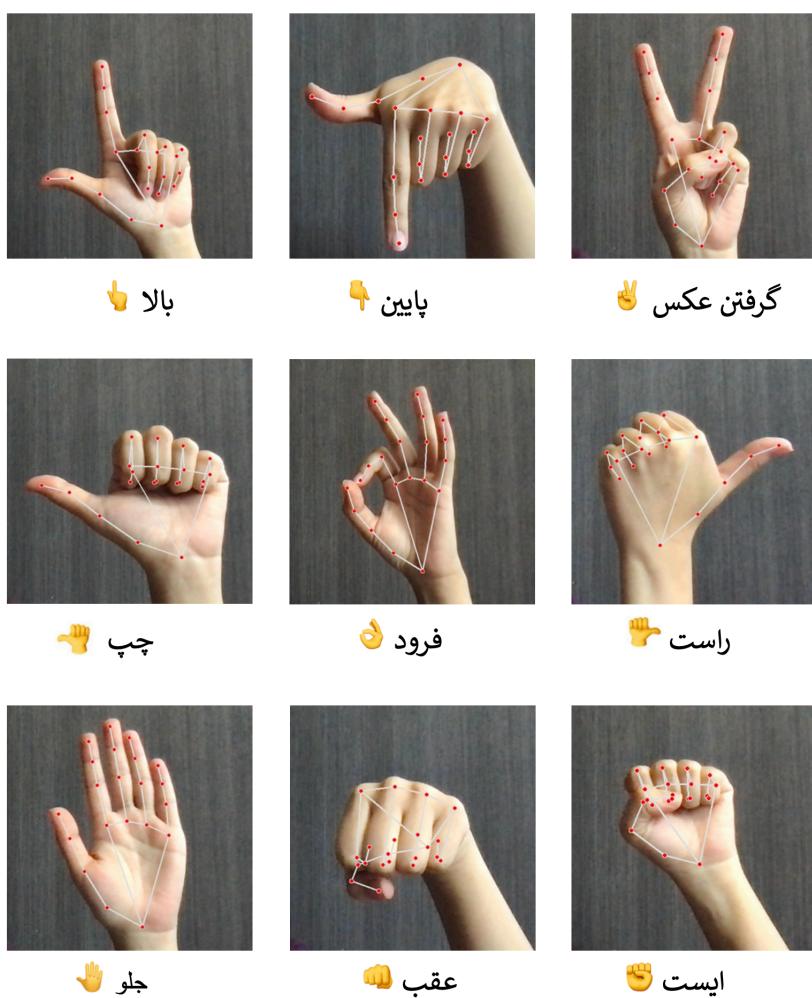
برای اجرای این پروژه راههای متفاوتی مورد بررسی قرار گرفته شد تا بتوان آنها را روی پهپاد پیاده‌سازی کرد. نتیجه نهایی پیاده‌سازی ۳ شبکه کانولوشن به صورت پی در پی است. شبکه اول برای آشکارسازی موقعیت کف دست است. بدین صورت که هر فریم گرفته شده از دوربین پهپاد به ورودی مدل داده می‌شود و پس از پردازش آن خروجی یک ماتریس  $3 \times 256 \times 256$  است که جعبه محدود کننده دست را شامل می‌شود. پس از آن، این ماتریس به مدل دوم به عنوان ورودی داده می‌شود، در این مدل جعبه مرزی برش خورده دست به صورت یک ماتریس گرفته شده و پس از پردازش، خروجی مدل برابر ۲۱ نقطه سه بعدی عطف دست و شاخص دست (راست یا چپ) است. در ادامه مدل سوم به عنوان ورودی، یک ماتریس  $2 \times 21$  می‌گیرد که مختصات نقاط طول و عرض هر نقطه عطف دست است چرا که عمق تصویر با توجه به ژست‌های در نظر گرفته شده از اهمیت بالایی برخوردار نیست. و در خروجی پیش‌بینی می‌کند که کدام ژست دست مدنظر کاربر است. با توجه به این پروژه ۹ ژست گوناگون مدنظر قرار گرفته شده (کاربر می‌تواند ژست‌های جدیدی اضافه کند)، خروجی شبکه کانولوشن شامل ۱۰ کلاس کلاس است که ۹ کلاس برای ژست‌ها و کلاسی برای زمانی که هیچ کدام از ژست‌ها انتخاب نشده در نظر گرفته شده.



شکل ۱-۳ تشخیص ژست دست با کمک نقاط کلیدی دست

## ۲-۳ انتخاب ژست‌های دست متناسب با حرکت پهپاد

انتخاب ژست‌های مناسب برای هر یک از حرکات پهپاد از اهمیت ویژه‌ای برخوردار است. چرا که ژست‌هایی که از لحاظ مفهومی به عملکرد پهپاد شبیه هستند راحت‌تر به خاطر سپرده شده و تجربه دلپذیرتری را در کاربر به وجود می‌آورند. ما برای این پروژه ۹ ژست دست را در نظر گرفته‌ایم تا بتوان حرکات پایه پهپاد را با آنها انجام داد. این حرکات شامل: حرکت رو به جلو، حرکت رو به عقب، حرکت به پایین، حرکت به بالا، حرکت به راست، حرکت به چپ، فرود آمدن، ایستادن در موقعیت کنونی و گرفتن عکس است که نمونه ژست دست آنها با توجه به حرکت پهپاد نشان داده شده است.



شکل ۲-۳ نمونه‌ای از ژست‌های انتخاب شده در مجموعه داده‌ها

### ۳-۳ دیتاست

برای جمع‌آوری دیتاست مناسب پروژه، از آنچایی که ژست‌های دست بر اساس عملکرد پهپاد تعیین شدند تا استفاده از آنها برای کاربر مورد پسند باشد، پیدا کردن دیتاست آماده غیرممکن است. جمع‌آوری دیتاست جامع و مناسب از اهمیت بالایی برخوردار است زیرا مدل باید در زمان واقعی کار کند و نمی‌توان از لایه‌های زیادی استفاده کرد. بنابراین، برای افزایش دقت مدل باید از روش‌های دیگری نظیر پیش‌پردازش، پس‌پردازش و استفاده از دیتاست مناسب بهره گرفت.

رویکردهای متفاوتی برای این پروژه پیاده‌سازی شدند تا بهترین راهکار برای مدلی با حجم کم و در عین حال جامع یافت شود. در نتیجه، دیتاست‌های گوناگونی جمع‌آوری شدند که هر یک از آنها ویژگی خاصی از تصویر را به عنوان داده‌ی مورد نیاز جمع‌آوری می‌کردند. این رویکردها شامل استفاده از ورودی کل عکس به صورت پیکسل‌های رنگی، پیدا کردن دست و ذخیره موقعیت آن به صورت پیکسل‌های  $256 \times 256$  و پیدا کردن نقاط کلیدی دست و ذخیره موقعیت آنها در دیتاست بود. در نهایت، گزینه سوم به عنوان بهترین راه ممکن برای ذخیره داده‌ها و اجرای پروژه انتخاب شد.

برای این پروژه، به منظور امکان اضافه کردن ژست دست جدید توسط کاربر، کدی پیاده‌سازی شد که با باز شدن دوربین و زدن یک حرف یا کلمه یکسان برای هر کلاس، اطلاعات آن تصویر استخراج شده و در دیتاست ذخیره شود. این روش نه تنها جمع‌آوری داده را راحت‌تر می‌کند، بلکه به کاربر اجازه می‌دهد با زدن یک دکمه، کلاس جدیدی را ایجاد کند.

### ۴-۳ اهمیت ژست دست

هنگام صحبت کردن، افراد از ژست‌ها استفاده می‌کنند. ژست‌ها جزء اساسی زبان هستند که اطلاعات معنادار و منحصر به فردی را منتقل می‌کنند. این حرکات به گوینده کمک می‌کنند تا اهداف خود را بهتر منعکس کند و نقش‌های مهمی در ارتباط، یادگیری و درک ایفا می‌کنند، هم برای افرادی که آنها را مشاهده می‌کنند و هم برای کسانی که آنها را انجام می‌دهند.

به حرکات خود به خودی دست که در ریتم گفتار ایجاد می‌شوند، حرکات هم‌گفتاری<sup>۱</sup> گفته می‌شود. مردم از تمامی فرهنگ‌ها و پیشینه‌های زبانی شناخته شده برای ارتباط بهتر از حرکات هم‌گفتاری استفاده می‌کنند. حتی نوزادان قبل از بیان اولین کلمات خود، از انواع ژست‌ها استفاده می‌کنند. دست‌ها به ما

<sup>1</sup>Co-Speech Gestures

کمک می‌کند صحبت کنیم، فکر کنیم و به خاطر بسپاریم، گاهی اوقات دانشی را که هنوز نمی‌توان به زبان آورد، آشکار می‌کنند. به طوری که می‌توان گفت ژست‌ها اغلب به عنوان زبان گفتاری ثانویه در نظر گرفته می‌شوند [۲۲].

ژست‌ها بهویژه زمانی مؤثر هستند که مزیتی نسبت به کلمات داشته باشند [۲۳]. توانایی درک شکل و حرکت دست‌ها می‌تواند یک جزء حیاتی در بهبود تجربه کاربر<sup>۲</sup> در حوزه‌ها و پلتفرم‌های مختلف فناوری باشد. درک مفهوم ژست دست در زمان واقعی برای افراد به طور طبیعی وجود دارد، اما این کار زمانی که توسط بینایی کامپیوتری رخ دهد می‌تواند چالش‌برانگیز باشد. زیرا دست‌ها اغلب خود یا یکدیگر را مسدود می‌کنند مانند انسداد انگشت، کف دست و حتی لرزش دست نیز می‌تواند مشکلاتی به وجود آورد [۱۹].

### ۵-۳ کنترل پهپاد

اکثر پهپادهای تجاری موجود در بازار یا دارای کنترلهای ویژه طراحی شده هستند، یا از فرستنده‌های سیگنال اختصاصی و برنامه‌های نرمافزاری استفاده می‌کنند که روی دستگاه‌های دستی کاربران مانند تلفن‌های همراه یا تبلت‌ها اجرا می‌شوند. در هر دو حالت، کنترل‌کننده فرمان‌هایی را با اطلاعات دقیق از طریق کانال‌های بی‌سیم مانند وای‌فای یا بلوتوث ارسال می‌کند.

اخیراً محصولات تجاری معرفی شده‌اند که از حرکات دست به عنوان یک مکانیسم کنترل قابل اجرا استفاده می‌کنند. برای دریافت ژست‌ها، دو رویکرد اصلی وجود دارد:

- استفاده از دستکش‌های ویژه طراحی شده: کنترل‌کننده بر روی دستکشی که توسط کاربران استفاده می‌شود نصب می‌گردد و در زمان واقعی انحراف، گام و چرخش دست را شناسایی می‌کند تا حرکات مربوط به پهپاد را تشخیص و ارسال کند. از جمله این محصولات می‌توان به Kd MenKind Motion Control Drone و Interactive Aura Drone اشاره کرد.

- استفاده از بینایی کامپیوتر از طریق دوربین: این دستگاه‌ها از دوربین نصب شده روی پهپاد استفاده می‌کنند تا بتوانند در لحظه تشخیص دهنند که دست کاربر کجاست و در چه حالتی قرار دارد تا پهپاد را کنترل کنند. از جمله این محصولات می‌توان به DJI Spark Drone اشاره کرد.

<sup>2</sup>User Experience

## ۳-۶ ابزارها و نرم افزارهای مورد استفاده

برای پیاده‌سازی این پروژه از ابزارها، نرم افزارها و کتابخانه‌های گوناگونی استفاده شده است که در ادامه به توضیح دقیق آنها می‌پردازیم. قابل ذکر است که از کتابخانه‌هایی از جمله `Copy`, `Csv`, `Itertools` و `Os` نیز در قسمت‌هایی از پروژه به کار برده شده است که به دلیل استفاده جزئی توضیح داده نشده‌اند.

### ۱-۶-۳ کتابخانه TensorFlow

یک کتابخانه نرم افزاری رایگان و منبع باز برای یادگیری ماشین و هوش مصنوعی است. این کتابخانه توسط گوگل برین توسعه داده شده و می‌تواند در طیف وسیعی از وظایف یادگیری ماشین مورد استفاده قرار گیرد. همچنین تمرکز ویژه‌ای بر آموزش و استنتاج شبکه‌های عصبی عمیق دارد. TensorFlow انعطاف‌پذیری بالا دارد. می‌تواند برای انواع مختلف مدل‌های یادگیری ماشین از جمله شبکه‌های عصبی پیچشی، شبکه‌های عصبی بازگشتی و شبکه‌های مولد متخصص<sup>۳</sup> مورد استفاده قرار گیرد. TensorFlow قابلیت اجرای مدل‌ها بر روی پردازنده‌های چندگانه، پردازنده‌های گرافیکی<sup>۴</sup> و واحد پردازشی تنسور<sup>۵</sup> را دارد. همچنین به دلیل محبوبیت و پشتیبانی گسترده، منابع آموزشی و کتابخانه‌های جانبی فراوانی برای آن وجود دارد. به طور کلی، TensorFlow یکی از ابزارهای قدرتمند و پرکاربرد در حوزه یادگیری ماشین و هوش مصنوعی است [۲۴].

### ۲-۶-۳ کتابخانه Scikit-learn

که با نام‌های Scikit-learn و Scikits.learn نیز شناخته می‌شود یک کتابخانه یادگیری ماشین رایگان و منبع باز برای زبان برنامه‌نویسی پایتون است. این کتابخانه شامل الگوریتم‌های مختلفی برای طبقه‌بندی، رگرسیون و خوشه‌بندی مانند ماشین‌های بردار پشتیبان، جنگل‌های تصادفی، تقویت گرادیان، SciPy و NumPy و DBSCAN می‌باشد. Scikit-learn به طور ویژه برای تعامل با کتابخانه‌های K-means طراحی شده است و ابزارهای متنوعی برای پیش‌پردازش داده‌ها، انتخاب و ارزیابی مدل‌ها و کاهش ابعاد فراهم می‌کند. این کتابخانه به کاربران کمک می‌کند تا به راحتی از آن در پروژه‌های یادگیری ماشین استفاده کنند. این کتابخانه به دلیل سادگی و کارایی خود در بین محققان و مهندسان داده بسیار محبوب

<sup>3</sup>Generative Adversarial Network

<sup>4</sup>Graphics Processing Unit

<sup>5</sup>Tensor Processing Unit

است و امکانات وسیعی را برای توسعه و ارزیابی مدل‌های یادگیری ماشین فراهم می‌کند [۲۵].

### ۳-۶-۳ رابط برنامه‌نویسی Keras

Keras یک رابط برنامه‌نویسی<sup>۶</sup> یادگیری عمیق است که به زبان پایتون نوشته شده و می‌تواند بر روی JAX و PyTorch و TensorFlow اجرا شود. هدف اصلی Keras کاهش پیچیدگی‌ها و بار شناختی توسعه‌دهنده‌گان است، به طوری که آنها بتوانند روی بخش‌های حیاتی و مهم پروژه‌های یادگیری ماشین تمرکز کنند. این رابط برنامه‌نویسی با رابط کاربری ساده و کاربرپسند، امکان توسعه سریع مدل‌های پیچیده را فراهم می‌کند. Keras عملکرد بالایی دارد و توسط سازمان‌های بزرگی نظیر ناسا، یوتیوب و Waymo برای تحلیل داده‌ها، بهبود الگوریتم‌های توصیه‌گر و توسعه سیستم‌های خودران مورد استفاده قرار می‌گیرد. این کتابخانه با مستندات جامع و پشتیبانی از جامعه کاربری بزرگ، به یکی از ابزارهای محبوب در حوزه یادگیری عمیق تبدیل شده‌است [۲۶].

### ۴-۶-۳ کتابخانه‌های MediaPipe

MediaPipe مجموعه‌ای از کتابخانه‌ها و ابزارهایی است که از تکنیک‌های هوش مصنوعی و یادگیری ماشین در برنامه‌های خود استفاده می‌کند. این کتابخانه برای برنامه‌نویسان یادگیری ماشین از جمله محققان، دانشجویان و توسعه‌دهنده‌گان نرم‌افزار، که برنامه‌های کاربردی یادگیری ماشین را پیاده‌سازی می‌کنند، نمونه‌های اولیه فناوری را طراحی می‌کند تا پروژه‌ها را تا حد امکان ساده کند.

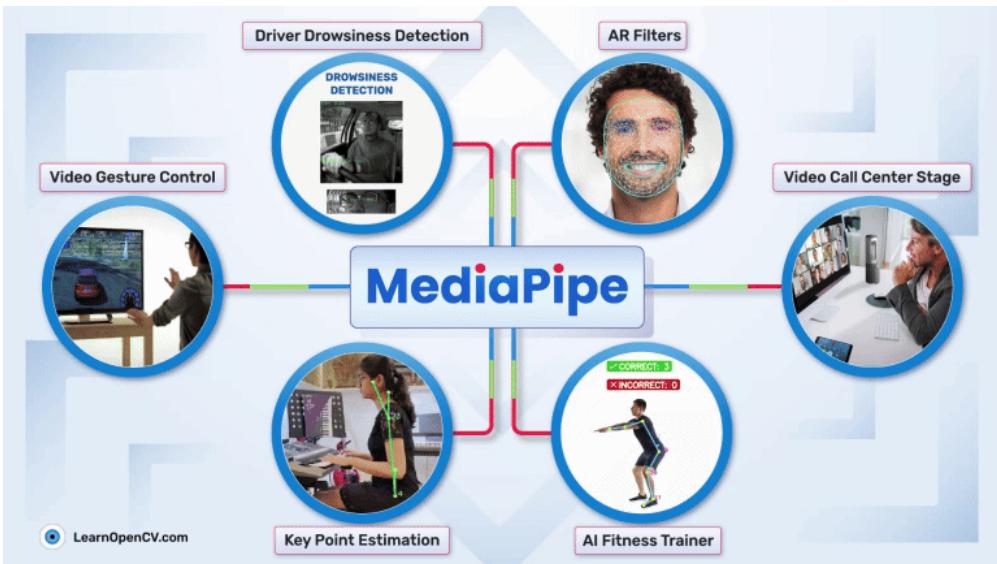
برنامه‌هایی که داده‌های حسی مانند ویدیو و صدا را با نرخ فریم بالا پردازش می‌کنند، به‌طور خاص برای بهبود تجربه کاربر طراحی شده‌اند. مراحل پردازش یا مدل‌های استنتاجی ممکن است پیچیده باشند، زیرا گاهی اتصال بین مراحل بسیار زیاد است. همچنین، توسعه برنامه برای پلتفرم‌های مختلف زمان بر است [۲۷].

این چالش‌ها را با انتزاع و اتصال مدل‌های مختلف به یکدیگر در یک چارچوب مناسب حل می‌کند. با استفاده از MediaPipe، می‌توان یک لوله پردازش را به صورت گراف از اجزای مختلف، از جمله مدل‌های استنتاجی و عملکردهای پردازش رسانه‌ای، ساخت. این کتابخانه همچنین قابل سفارشی‌سازی است و می‌تواند بر روی پلتفرم‌های مختلف توسعه یابد.

در مجموعه MediaPipe نیز از کتابخانه‌های مختلفی برای پیاده‌سازی برنامه‌ها استفاده می‌شود. از جمله

<sup>6</sup>Application Programming Interface

آنها می‌توان به TensorFlow، PyTorch و MXNet اشاره کرد [۲۸].



شکل ۳-۳ برخی کاربردهای کتابخانه MediaPipe

### ۵-۶-۳ کتابخانه NumPy

کتابخانه‌ای برای محاسبات علمی در پایتون است که آرایه‌های چندبعدی و توابعی برای عملیات سریع روی آرایه‌ها ارائه می‌دهد. در هسته‌ی NumPy، شیء ndarray وجود دارد که آرایه‌های n-بعدی با نوع داده‌ی همگن را در بر می‌گیرد و بسیاری از عملیات‌های ریاضی در آن انجام می‌شوند. این کتابخانه امکان انجام عملیات ریاضی پیشرفته و سایر عملیات‌ها روی تعداد زیادی داده را با کارایی بالا فراهم می‌کند، که با استفاده از دنباله‌های پایتون معمولی کارآمدی و کد کمتری دارند [۲۹].

### ۶-۶-۳ کتابخانه Matplotlib

Matplotlib یک کتابخانه متقابل پلتفرم<sup>7</sup>، برای تجسم داده‌ها و نمودارهای گرافیکی از جمله هیستوگرام، نمودارهای پراکنده و نمودار میله‌ای برای پایتون است. توسعه دهنده‌گان همچنین می‌توانند از رابطهای برنامه‌نویسی Matplotlib برای جاسازی نمودارها در برنامه‌های رابط کاربری گرافیکی استفاده کنند. یک اسکریپت matplotlib پایتون به گونه‌ای ساختار یافته است که چند خط کد تنها چیزی است که در بیشتر موارد برای تولید نمودار داده بصری مورد نیاز است. لایه برنامه‌نویسی Matplotlib دو رابط برنامه‌نویسی را پوشش می‌دهد:

<sup>7</sup>Cross-Platform

رابط برنامه‌نویسی Matplotlib.Pyplot، که سلسله مراتبی از اشیاء کد پایتون است که در بالای آن قرار دارد. و رابط برنامه‌نویسی اشیاء گرا<sup>۸</sup> که می‌تواند با انعطاف پذیری بیشتری نسبت به Pyplot مونتاژ شوند. این رابط برنامه‌نویسی دسترسی مستقیم به لایه‌های Backend Matplotlib را فراهم می‌کند [۳۰].

### ۷-۶-۳ کتابخانه OpenCV

OpenCV یک کتابخانه متن باز برای بینایی کامپیوتری و یادگیری ماشین است که برای فراهم کردن زیرساخت مشترک برای برنامه‌های بینایی کامپیوتری و تسريع استفاده از ادراک ماشین در محصولات تجاری طراحی شده است. این کتابخانه شامل بیش از ۲۵۰۰ الگوریتم بهینه‌سازی شده است که مجموعه جامعی از الگوریتم‌های کلاسیک و جدید بینایی کامپیوتری و یادگیری ماشین را فراهم می‌کند. OpenCV به طور گسترده‌ای در شرکت‌ها، گروه‌های تحقیقاتی و نهادهای دولتی برای انجام پروژه‌های بینایی کامپیوتری و یادگیری ماشین استفاده می‌شود. این کتابخانه رابطه‌ای برنامه‌نویسی متعددی از جمله C++, MATLAB و Java Python، C++ دارد که امکان انجام پروژه‌های بینایی با استفاده از زبان‌های برنامه‌نویسی مختلف را فراهم می‌کند [۳۱].

### ۸-۶-۳ پهپاد DJI Tello

DJI Tello یک پهپاد کوادکوپتر کوچک و قابل برنامه‌ریزی است که برای مصارف آموزشی و تست پروتوتایپ توسط DJI طراحی شده است. این پهپاد دارای ویژگی‌های ویژه‌ای مانند حرکات پایه‌ای کوادکوپتری و همچنین تکنولوژی کنترل پرواز DJI و یک پردازنده Intel بسیار قوی است. دوربین ۵ مگاپیکسلی این پهپاد امکان ضبط ویدیو با کیفیت خوب را فراهم می‌کند. همچنین، پهپاد دارای یک سیستم موقعیت‌یابی بصری<sup>۹</sup> است که شامل یک دوربین و یک مژول مادون قرمز ۳ بعدی است و قادر است در فواصل ۰.۳۰ متر تا ۳۰ متر ارتفاع کار کند.

هسته پهپاد به عنوان مرکز پردازشی و کنترلی آن عمل می‌کند و از یک پردازنده Intel قدرتمند پشتیبانی می‌کند. پهپاد DJI Tello برای اجرای پروژه‌های هوش مصنوعی مانند تشخیص اشیاء روی پهپاد، از زبان برنامه‌نویسی پایتون و SDK مربوطه پشتیبانی می‌کند. این SDK به کاربران این امکان را می‌دهد که پروتوتایپ‌های پایه‌ای پروژه‌های خود را توسعه دهند و آن‌ها را بر روی پهپاد اجرا کنند.

<sup>8</sup>Object Oriented

<sup>9</sup>Vision Positioning System

این پهپاد دارای باتری با جزئیات خاصیت مانند زمان پرواز و زمان شارژ است که می‌تواند از نظر عملکرد و ماندگاری باتری نسبت به پهپادهای دیگر مزیت داشته باشد. همچنین، پروژه‌های هوش مصنوعی که روی پهپاد DJI Tello پیاده‌سازی می‌شوند، می‌توانند شامل تشخیص اشیاء، پیش‌بینی حرکت‌ها و یا حتی خودکارسازی فرآیندهای پروازی باشند. از جمله مدل‌های هوش مصنوعی که می‌تواند روی پهپاد DJI Tello پیاده‌سازی شود، می‌توان به YOLOv3 اشاره کرد که برای تشخیص اشیاء با دقت بالا استفاده می‌شود [۳۲].

Weight	87 g
Dimensions	98×92.5×41 mm
Propeller	3 inches
Integrated Functions	Telemetric sensor
	Barometer
	LED
	Vision System
	Wi-Fi 2.4 GHz 802.11n
	Real-time streaming 720p
	USB battery charging port
Port	from 0° to 40°
Operating temperature range	from 2.4 to 2.4835 GHz
Transmitter (EIRP)	20 dBm (FCC)
	19 dBm (CE)
	19 dBm (SRRC)

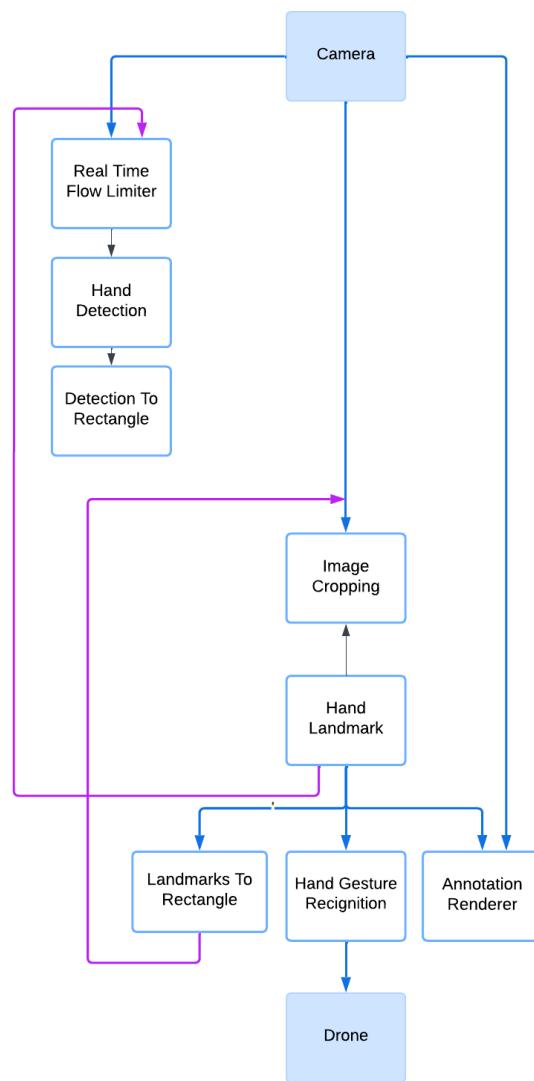
شکل ۴-۳ اطلاعات پهپاد DJI Tello

### ۷-۳ فلوچارت اجرای پروژه

در این پروژه از ۳ مدل اصلی به صورت متوالی برای پیش‌بینی تعیین ژست دست استفاده شده است. این سه مدل شامل تشخیص دست، تشخیص نقاط کلیدی دست و در نهایت تعیین ژست دست است. پس از استفاده از این مدل‌های بینایی ماشین، دستور پیش‌بینی شده برای اجرا به پهپاد DJI Tello فرستاده می‌شود.

برای بهینه کردن روند این پروژه مأذول‌های گوناگونی از جمله محدود کننده جریان زمان واقعی، تشخیص به مستطیل، برش تصویر، نقاط عطف یه مستطیل و ارائه کننده حاشیه نویسی نیز پیاده‌سازی شده‌اند تا روند پیش‌بینی ژست دست را با سرعت و دقت بالاتری اجرا کنند.

<sup>10</sup>Hand Detection



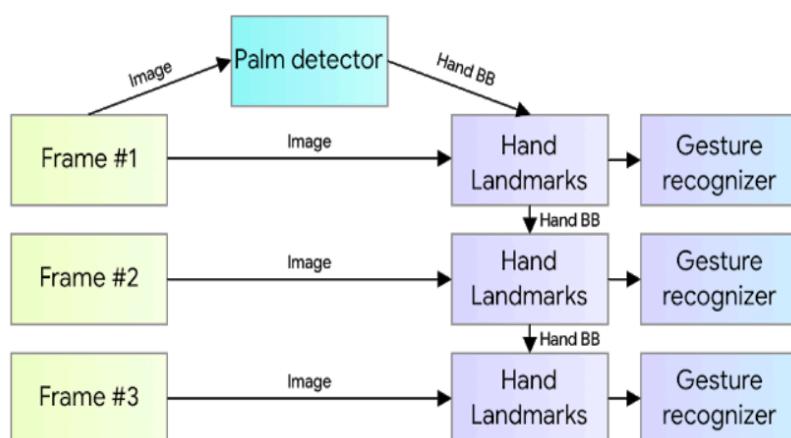
شکل ۳-۵ فلوچارت پروژه

### ۱-۷-۳ تشخیص دست<sup>۱۰</sup>

ماژول تشخیص کف دست، یکی از سه ماژول اصلی، با دقیق متوسط ۹۵.۷ درصد عمل می‌کند که این دقیق بالا با استفاده از استراتژی‌های مختلف به دست آمده است. این ماژول به جای تشخیص دست، از یک مدل تشخیص کف دست استفاده می‌کند زیرا تشخیص محدوده‌های اجسام سفت و سخت مانند کف دست و مشت بسیار ساده‌تر از تشخیص دست‌ها با انگشتان مفصلی است. از الگوریتم سرکوب غیر حداقلی برای حذف تشخیص‌های تکراری و انتخاب مرتبط‌ترین اشیاء شناسایی شده استفاده می‌شود که به کاهش مثبت کاذب و پیچیدگی محاسباتی کمک می‌کند. وظیفه اصلی این ماژول تشخیص دست در تصویر و محاسبه مکان دقیق دست است.

این مژول قادر است به صورت دقیق و کارآمد موقعیت دست‌ها را شناسایی کند و نواحی مربوطه را برای پردازش‌های بعدی فراهم کند. ورودی این مژول شامل تصویر یا فریم ویدیو و پارامترهای تنظیمات است، در حالی که خروجی آن شامل مستطیل‌های محدوده دست‌ها، نمرات اطمینان و در صورت فعال بودن، دست غالب<sup>۱۱</sup> است. معماری این مژول شامل مراحل پیش پردازش<sup>۱۲</sup>، مدل تشخیص دست و پس پردازش<sup>۱۳</sup> است که به ترتیب شامل نرم‌افزاری و تغییر اندازه تصویر، شبکه عصبی تشخیص دست و فیلترینگ و محاسبه نواحی مستطیلی دست‌ها می‌باشد.

وظیفه اصلی مژول شناسایی و محصور کردن دست‌ها در تصاویر و ویدیوها است که در کاربردهای مختلفی از جمله تشخیص حرکات دست، رابطه‌ای کاربری بدون لمس، تحلیل رفتار و ژست‌ها و کمک به افراد کم‌توان اهمیت دارد. این مژول به توسعه‌دهندگان این امکان را می‌دهد که به راحتی و با دقت بالا دست‌ها را در تصاویر و ویدیوها شناسایی کرده و از این اطلاعات برای پردازش‌های بعدی استفاده کنند.



شکل ۳-۶ خط لوله تشخیص دست

## ۲-۷-۳ تشخیص نقاط عطف دست<sup>۱۴</sup>

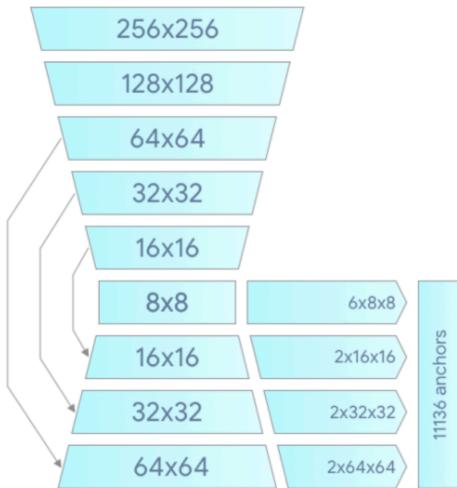
این مژول که دیگر اصلی است، یک ابزار قدرتمند برای تشخیص و ردیابی نقاط کلیدی دست است. این مژول وظیفه تشخیص و محاسبه نقاط عطف دست را بر عهده دارد. ورودی این مژول یک تصویر است که شامل دست یا دست‌هایی است که می‌خواهیم نقاط کلیدی آن‌ها را تشخیص دهیم. این تصویر باید

<sup>11</sup> Handedness

<sup>12</sup> Preprocessing

<sup>13</sup> Postprocessing

<sup>14</sup> Hand Landmark



شکل ۳-۷ معماری مدل آشکارساز کف دست

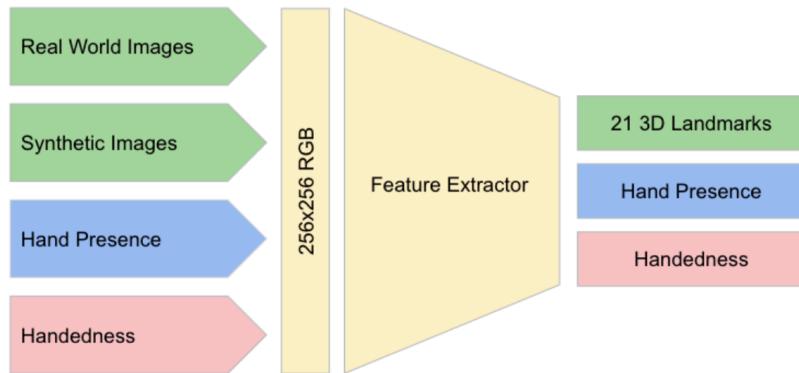
از پیش‌پردازش شده و دارای ناحیه‌ای باشد که دست در آن قرار دارد (جعبه مرزی) که توسط ماژول دیگری مانند تشخیص دست مشخص شده است.

خروجی این ماژول شامل موقعیت سه‌بعدی ۲۱ نقطه کلیدی دست است که شامل مفاصل انگشتان و نوک انگشتان می‌باشد. این نقاط کلیدی به صورت مجموعه‌ای از مختصات ( $x, y, z$ ) ارائه می‌شود که موقعیت هر نقطه را در فضای سه‌بعدی نشان می‌دهد و  $z$  نشان‌دهنده عمق نقطه نسبت به تصویر ورودی است.

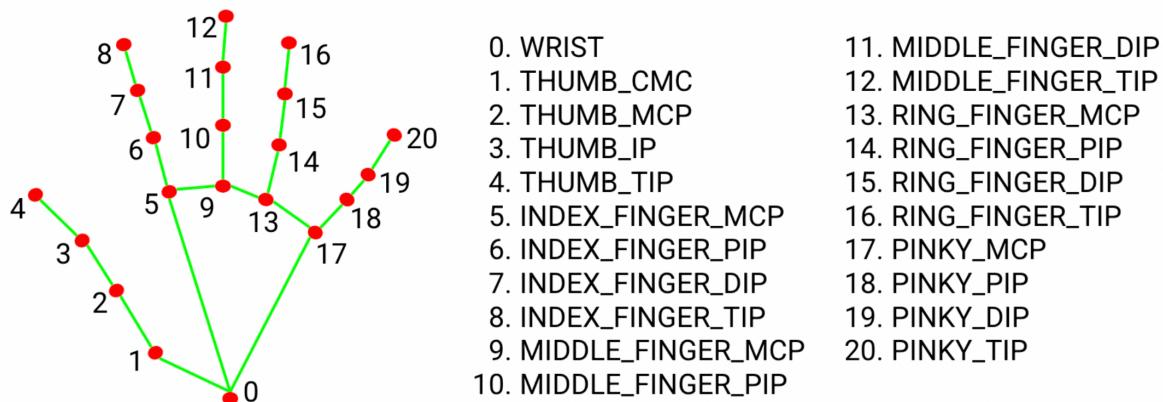
معماری این ماژول شامل استفاده از شبکه‌های عصبی عمیق برای تشخیص و ردیابی نقاط کلیدی دست است. این معماری شامل مراحل پیش‌پردازش برای تغییر اندازه و نرمال‌سازی تصویر، تشخیص دست برای شناسایی ناحیه‌های دست، مدل نقاط عطف برای تشخیص دقیق نقاط کلیدی دست و پس پردازش برای تبدیل مختصات نقاط کلیدی به فرمت خروجی نهایی و اعمال اصلاحات لازم است.

وظیفه اصلی این ماژول تشخیص و ردیابی دقیق نقاط کلیدی دست‌ها در تصاویر و ویدیوها است. این قابلیت می‌تواند در برنامه‌های مختلفی مانند واقعیت افزوده<sup>۱۵</sup>، رابطه‌ای کاربری بدون لمس، تحلیل حرکات، تشخیص حرکات دست در زبان اشاره و ... استفاده شود و توسعه‌دهندگان را قادر می‌سازد از این قابلیت‌های پیشرفته در برنامه‌های خود بهره ببرند [۱۹].

<sup>۱۵</sup>Augmented Reality



شکل ۸-۳ معماری مدل نقطه عطف دست. این مدل دارای سه خروجی است که یک استخراج کننده ویژگی را به اشتراک می‌گذارند. هر سر توسط مجموعه داده‌های مربوطه که با همان رنگ مشخص شده آنده آموزش داده می‌شود.



شکل ۹-۳ موقعیت ۲۱ نقطه کلیدی در ناحیه دست

### ۳-۷-۳ پیش‌بینی ژست دست

برای پیش‌بینی ژست دست از مدل‌های گوناگونی استفاده شده است که معماری و مشخصات هر یک به تفکیک توضیح داده شده‌اند. این معماری‌ها شامل شبکه پرسپترون چند لایه، شبکه عصبی کانولوشن، شبکه عصبی بازگشته و شبکه‌های حافظه کوتاه مدت بلند مدت است. دلیل استفاده از این مدل‌ها حجم کم و در عین حال دقیق بالای آنها است تا بتوان آنها را با یکدیگر قیاس و در نهایت بهترین مدل را برگزید. ورودی این مدل نقاط کلیدی دست و خروجی آن پس از پردازش شامل ۱۰ کلاس است. قابل ذکر است که کاربر توانایی اضافه کردن کلاس و در نتیجه ژست جدید را نیز در صورتی که پهپاد قابلیت انجام آن را داشته باشد دارد.

<sup>16</sup>Real Time Flow Limiter

### ۴-۷-۳ محدود کننده جریان زمان واقعی<sup>۱۶</sup>

ماژول محدود کننده جریان زمان واقعی وظیفه محدود کردن جریان پردازش به سرعت زمان واقعی را بر عهده دارد. این ماژول امکان کنترل پردازش داده‌ها را به صورت کارآمد و در زمان واقعی فراهم می‌کند، بدون ایجاد تأخیر یا بار اضافی بر سیستم. ورودی‌های این ماژول شامل جریان‌های بسته<sup>۱۷</sup> و مهر زمان‌ها<sup>۱۸</sup> می‌شود و خروجی‌های آن شامل جریان‌های بسته با نرخ محدود<sup>۱۹</sup> و اطلاعات مربوط به بسته‌های حذف شده ناشی از محدودیت نرخ پردازش است. معماری این ماژول شامل مراحل بافر ورودی<sup>۲۰</sup>، تجهیزه و تحلیل مهر زمانی<sup>۲۱</sup>، الگوریتم محدود کردن نرخ<sup>۲۲</sup> و بافر خروجی<sup>۲۳</sup> است که به منظور محدود کردن نرخ جریان داده‌ها و انجام پردازش در زمان واقعی طراحی شده است. این ماژول کمک می‌کند تا بار پردازش کاهش یابد، تأخیر کاهش یابد، کیفیت خدمات حفظ شود و منابع محاسباتی بهینه‌سازی شوند [۱۹].

### ۵-۷-۳ تشخیص به مستطیل<sup>۲۴</sup>

این ماژول وظیفه تبدیل نتایج تشخیص اشیاء به مستطیل‌های محدود کننده دست دارد. این ماژول ورودی‌هایی مانند تشخیص پرونوی<sup>۲۵</sup> و فریم تصویر<sup>۲۶</sup> را دریافت کرده و مستطیل‌های محدود کننده را به صورت نرم‌السازی شده و یا مطلق برای نواحی تشخیص داده شده تولید می‌کند. معماری این ماژول شامل مراحل پیش‌پردازش، تبدیل تشخیص<sup>۲۷</sup> و پس‌پردازش است. وظیفه اصلی این ماژول تبدیل نتایج تشخیص به مستطیل‌های محدود کننده است که در کاربردهای مختلف مانند تشخیص و ردیابی چهره، تشخیص دست، امنیتی و واقعیت افزوده کاربرد دارد [۱۹].

<sup>17</sup>Packet Streams

<sup>18</sup>Time Stamp

<sup>19</sup>Rate-Limited Packet Streams

<sup>20</sup>Input Buffering

<sup>21</sup>Timestamp Analysis

<sup>22</sup>Rate Limiting Algorithm

<sup>23</sup>Output Buffering

<sup>24</sup>Detection To Rectangle

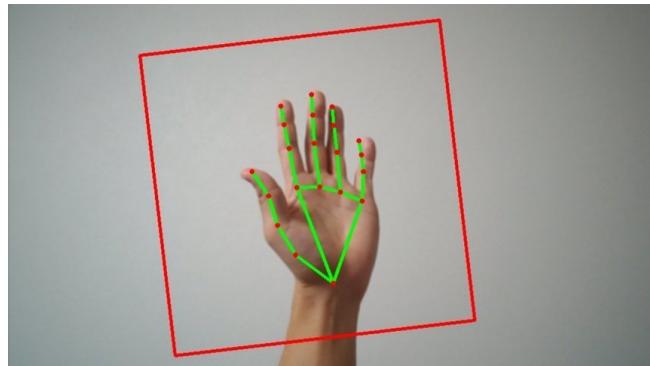
<sup>25</sup>Detection Proto

<sup>26</sup>Image Frame

<sup>27</sup>Detection Conversion

<sup>28</sup>Image Cropping

<sup>29</sup>Bounding Box



شکل ۳-۱۰ پیدا کردن مستطیل حاوی دست

### ۶-۷-۳ برش تصویر<sup>۲۸</sup>

ماژول برش تصویر برای برش و استخراج ناحیه‌های مورد نظر از تصاویر استفاده می‌شود. این ماژول ورودی‌هایی مانند تصویر اصلی و جعبه مرزی<sup>۲۹</sup> یا مختصات برش را می‌پذیرد و تصویر برش خورده حاوی ناحیه مورد نظر را تولید می‌کند. معماری این ماژول شامل مراحل پیش‌پردازش، عملیات برش<sup>۳۰</sup> و پس پردازش می‌باشد. وظیفه اصلی این ماژول برش دقیق ناحیه‌های مورد نظر از تصویر اصلی است که در کاربردهای مختلفی مانند پیش‌پردازش برای تشخیص چهره یا دست، تحلیل تصاویر پزشکی، واقعیت افزوده و پردازش تصویر در کاربردهای امنیتی مورد استفاده قرار می‌گیرد. این ماژول به توسعه‌دهنده‌گان این امکان را می‌دهد که ناحیه‌های خاصی از تصاویر را به صورت دقیق و مؤثر استخراج کرده و برای پردازش‌های بعدی آماده کنند [۱۹].

### ۷-۷-۳ نقاط عطف یه مستطیل<sup>۳۱</sup>

این ماژول برای تبدیل نقاط کلیدی به مستطیل‌های محدود‌کننده استفاده می‌شود. این ماژول وظیفه دقیق محصور کردن نقاط کلیدی یک شیء، مانند دست یا چهره، را بر عهده دارد. با استفاده از این مستطیل‌ها، می‌توان موقعیت دقیق‌تر اشیاء در تصویر را نمایش داد و از آنها به عنوان ورودی برای ماژول‌های بعدی در گراف استفاده کرد. این ماژول شامل مراحل پیش‌پردازش داده‌های ورودی، محاسبه مستطیل محدود‌کننده، نرمال‌سازی مختصات مستطیل‌ها و تولید خروجی نهایی می‌باشد. وظیفه اصلی این ماژول تبدیل نقاط کلیدی به مستطیل‌های محدود‌کننده دقیق است که در بسیاری از کاربردها اهمیت دارد، از جمله ردیابی اشیاء، تحلیل حرکات، پیش‌پردازش برای مدل‌های دیگر و افزایش دقت در

<sup>۳۰</sup>Cropping Operation

<sup>۳۱</sup>Landmarks To Rectangle

پردازش تصویر [۱۹].

### ۸-۷-۳ ارائه کننده حاشیه نویسی<sup>۳۲</sup>

این مازول برای نمایش گرافیکی نتایج پردازش‌های مختلف بر روی تصاویر یا ویدیوها استفاده می‌شود. این مازول اطلاعات حاشیه‌نویسی شامل نقاط کلیدی، مستطیل‌های محدود‌کننده، خطوط، متن و سایر اشکال گرافیکی را به صورت بصری بر روی تصاویر نمایش می‌دهد. وظیفه اصلی این مازول شامل نمایش بصری نقاط کلیدی، مستطیل‌های محدود‌کننده، خطوط و اتصالات، متن و توضیحات بر روی تصاویر یا ویدیوها است. این مازول به توسعه دهنده‌گان امکان می‌دهد تا به راحتی و به صورت بصری نتایج پردازش‌های خود را مشاهده کنند و از این طریق به بهبود و ارزیابی عملکرد مدل‌ها و الگوریتم‌های خود بپردازند [۱۹].

### ۹-۷-۳ مدل‌های کلاس‌بندی برای تعیین ژست دست

#### ۱۰-۷-۳ شبکه پرسپترون چند لایه

شبکه پرسپترون چند لایه<sup>۳۳</sup> نوعی شبکه عصبی مصنوعی است که از چندین لایه نورون تشکیل شده است. نورون‌ها در پرسپترون‌های چند لایه معمولاً از توابع فعال‌سازی غیرخطی<sup>۳۴</sup> استفاده می‌کنند که به شبکه اجازه می‌دهد الگوهای پیچیده در داده‌ها را بیاموزد. پرسپترون‌های چند لایه در یادگیری ماشین اهمیت بالایی دارند زیرا می‌توانند روابط غیرخطی در داده‌ها را یاد بگیرند و آنها را به مدل‌های قدرتمندی برای کارهایی مانند طبقه‌بندی، رگرسیون و تشخیص الگو تبدیل می‌کند.

پرسپترون چندلایه نوعی شبکه عصبی پیش‌خور<sup>۳۵</sup> است که از نورون‌های کاملاً متصل با نوع غیرخطی تابع فعال‌سازی تشکیل شده است و برای تشخیص داده‌هایی که به صورت خطی قابل تفکیک نیستند استفاده می‌شود.

پرسپترون‌های چند لایه به طور گسترده در زمینه‌های مختلف از جمله تشخیص تصویر، پردازش زبان طبیعی و تشخیص گفتار و سایر موارد استفاده شده‌اند. انعطاف پذیری آنها در معماری و توانایی تقریب هر عملکرد تحت شرایط خاص آنها را به یک بلوک اساسی در یادگیری عمیق و تحقیقات شبکه عصبی

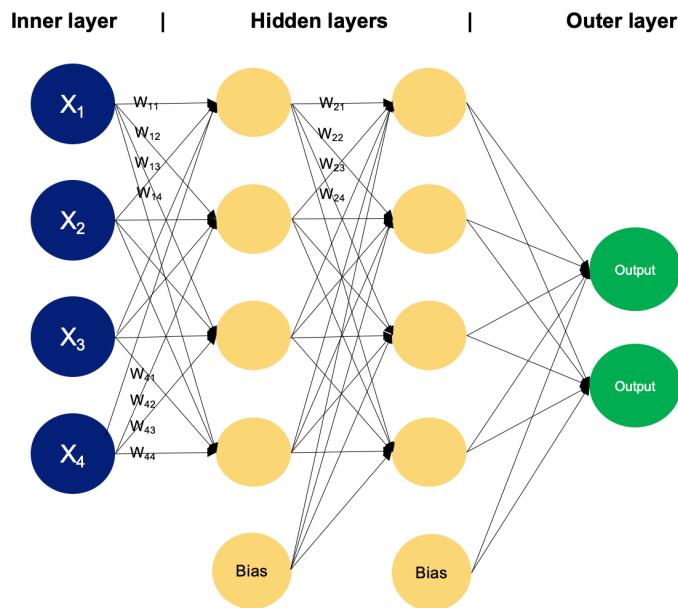
<sup>32</sup>Annotation Renderer

<sup>33</sup>Multilayer Perceptron

<sup>34</sup>Nonlinear Activation Functions

<sup>35</sup>Feedforward

تبدیل می‌کند.



شکل ۱۱-۳ نمونه‌ای از پرسپترون‌های چند لایه دارای دو لایه پنهان

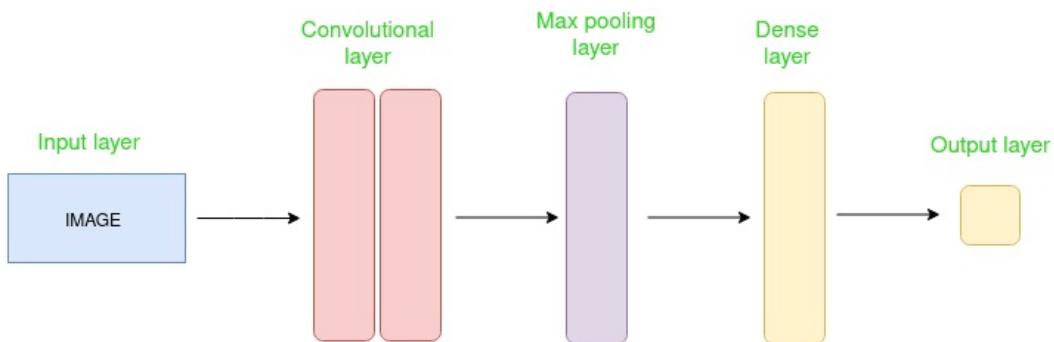
### ۱۱-۷-۳ شبکه عصبی کانولوشنال

شبکه عصبی کانولوشنال<sup>۳۶</sup> یک الگوریتم یادگیری عمیق است از جمله کاربردهای آن تشخیص اشیا مانند طبقه بندی، تشخیص و تقسیم بندی تصویر بسیار مهم است. بسیاری از برنامه‌های کاربردی مانند اتومبیل‌های خودران، دوربین‌های نظارتی و موارد دیگر از شبکه عصبی کانولوشنال استفاده می‌کنند. برخلاف مدل‌های سنتی یادگیری ماشینی مانند ماشین بردار پشتیبانی و درخت‌های تصمیم که نیاز به استخراج دستی ویژگی‌ها دارند، شبکه‌های عصبی کانولوشنال می‌توانند استخراج خودکار ویژگی‌ها را در مقیاس انجام دهند و آنها را کارآمد کند. این شبکه‌ها می‌توانند الگوها را از داده‌ها تشخیص دهند و ویژگی‌ها را بدون توجه به موقعیت آن‌ها، اعم از چرخش، مقیاس یا جابجایی تصویر استخراج کنند. معماری شبکه‌های عصبی کانولوشنال سعی می‌کند ساختار نورون‌ها را در سیستم بینایی انسان متشكل از چندین لایه تقلید کند، جایی که هر یک مسئول تشخیص یک ویژگی خاص در داده‌ها است. همانطور که در تصویر بیب نیز نشان داده شده است، شبکه عصبی کانولوشن از چندین لایه مانند لایه ورودی، لایه کانولوشن، لایه ادغام<sup>۳۷</sup> و لایه‌های کاملاً متصل تشکیل شده است.

<sup>36</sup>Convolutional Neural Network

<sup>37</sup>Pooling

لایه کانولوشنال فیلترهایی را روی تصویر ورودی اعمال می‌کند تا ویژگی‌ها را استخراج کند، لایه ادغام ابعاد تصویر را کاهش می‌دهد تا محاسبات را سریع‌تر و کمتر کند، لایه کاملاً متصل پیش‌بینی نهایی را انجام می‌دهد. بدین صورت که شبکه فیلترهای بهینه را از طریق پس انتشار و نزول گرادیان می‌آموزد.



شکل ۱۲-۳ نمونه معماری شبکه عصبی کانولوشنال

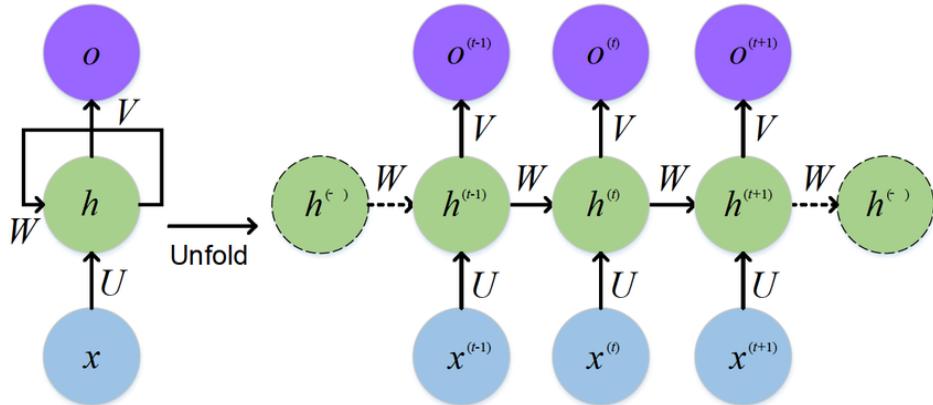
### ۱۲-۷-۳ شبکه عصبی بازگشتی

شبکه عصبی بازگشتی<sup>۳۸</sup> نوعی شبکه عصبی است که در آن خروجی مرحله قبل به عنوان ورودی به مرحله فعلی تغذیه می‌شود. در شبکه‌های عصبی سنتی، تمامی ورودی‌ها و خروجی‌ها مستقل از یکدیگر هستند. برای مثال در مواردی که پیش‌بینی مدل متکی به موارد قبلی است و نیاز است آنها را نیز مدنظر قرار دهد این شبکه بسیار کارآمد است. شبکه عصبی بازگشتی با کمک یک لایه پنهان این امکان را فراهم می‌کند. اصلی‌ترین و مهم‌ترین ویژگی شبکه عصبی بازگشتی حالت پنهان آن است که برخی از اطلاعات یک دنباله را به خاطر می‌سپارد. این حالت به عنوان حالت حافظه نیز شناخته می‌شود زیرا ورودی قبلی شبکه را به خاطر می‌آورد. و از پارامترهای یکسانی برای هر ورودی استفاده می‌کند زیرا وظیفه یکسانی را روی تمام ورودی‌ها یا لایه‌های پنهان برای تولید خروجی انجام می‌دهد. این برخلاف سایر شبکه‌های عصبی، پیچیدگی پارامترها را کاهش می‌دهد.

### ۱۳-۷-۳ شبکه‌های حافظه کوتاه مدت بلند مدت

شبکه عصبی بازگشتی یک حالت پنهان دارد که در طول زمان منتقل می‌شود، که می‌تواند یادگیری وابستگی‌های طولانی مدت را برای شبکه دشوار کند. شبکه‌های حافظه کوتاه مدت بلند مدت این مشکل

<sup>38</sup>Recurrent Neural Network



شکل ۳-۱۳ نمونه معماری شبکه عصبی بازگشتی

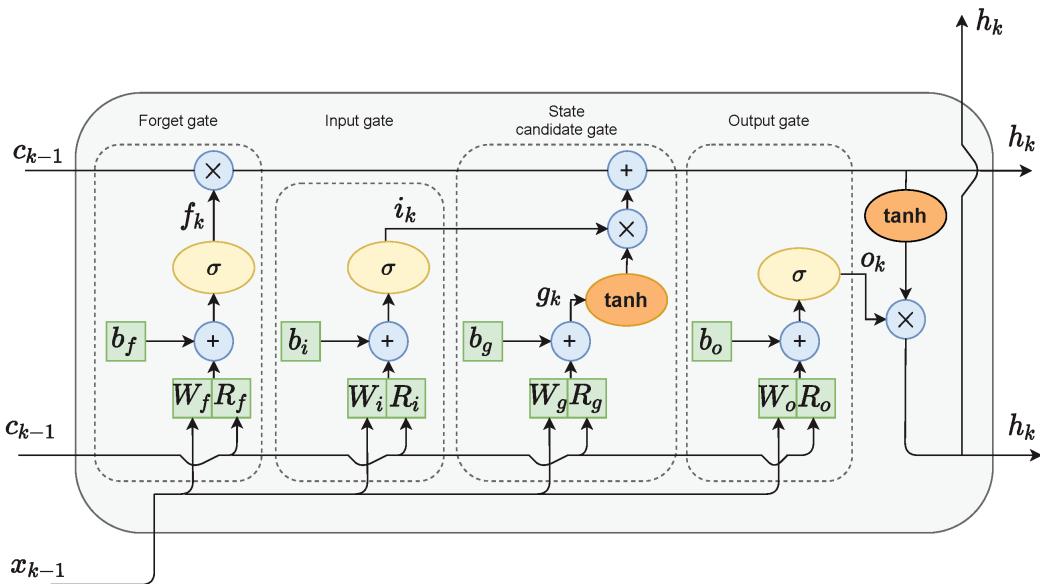
را با معرفی یک سلول حافظه، که محفظه ای است که می‌تواند اطلاعات را برای مدت طولانی نگهداری کند، برطرف می‌کند. شبکه‌های <sup>۳۹</sup> قادر به یادگیری وابستگی‌های طولانی مدت در داده‌های متوالی هستند، که آنها را برای کارهایی مانند ترجمه زبان، تشخیص گفتار و پیش‌بینی سری‌های زمانی مناسب می‌سازد. شبکه‌های حافظه کوتاه مدت بلند مدت همچنین می‌توانند در ترکیب با دیگر عماری‌های شبکه عصبی، مانند شبکه‌های عصبی کانولوشن برای تجزیه و تحلیل تصویر و ویدئو استفاده شوند.

سلول حافظه توسط سه گیت کنترل می‌شود: گیت ورودی، دروازه فراموشی و گیت خروجی. این گیت‌ها تصمیم می‌گیرند که چه اطلاعاتی را به سلول حافظه اضافه، حذف کرده و از آن خروجی بگیرند. گیت ورودی کنترل می‌کند که چه اطلاعاتی به سلول حافظه اضافه می‌شود. دروازه فراموشی کنترل می‌کند که چه اطلاعاتی از سلول حافظه حذف می‌شود. و گیت خروجی کنترل می‌کند که چه اطلاعاتی از سلول حافظه خروجی می‌شود. این به شبکه‌های حافظه کوتاه مدت بلند مدت اجازه می‌دهد تا به‌طور انتخابی اطلاعات را در جریان جریان در شبکه حفظ یا کنار بگذارند، که به آنها امکان می‌دهد وابستگی‌های طولانی مدت را بیاموزند.

۸-۳ پیش‌پردازش

برای ورود نقاط عطف دست به مدل تعیین ژست به ۲۱ مختصات طول و عرض نیاز داریم. خروجی مدل تعیین مختصات نقاط عطف دست برابر مختصات مطلق پیکسل‌ها نسبت به گوشه سمت چپ پایین تصویر است. این نقاط با توجه به اندازه عکس می‌توانند گستردگی باشند برای مثال در یک عکس با اندازه ۲۰۴۸\*۲۰۴۸ این اعداد از بین ۰ تا ۲۰۴۸ متغیرند. اگر این مختصات را به صورت مستقیم به مدل

<sup>39</sup>Long Short-Term Memory



شکل ۱۴-۳ نمونه معماری شبکه حافظه کوتاه مدت بلند مدت

تعیین ژست دست بدھیم دقیق مدل برابر ۸۷ درصد خواهد بود که به میزان کافی مورد قبول نیست.  
برای بهبود آن باید پیشپردازش‌هایی بر روی داده ورودی انجام شود.

از جمله این پیشپردازش‌ها می‌توان به نسبی کردن و نرمال‌سازی داده‌ها اشاره کرد. برای این کار ابتدا باید یک مرجع واحد در نظر گرفت تا نقاط، نسبت به آن مشخص شوند. در این پروژه ما مرجع را نقطه مشخص شده روی مج در نظر می‌گیریم. مختصات نقطه مرجع را برابر  $(0, 0)$  قرار می‌دهیم. سپس نسبت به آن و با توجه به فرمول زیر مختصات نقاط دیگر را به روز رسانی می‌کنیم.

$$X_{\text{rel}} = X_{\text{ref}} - X$$

پس از به نسبی کردن نقاط نسبت به مبدأ، آنها را با کمک فرمول زیر نرمال‌سازی می‌کنیم تا تمام طول و عرض نقاط به عددی میان صفر و یک به روز رسانی شوند.

$$X_{\text{new}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

در انتها این این مختصات را به عنوان ورودی به شبکه تعیین ژست دست می‌دهیم. با توجه به اینکه معماری هیچ یک از مدل‌ها تغییر نکرد و تنها داده‌های مختصات به روز رسانی شدند، دقیق نهایی مدل به ۹۷ درصد افزایش پیدا کرد و پیشپردازش تاثیر بهسزایی در بهینه کردن پروژه داشت.

### ۹-۳ پسپردازش

با وجود اینکه دقت مدل پیاده‌سازی شده بالا است و عملکرد بسیار چشم‌گیری از خود نشان می‌دهد، در عین حال پیش‌بینی اشتباه مدل می‌تواند عواقب زیان‌باری را به ارمغان آورد، از تجربه ناپسند برای کاربر گرفته تا برخورد پهپاد به اجسام و هزینه مالی. لذا باید دقت انجام پروژه را از آنچه مدل پیش‌بینی می‌کند نیز بالاتر برد. برای این کار از رأی‌گیری پنجره‌ای استفاده کرده‌ایم. بدین صورت که متغیری را با توجه به FPS دوربین در نظر می‌گیریم بدین صورت که هر چه FPS دوربین پهپاد بیشتر باشد متغیر در نظر گرفته شده نیز بیشتر است، برای مثال در پروژه ما از آنجایی که دوربین پهپاد برابر ۳۰ FPS است ما این متغیر را ۱۰ قرار داده‌ایم. سپس حد مناسبی را نیز بین صفر تا یک قرار می‌دهیم که ما در پروژه آن را برابر ۷۰ قرار داده‌ایم. طبق این راه ما ۱۰ فریم متناوب گرفته شده از پهپاد را به مدل پیاده‌سازی شده می‌دهیم اما تنها در صورتی دستور پیش‌بینی شده را به پهپاد پهپاد می‌دهیم که حد ممکن را بدست آورند. برای مثال اگر ۷ یا بیشتر از ۱۰ حرکت پیش‌بینی شده، دستور حرکت رو به جلو باشد آنگاه به پهپاد دستور داده می‌شود تا به جلو حرکت کند. در غیر این صورت اگر کمتر از ۷ عدد از فریمهای یک ژست دست را پیش‌بینی نکنند، پهپاد در حالت قبلی خود باقی می‌ماند و دستوری به آن داده نمی‌شود.

### ۱۰-۳ جمع‌بندی

در این فصل قسمت‌های مختلف پیاده‌سازی شده پروژه با جزئیات کامل توضیح داده شد. از جمله آنها نرم‌افزارها و سخت‌افزارها، کتابخانه‌ها، پیش‌پردازش، پس‌پردازش و کد اصلی که شامل سه مدل به صورت پی در پی که دو تا از آنها با کمک کتابخانه MediaPipe و دیگری یک کلاس‌بندی با ۹ حالت دست است بود.

# فصل چهارم

## نتایج و ارزیابی

## ۱-۴ مقدمه

سیستم تشخیص حرکات دست به نقش مهمی در ایجاد تعامل کارآمد بین انسان و ماشین تبدیل شده است. پیاده‌سازی این سیستم با استفاده از تشخیص ژست دست، نوید وسیعی از کاربردها را در صنعت فناوری می‌دهد. در این پژوهه، معماری‌های گوناگونی مانند شبکه‌های عصبی پیچشی، شبکه‌های حافظه کوتاه‌مدت بلند، و شبکه‌های عصبی چندلایه مورد آزمایش قرار گرفتند تا بهترین پیاده‌سازی برای تشخیص حرکات دست انتخاب شود.

در این فصل، به بررسی نتایج به دست آمده از این پژوهه پیاده‌سازی شده می‌پردازیم و دقت و عملکرد سیستم در شرایط مختلف را ارزیابی می‌کنیم. همچنین، نقاط قوت و ضعف هر یک از معماری‌های مورد استفاده را تحلیل کرده و پیشنهاداتی برای بهبود سیستم ارائه خواهیم داد. هدف این فصل، ارائه یک تحلیل جامع از کارایی سیستم و شناخت دقیق‌تر از عواملی است که می‌توانند به ارتقاء عملکرد آن کمک کنند.

## ۲-۴ ارزیابی عملکرد مدل‌ها

این پژوهه با مدل‌های گوناگونی پیاده‌سازی شد تا بتوان بهترین آنها را برای نتیجه نهایی بر روی پهپاد اجرا کرد. معیارهای ارزیابی شامل دقت، صحت<sup>۱</sup>، فراخوانی<sup>۲</sup>، امتیاز F1 و تعداد نمونه‌ها برای هر یک از ژست‌ها می‌باشد.

معیارهای دقت، صحت، فراخوانی و امتیاز F1 معیارهای ضروری برای ارزیابی عملکرد مدل یادگیری ماشینی هستند. آنها هر دو مثبت کاذب و منفی کاذب را در نظر می‌گیرند و درک دقیقی از قابلیت‌های پیش‌بینی یک مدل ارائه می‌دهند. این ارزیابی دقیق با برجسته کردن نقاط قوت و ضعف خاص به اصلاح مدل کمک می‌کند.

## ۱-۲-۴ دقت

دقت مدل معیاری است که نشان می‌دهد یک مدل یادگیری ماشینی تا چه اندازه قادر به پیش‌بینی یا تصمیم‌گیری صحیح بر اساس داده‌ها است. این معیار به صورت نسبت مجموع مثبت و منفی واقعی به تعداد کل نمونه‌ها محاسبه می‌شود.

<sup>1</sup>Precision

<sup>2</sup>Recall

دقت، بصری ترین معیار عملکرد است و به نسبت مشاهدات پیش‌بینی شده صحیح به کل مشاهدات اشاره دارد. این معیار برای مقایسه عملکرد مدل‌های مختلف و ارزیابی اثربخشی یک مدل خاص برای یک وظیفه معین استفاده می‌شود. دقต زمانی مناسب است که توزیع کلاس‌ها متعادل باشد و هزینه‌های مثبت کاذب و منفی کاذب مشابه باشند [۳۳].

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$$

معماری مدل دوره ۳ دقت داده آموزش دقت داده تست			
۱	۷	۶	MLP
۲	۸	۷	CNN
۳	۸	۵	LSTM

جدول ۱-۴ جدول ارزیابی دقت مدل‌ها

## ۲-۲-۴ صحت

صحت، یک معیار آماری برای ارزیابی کیفیت یک مدل پیش‌بینی است. این معیار یکی از کلیدی‌ترین معیارها برای تعیین عملکرد یک مدل، به ویژه در وظایف طبقه‌بندی، محسوب می‌شود. صحت نسبت مثبت واقعی به مجموع مثبت‌های واقعی و مثبت‌های کاذب (نمونه‌هایی که به اشتباه به عنوان مثبت شناسایی شده‌اند) را نشان می‌دهد.

صحت بالا نشان‌دهنده این است که یک مدل در جلوگیری از مثبت‌های کاذب عملکرد خوبی دارد، به این معنا که نمونه‌های منفی را به عنوان مثبت طبقه‌بندی نمی‌کند. این امر به ویژه در برنامه‌هایی که هزینه مثبت کاذب بالا است، اهمیت دارد [۳۴].

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

### ۳-۲-۴ فراخوانی

فراخوانی، که به عنوان نرخ مثبت واقعی نیز شناخته می‌شود، معیاری است که نشان می‌دهد یک مدل یادگیری ماشینی چقدر قادر به تشخیص درست نمونه‌های مثبت از بین کل نمونه‌های یک کلاس خاص است.

فراخوانی زمانی استفاده می‌شود که به حداقل رساندن منفی‌های کاذب از اهمیت ویژه‌ای برخوردار باشد. این بدان معناست که در کاربردهایی که هزینه منفی‌های کاذب بالا است یا از دست دادن نمونه‌های مثبت واقعی ضرر زیادی به همراه دارد، فراخوانی اهمیت بیشتری پیدا می‌کند [۳۵].

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

### ۴-۲-۴ امتیاز F1

امتیاز F1 معیاری است که میانگین هارمونیک دقت و فراخوانی را محاسبه می‌کند. این امتیاز، که معمولاً به عنوان معیار ارزیابی در طبقه‌بندی باینری و چند کلاسه استفاده می‌شود، دقت و فراخوانی را در یک متريک واحد ادغام می‌کند تا ارزیابی جامعی از عملکرد مدل به دست دهد.

امتیاز F1 به ویژه زمانی مفید است که داده‌ها نامتعادل باشند، زیرا نه تنها تعداد پیش‌بینی‌های نادرست را در نظر می‌گیرد، بلکه نوع خطاهای - مثبت کاذب و منفی کاذب - را نیز مد نظر قرار می‌دهد. این امر باعث می‌شود که F1 معیار بهتری برای ارزیابی عملکرد مدل‌هایی باشد که با کلاس‌های نادر و یا داده‌های نامتعادل سروکار دارند [۳۶].

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

### ۵-۲-۴ گزارش معیارهای ارزیابی در مدل‌ها

در این بخش، نتایج ارزیابی مدل‌ها برای تشخیص ۹ ژست مختلف دست ارائه شده است.

<sup>4</sup>Confusion Matrix

		precision	recall	f1-score	support
0	0.96	0.96	0.96	24	
1	0.92	0.98	0.95	47	
2	1.00	1.00	1.00	23	
3	0.95	0.88	0.91	24	
4	0.91	1.00	0.95	21	
5	1.00	0.83	0.91	30	
6	0.93	1.00	0.96	13	
7	1.00	1.00	1.00	9	
8	1.00	1.00	1.00	21	
9	0.89	0.94	0.91	17	
accuracy				0.95	229
macro avg		0.96	0.96	0.96	229
weighted avg		0.95	0.95	0.95	229

شکل ۱-۴ معیارهای ارزیابی برای تشخیص ژست دست در مدل MLP

#### ۶-۲-۴ ماتریس درهم‌ریختگی<sup>۴</sup>

ماتریس درهم‌ریختگی یک ابزار مهم در ارزیابی عملکرد مدل‌های طبقه‌بندی است. این ماتریس نشان‌دهنده تعداد نمونه‌هایی است که به درستی به هر کلاس تشخیص یافته‌اند و همچنین تعداد نمونه‌هایی که به اشتباه به کلاس‌های دیگر تشخیص داده شده‌اند. به عبارت دیگر، هر سطر این ماتریس نشان‌دهنده کلاس واقعی است و هر ستون نشان‌دهنده کلاسی است که مدل پیش‌بینی کرده است. اگر خانه  $z$ ، این ماتریس، عدد  $n$  را داشته باشد، این به این معنا است که مدل  $n$  بار نمونه‌های کلاس  $z$  را به درستی به کلاس  $z$  تشخیص داده است.

ماتریس درهم‌ریختگی مفید است زیرا به ما اطلاعات دقیقی از عملکرد مدل در هر کلاس را می‌دهد. از این اطلاعات می‌توان برای ارزیابی عملکرد کلی مدل، شناسایی نقاط ضعف و قوت مدل، و بهبود آن استفاده کرد. همچنین، این ماتریس به ما امکان می‌دهد بررسی کنیم که آیا مدل ما به نسبت هر کلاس اشتباه می‌کند یا اشتباهات آن به کلاس‌های خاصی متمرکز شده‌اند [۳۷].

#### ۳-۴ نمودارهای دقت و خطای بر حسب دوره

یک دوره زمانی است که کل مجموعه داده تنها یک بار از طریق شبکه عصبی به جلو و عقب منتقل می‌شود. از آنجایی که یک دوره ممکن است بسیار بزرگ باشد و نتوان آن را به یکباره به سیستم وارد

<sup>5</sup>Batch

Vowel produced \ Perceived vowel	i	e	a	o	u
i	15		1		
e	1		1		
a			79	5	
o			4	15	3
u				2	2

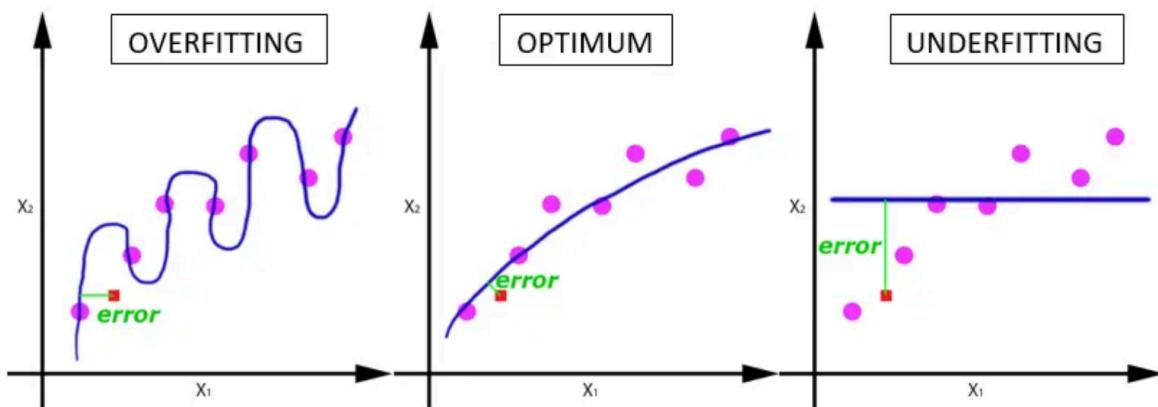
شکل ۲-۴ ماتریس درهم ریختگی در مدل MLP

کرد، به چند دسته کوچکتر تحت عنوان "دسته"<sup>۵</sup> تقسیم می‌شود. انتقال کل مجموعه داده از طریق یک شبکه عصبی به تنهایی کافی نیست و باید مجموعه داده را چندین بار به شبکه ارسال کرد. در این پروژه، از یک مجموعه داده محدود استفاده شده و برای بهینه‌سازی یادگیری آن از الگوریتم گرادیان کاهشی استفاده می‌کنیم.

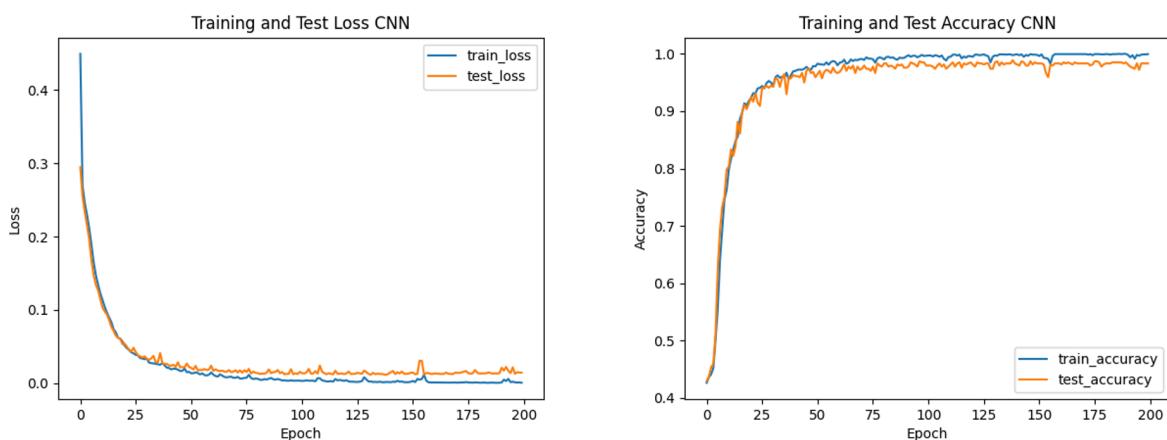
تعیین تعداد صحیح دوره‌ها از اهمیت بالایی برخوردار است. با افزایش تعداد دوره‌ها، تعداد دفعات تغییر وزن در شبکه عصبی بیشتر می‌شود و منحنی یادگیری از حالت کم‌پرازش<sup>۶</sup> به حالت بهینه<sup>۷</sup> و در نهایت به حالت بیش‌پرازش<sup>۸</sup> تغییر می‌یابد. تعداد دوره‌ها باید به گونه‌ای تعیین شود که تعادلی برقرار شود و بتوان منحنی یادگیری را به بهترین حالت ممکن رساند.

در این پروژه، به منظور نظارت و ارزیابی عملکرد مدل، نمودارهای دقت و خطای بر حسب دوره ترسیم شده‌اند. این نمودارها نشان می‌دهند که چگونه دقت مدل و میزان خطای در طول زمان تغییر می‌کنند. بررسی این نمودارها می‌تواند به شناسایی نقاط بهینه و جلوگیری از بیش‌پرازش کمک کند.

<sup>6</sup>Underfitting<sup>7</sup>Optimal<sup>8</sup>Overfitting



شکل ۳-۴ منحنی‌های کم‌برازش، بهینه و بیش‌برازش



شکل ۴-۴ نمودار روند دقت و خطا بر حسب دوره در داده‌های آموزش و تست در مدل CNN

#### ۴-۴ سرعت اجرای برنامه

به منظور رسیدن به یکی از اهداف اصلی این پروژه، که زمان واقعی بودن آن می‌باشد، لازم است که میزان پاسخ‌گویی مدل‌ها نیز مورد بررسی و ارزیابی قرار گیرد. در جدول زیر، زمان پاسخ‌گویی مدل‌ها از زمانی که داده‌ها از طریق دوربین خوانده می‌شوند تا زمانی که دستور به پهپاد داده می‌شود، آورده شده است.

معماری مدل	زمان پاسخ‌گویی مدل
۶	MLP
۷	CNN
۵	LSTM

جدول ۲-۴ جدول ارزیابی زمان پاسخ‌گویی مدل‌ها

## ۴-۵ سختافزار مورد نیاز

این پروژه باید به گونه‌ای اجرا می‌شد که بر روی ساده‌ترین سیستم‌های کامپیوتری نیز قابل اجرا باشد، زیرا سختافزار پهپادها به‌طور معمول دارای پردازنده‌های ضعیف‌تری هستند. همچنین، استفاده از کارت گرافیکی ممکن نبود، چرا که پهپادها قادر کارت گرافیکی می‌باشند. معماری‌های پیاده‌سازی شده به نحوی طراحی شدند که تعادل میان دقت و بهره‌وری از سختافزار حفظ شود، به طوری که هم قابلیت اجرای زمان واقعی داشته باشند و هم امکان پیاده‌سازی آن‌ها بر روی پهپاد فراهم باشد. از این‌رو، معماری‌ها به گونه‌ای پیاده‌سازی شدند که بر روی پردازنده اجرا شوند. میزان استفاده از پردازنده برای اجرای معماری‌های پیاده‌سازی شده در این پروژه به شرح زیر است:

معماری مدل پردازنده فضای ذخیره‌شده		
۷	۶	MLP
۸	۷	CNN
۸	۵	LSTM

جدول ۳-۴ جدول ارزیابی سختافزار موردنیاز مدل‌ها

## ۴-۶ مقایسه دقت پروژه ما با کارهای مشابه

پروژه پیاده‌سازی از دقت بسیار بالایی برخوردار است که این دقت به دلیل ترکیب پیش‌پردازش، کتابخانه MediaPipe، مدل کلاس‌بندی و پس‌پردازش است. در این قسمت به مقایسه دقت پروژه پیاده‌سازی شده با پروژه‌های مشابه با هدف کنترل پهپاد با کمک تشخیص ژست دست پرداخته‌ایم.

## ۷-۴ جمع‌بندی

مدل‌های ارائه شده در این پروژه دارای دقت و کارایی مناسبی هستند که امکان استفاده آن‌ها در کاربردهای واقعی را فراهم می‌کند. این مدل‌ها قادرند به خوبی ژست‌های دست را تشخیص داده و از آن‌ها در عملیات مختلفی مانند کنترل دستگاه‌ها و رابطه‌ای کاربری استفاده شود.

نام مقاله	تعداد ژست‌های دست	دقت تشخیص ژست دست
برنامه ما	۹	۶
Hand Gesture Controlled Drones: An Open Source Library[5]	۵	۹۷.۴۷۱
A Real-Time Hand Gesture Recognition Method[14]	۶	۹۳.۸
Hand Gestures For Drone Control Using Deep Learning[6]	۹	۸۳.۳
UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition[15]	۱۳	۹۱.۹
Hand Gesture Recognition system for Real-Time Application[16]	۶	۹۰.۸
An Improved Hand Gesture Recognition System Using Keypoints and Hand Bounding Boxes[17]	۶	۹۴
Hand-Gesture Recognition Using Computer-Vision Techniques[38]	۶	۹۳.۱
Hand Gesture Recognition using Image Processing and Feature Extraction Techniques[39]	۲۸	۸۹.۵۴
Applying Hand Gesture Recognition for User Guide Application Using MediaPipe[28]	۱۰	۹۵
MediaPipe Hands: On-device Real-time Hand Tracking[19]	۸	۹۴.۷
Hand Gesture Recognition System based in Computer Vision and Machine Learning[40]	۲۸	۹۳.۷۲
Hand Gesture Recognition Using Hidden Markov Models[41]	۵	۹۲.۱
Hand Gesture Recognition Using Kinect[42]	۳۸	۸۴
Hand Gesture Recognition Using Neural Networks[43]	۱۰	۸۹

جدول ۴-۴ جدول مقایسه پروژه با کارهای مشابه

علاوه بر این، با مقایسه عملکرد پروژه ما با کارهای مشابه در حوزه تشخیص ژست‌های دست، مشخص شده است که پروژه ما دارای دقیق‌ترین مشابهی باحتی بالاتری است، به ویژه اگر تعداد کلاس‌های ژست دست در نظر گرفته شود. این نکته نشان می‌دهد که مدل‌های ارائه شده به خوبی تنوع و پیچیدگی ژست‌های دست را درک می‌کنند و قادر به تشخیص آن‌ها هستند، که این امر یکی از چالش‌های اصلی در این زمینه است.

با توجه به این نتایج، می‌توانیم اطمینان داشته باشیم که پروژه ما قادر است به عنوان یک راه حل کارآمد

برای تشخیص ژستهای دست در برنامه‌ها و سیستم‌های واقعی مورد استفاده قرار بگیرد.

# فصل پنجم

## نتیجه گیری و پیشنهادات

## ۱-۵ مقدمه

در این فصل، ما به بررسی نتایج بدست آمده و ارائه پیشنهاداتی می‌پردازیم که می‌توانند عملکرد سیستم کنترل پهپاد را از جنبه‌های مختلف بهبود بخشدند. نتایج بهدست آمده در این پروژه نشان‌دهنده دقت بالای سیستم در تشخیص ژست‌ها و انتقال دستورات به پهپاد می‌باشد. با این حال، همچنان می‌توان بهبودهایی در عملکرد سیستم اعمال کرد تا دقت و کارایی آن افزایش یابد.

این پیشنهادات شامل افزایش سرعت اجرای برنامه، بهبود دقت پیش‌بینی ژست‌ها و کاهش محدودیت‌های منابع می‌باشند. هدف از این فصل، ارائه راهکارهای عملی و کاربردی است که بتوانند سیستم را به سطح بالاتری از کارایی و اطمینان برسانند.

## ۲-۵ نتیجه گیری

نتایج بهدست آمده در این پروژه نشان‌دهنده عملکرد دقیق و موفقیت‌آمیز آن است. این پروژه به تمامی اهداف از پیش تعیین شده، از جمله دقت بالا، صحت قابل قبول، فراخوانی مناسب، امتیاز F1 بالا، اجرای زمان واقعی، رضایت کاربر، خطای پایین و قابلیت پیاده‌سازی بر روی پهپاد دست یافته است. عملکرد دقیق سیستم در تشخیص و کنترل پهپاد با استفاده از ژست‌های دست، نشان‌دهنده پتانسیل بالای این روش برای کاربردهای مختلف در صنعت و تکنولوژی است.

## ۳-۵ پیشنهادات

### ۱-۳-۵ محدودیت وجود پهپادهای منبع باز

در مورد محدودیت‌های استفاده از پهپادهای منبع باز و شرکتی برای پروژه کنترل پهپاد با استفاده از ژست دست بر مبنای بینایی ماشین، چندین موضوع موردنظر وجود دارد. ابتدا، در مورد پهپادهای شرکتی، باید توجه داشت که این پهپادها معمولاً دارای سیستم‌های بسته و غیرقابل تغییر هستند که مانع از انجام تغییرات و سفارشی‌سازی‌های موردنیاز برای اجرای الگوریتم‌های پیچیده بر روی آنها می‌شوند. همچنان، محدودیت‌های قانونی و مقرراتی مانند نیاز به مجوزهای خاص برای استفاده از پهپادهای تجاری در برخی کشورها می‌توانند چالش‌هایی را ایجاد کند.

در مورد پهپادهای منبع باز، محدودیت‌های متفاوتی وجود دارد. این پهپادها ممکن است دارای سخت‌افزارهای

قدرتمندی نباشد که بتوانند الگوریتم‌های پیچیده پردازش تصویر را اجرا کنند. همچنین، عدم پشتیبانی کامل از زبان‌های برنامه‌نویسی مدرن می‌تواند باعث پیچیدگی در توسعه و اجرای برنامه‌ها شود. با در نظر گرفتن این محدودیت‌ها، راهکار ساخت یک پهپاد سفارشی با سختافزار مناسب و قابلیت برنامه‌نویسی مورد نیاز می‌تواند راه حلی مناسب باشد. این رویکرد امکان اجرای الگوریتم‌های پیچیده و پیاده‌سازی برنامه‌های سفارشی را فراهم می‌کند، که باعث بهبود کارایی و انعطاف‌پذیری سیستم در ارتباط با کنترل پهپاد با ژست دست خواهد شد.

### ۲-۳-۵ وجود چندین دست در تصویر

وجود چندین دست در تصویر یک چالش بزرگ در برنامه است. در این مورد باید به این نکته توجه کنیم که در یک فریم تصویر، ممکن است چندین دست قابل مشاهده باشند که این می‌تواند برای الگوریتم‌های تشخیص دست مشکل‌ساز شود. زیرا الگوریتم‌ها باید بتوانند دست‌ها را تفکیک کرده و ژست‌های متفاوت آنها را تشخیص دهند. همچنین ممکن است دست‌های غیرمرتبط با کاربر، مانند دست‌های در پس‌زمینه یا دست‌های دیگر افراد، نیز در تصویر حضور داشته باشند که این موضوع می‌تواند دقیقیت الگوریتم را کاهش دهد.

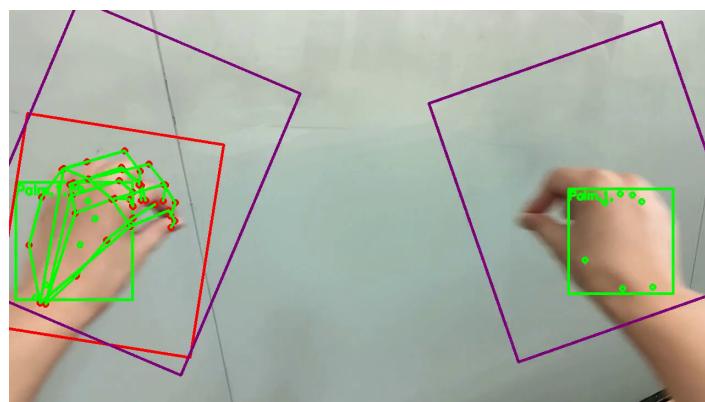
یکی از راهکارهای موردنظر برای حل این چالش، شخصی‌سازی مدل تشخیص دست با توجه به داده‌هایی که از کاربران جمع‌آوری می‌شود، است. این روش می‌تواند به کمک داده‌های جمع‌آوری شده از دست‌های واقعی کاربران، به مدل کمک کند تا الگوهای مختلف دست‌ها را بیشتر بشناسد و از این طریق دقیق‌تر تشخیص را افزایش دهد. با این حال، این روش نیازمند زمان و تلاش برای جمع‌آوری و برچسب‌گذاری داده‌های موردنیاز است و ممکن است برای کاربران خوشایند نباشد.

راهکار دیگری که می‌توان برای حل این مسئله در نظر گرفت، استفاده از یک سیستم ارتباطی میان دست و پهپاد است. به این ترتیب، کاربر می‌تواند یک دستبند یا سیمی را در دست خود داشته باشد که این دستبند به عنوان نشانه تشخیص دست و ارتباط بین کاربر و پهپاد عمل کند. با این روش، پهپاد تنها دستی را که دارای دستبند است شناسایی می‌کند و به دستورات آن پاسخ می‌دهد. این راهکار می‌تواند از دقیقیت و سرعت الگوریتم تشخیص بهره‌مند شود و به کاربر اطمینان بیشتری بدهد. با این حال، برای اجرای این روش نیازمند توسعه و اعمال یک مدل دیگر برای تشخیص دست با دستبند است که این می‌تواند زمان و هزینه بیشتری را به بار آورد.

### ۳-۳-۵ شناسایی یک دست در دو مستطیل در نتیجه دو خروجی دست

یکی از مشکلاتی که بسیاری از استفاده‌کنندگان کتابخانه MediaPipe با آن مواجه شده‌اند، مربوط به شناسایی یک دست با دو مستطیل متفاوت و در نتیجه دو خروجی است. این مشکل می‌تواند به علت همپوشانی بین دو مستطیلی که دست را شناسایی می‌کنند، ایجاد شود. با این حال، این مشکل برای پژوهه خاصی که ما در دست داریم تفاوتی ایجاد نکرده است، زیرا ما یک گزینه را انتخاب کرده و ژست دست را در آن پیش‌بینی می‌کنیم.

اما از بین بردن این مشکل می‌تواند بهبود و بهینه‌سازی کارایی کتابخانه برای سایر استفاده‌کنندگان مفید باشد. برای حل این مشکل، می‌توان محدودیتی برای همپوشانی مستطیل‌های حاوی دست را در نظر گرفت. به این صورت که اگر دو مستطیل بیشتر از حد در نظر گرفته شده همپوشانی داشته باشند، یعنی یک دست در دو تصویر تشخیص داده شده باشد، تنها یکی از آنها به مدل بعدی ارسال شود. این روش می‌تواند به استفاده‌کنندگان دیگر از کتابخانه MediaPipe کمک کند تا با مشکلاتی از این دست مواجه نشوند و از امکانات بهتری برای تشخیص دست بهره‌مند شوند.



شکل ۱-۵ شناسایی یک دست دو مرتبه

### ۴-۳-۵ اجرا روی کارت گرافیکی<sup>۱</sup>

پژوهه ما به صورتی پیاده‌سازی شده است که تنها بر روی پردازنده مرکزی قابل اجرا است، به علت عدم در دسترس بودن پهپادی که از کارت گرافیکی برای اجرای مدل‌ها استفاده کند. با این حال، اگر پهپادی با توانایی اجرای مدل‌ها بر روی کارت گرافیکی در دسترس باشد، ۳ مدل که به صورت پی‌درپی اجرا می‌شوند می‌توانند بر روی کارت گرافیکی اجرا شده تا عملکرد سیستم را بهبود بخشیده و تجربه کاربری را بهبود

<sup>۱</sup>Graphics Processing Units

بخشنده استفاده از کارت گرافیکی برای اجرای مدل‌ها می‌تواند سرعت پردازش را بسیار افزایش دهد، زیرا کارت‌های گرافیکی معمولاً دارای تعداد زیادی هسته محاسباتی هستند که برای پردازش موازی و سریع داده‌ها بسیار مناسب هستند. این امر می‌تواند منجر به بهبود قابل توجهی در زمان پاسخ و دقت تشخیص ژست دست شود و تجربه کاربری را بهبود بخشد. به علاوه، با استفاده از کارت گرافیکی، می‌توانیم بار محاسباتی را از پردازنده مرکزی کاهش داده و عملکرد کلی سیستم را بهبود بخشیم.

### ۵-۳-۵ اجرای پروژه با برنامه C++

با توجه به مزایای زبان C++ نسبت به پایتون، انتخاب این زبان برای این می‌تواند به بهبود عملکرد و کارایی سیستم کمک شایانی کند. در ادامه به برخی از مزایای استفاده از زبان C++ نسبت به پایتون در این پروژه پرداخته شده است:

- عملکرد و سرعت اجرای بالا: C++ یک زبان برنامه‌نویسی کامپایل شده است که به کد ماشین تبدیل می‌شود، به همین دلیل عملکرد و سرعت اجرای بسیار بالاتری نسبت به پایتون دارد که یک زبان تفسیری است. این ویژگی واقعی مانند کنترل پهپاد با استفاده از ژست دست بسیار حیاتی است، زیرا زمان پاسخ‌دهی سریع و دقیق در این سیستم‌ها اهمیت زیادی دارد.
- مدیریت دقیق حافظه: C++ امکان مدیریت دقیق حافظه را فراهم می‌کند. این ویژگی می‌تواند بهینه‌سازی استفاده از منابع سیستم و کاهش مصرف حافظه را به دنبال داشته باشد، که برای سیستم‌هایی با منابع محدود مانند پهپادها بسیار مهم است. مدیریت صحیح حافظه می‌تواند به جلوگیری از نشت حافظه و افزایش پایداری سیستم کمک کند.
- کتابخانه‌ها و چارچوب‌های قدرتمند: C++ دارای کتابخانه‌ها و چارچوب‌های قدرتمندی برای پردازش تصویر و بینایی ماشین مانند OpenCV است. به زبان C++ نوشته شده و عملکرد بهتری نسبت به معادلهای پایتونی خود دارد. این کتابخانه قابلیت‌های متعددی را برای پردازش تصاویر، شناسایی و تشخیص اشیا فراهم می‌کند که می‌تواند در پروژه‌های بینایی ماشین بسیار موثر باشد.
- دسترسی و کنترل سطح پایین سخت‌افزار: C++ امکان دسترسی و کنترل سطح پایین سخت‌افزار را فراهم می‌کند. این ویژگی برای کنترل مستقیم پهپاد و بهینه‌سازی ارتباط با سنسورها و موتورها بسیار مفید است. با استفاده از C++ می‌توان به طور مستقیم با اجزای سخت‌افزاری پهپاد ارتباط

برقرار کرده و عملکرد سیستم را بهینه کرد.

- پشتیبانی قوی از چندریسمانی<sup>۲</sup>: C++ پشتیبانی قوی تری از چندریسمانی و پردازش موازی دارد. این قابلیت می‌تواند برای بهبود کارایی و پاسخ‌دهی سریع در پردازش‌های پیچیده بینایی ماشین و کنترل پهپاد بسیار مهم باشد. با استفاده از چندریسمانی، می‌توان عملیات مختلف را به صورت همزمان انجام داد و زمان اجرای کل سیستم را کاهش داد.

### ۶-۳-۶ استفاده از مدل‌های موازی برای تشخیص ژست دست

برای بهبود دقت و کارایی پروژه کنترل پهپاد با استفاده از ژست دست، می‌توان از رویکردی مبتنی بر اجرای موازی مدل‌ها بهره برد. در این روش، تصویر تشخیص داده شده از دست به طور همزمان به دو مدل با معماری‌های مختلف ارسال می‌شود. به عنوان مثال، یک مدل می‌تواند از معماری شبکه‌های عصبی پیچشی و مدل دیگر از معماری حافظه بلندمدت کوتاه‌مدت استفاده کند. تنها در صورتی که هر دو مدل تشخیص یکسانی از ژست دست داشته باشند، دستور نهایی به پهپاد ارسال می‌شود. این رویکرد می‌تواند به طور قابل توجهی دقت تشخیص ژست دست را افزایش دهد و ریسک‌های مالی مرتبط با اشتباهات تشخیصی را کاهش دهد.

### ۷-۳-۵ توسعه برنامه با استفاده از سایر قسمت‌های بدن انسان

یکی از مسیرهای جذاب برای توسعه آینده پروژه کنترل پهپاد با ژست دست، گسترش سیستم به گونه‌ای است که بتوان از قسمت‌های دیگر بدن انسان برای کنترل حرکت پهپاد استفاده کرد. این رویکرد می‌تواند به ایجاد تعاملات طبیعی‌تر و راحت‌تر بین کاربر و پهپاد منجر شود. در زیر به برخی از این ایده‌ها و مزایای آن‌ها پرداخته می‌شود.

### کنترل پهپاد با حرکات چشم

یکی از نوآورانه‌ترین روش‌ها برای کنترل پهپاد، استفاده از حرکات چشم کاربر است. این سیستم می‌تواند به وسیله ردیابی حرکت چشم کاربر، مکانی نسبی را که کاربر به آن نگاه می‌کند، شناسایی کرده و پهپاد را به سمت آن نقطه هدایت کند. برای پیاده‌سازی این روش، از تکنیک‌های پیشرفته بینایی ماشین و

<sup>2</sup>Multithreading

ردیابی چشم استفاده می‌شود.

استفاده از حرکات چشم برای کنترل پهپاد مزایای متعددی دارد. این روش می‌تواند دقیق و کارایی سیستم را افزایش دهد، زیرا حرکات چشم نقاط دقیق‌تری را نسبت به حرکات دست مشخص می‌کنند. این امر به پهپاد امکان می‌دهد تا با دقیق‌تری به نقاط مورد نظر حرکت کند. علاوه بر این، استفاده از چشم برای کنترل پهپاد می‌تواند بسیار طبیعی‌تر و راحت‌تر از حرکات دست باشد، به ویژه در شرایطی که دست‌ها مشغول هستند یا امکان استفاده از آن‌ها وجود ندارد. این روش همچنین می‌تواند خستگی ناشی از استفاده مداوم از دست‌ها برای کنترل پهپاد را کاهش دهد، که در نتیجه تجربه کاربری بهتری را فراهم می‌کند.

### کنترل پهپاد با حرکات سر

روش دیگر برای کنترل پهپاد، استفاده از حرکات سر است. در این سیستم، کاربر می‌تواند با چرخش سر به جهات مختلف، مسیر حرکت پهپاد را تعیین کند. حرکات سر به طور طبیعی و بدون نیاز به تجهیزات اضافی قابل انجام هستند، که این امر تعامل با پهپاد را ساده‌تر می‌کند. استفاده از حرکات سر به ویژه در محیط‌هایی که استفاده از دست‌ها ممکن نیست (مثل محیط‌های کاری خاص یا شرایط پژوهشی) بسیار مفید است. این روش می‌تواند به سادگی و طبیعی بودن تعامل با پهپاد کمک کند و تجربه کاربری را بهبود بخشد.

## ۴-۵ جمع‌بندی

در این فصل، به نتیجه بدست آمده و بررسی چندین چالش‌ها و محدودیت‌های سیستم کنترل پهپاد پرداختیم و راهکارهای متعددی برای بهبود عملکرد آن ارائه دادیم.

اولین محدودیت، استفاده از پهپادهای منبع‌باز است که با وجود مزایای فراوان، ممکن است دارای محدودیت‌های سخت‌افزاری و نرم‌افزاری باشند. برای غلبه بر این محدودیت، پیشنهاد شد که از پهپادهای با قابلیت‌های پیشرفته‌تر و بهینه‌سازی کدهای نرم‌افزاری استفاده شود. همچنین، راهکارهایی برای مدیریت چندین دست در تصویر و جلوگیری از شناسایی نادرست یک دست در دو مستطیل ارائه شد که می‌تواند دقیق‌تر سیستم را بهبود بخشد.

در ادامه، به اهمیت اجرای پروژه روی کارت گرافیکی و برنامه‌نویسی به زبان C++ پرداختیم که می‌تواند سرعت اجرای برنامه را افزایش دهد. استفاده از مدل‌های موازی برای تشخیص ژست دست نیز به عنوان

یک راهکار مؤثر برای بهبود دقیق و سرعت پیشنهاد شد. همچنین، روش‌های جدیدی مانند کنترل پهپاد با حرکات چشم و سر معرفی شدند که امکانات بیشتری برای کاربران فراهم می‌کنند.

در نتیجه، با توجه به بررسی‌ها و پیشنهادات ارائه شده در این فصل، می‌توان انتظار داشت که با اعمال این بهبودها، سیستم کنترل پهپاد به سطح بالاتری از کارایی و اطمینان دست یابد. این راهکارها نه تنها عملکرد سیستم را بهبود می‌بخشند، بلکه تجربه کاربری بهتری را نیز فراهم می‌آورند.

# كتاب نامه

- [1] Walter, Ian and Khadr, Monette. Gesture controlled drone.
- [2] Puri, Vikram, Nayyar, Anand, and Raja, Linesh. Agriculture drones: A modern breakthrough in precision agriculture. *Journal of Statistics and Management Systems*, 20(4):507–518, 2017.
- [3] Gatteschi, Valentina, Lamberti, Fabrizio, Paravati, Gianluca, Sanna, Andrea, Demartini, Claudio, Lisanti, Alberto, and Venezia, Giorgio. New frontiers of delivery services using drones: A prototype system exploiting a quadcopter for autonomous drug shipments. in *2015 IEEE 39th annual computer software and applications conference*, vol. 2, pp. 920–927. IEEE, 2015.
- [4] Moore, T. Nypd considering using drones to fight crime. *New York Daily*, 2014.
- [5] Natarajan, Kathiravan, Nguyen, Truong-Huy D, and Mete, Mutlu. Hand gesture controlled drones: An open source library. in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, pp. 168–175. IEEE, 2018.
- [6] Hadri, Soubhi. Hand gestures for drone control using deep learning. 2018.
- [7] Zhu, Pengfei, Wen, Longyin, Bian, Xiao, Ling, Haibin, and Hu, Qinghua. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.
- [8] Guvenc, Ismail, Koohifar, Farshad, Singh, Simran, Sichitiu, Mihail L, and Matolak, David. Detection, tracking, and interdiction for amateur drones. *IEEE Communications*

*Magazine*, 56(4):75–81, 2018.

- [9] Hassanalian, Mostafa and Abdelkefi, Abdessattar. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace sciences*, 91:99–131, 2017.
- [10] Yoo, Minjeong, Na, Yuseung, Song, Hamin, Kim, Gamin, Yun, Junseong, Kim, Sangho, Moon, Changjoo, and Jo, Kichun. Motion estimation and hand gesture recognition-based human–uav interaction approach in real time. *Sensors*, 22(7):2513, 2022.
- [11] Ma, Yuntao, Liu, Yuxuan, Jin, Ruiyang, Yuan, Xingyang, Sekha, Raza, Wilson, Samuel, and Vaidyanathan, Ravi. Hand gesture recognition with convolutional neural networks for the multimodal uav control. in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pp. 198–203. IEEE, 2017.
- [12] Hu, Bin and Wang, Jiacun. Deep learning based hand gesture recognition and uav flight controls. *International Journal of Automation and Computing*, 17(1):17–29, 2020.
- [13] Sarkar, Ayanava, Patel, Ketul Arvindbhai, Ram, RK Ganesh, and Kapoor, Geet Krishna. Gesture control of drone using a motion controller. in *2016 international conference on industrial informatics and computer systems (ciics)*, pp. 1–5. IEEE, 2016.
- [14] Fang, Yikai, Wang, Kongqiao, Cheng, Jian, and Lu, Hanqing. A real-time hand gesture recognition method. in *2007 IEEE International Conference on Multimedia and Expo*, pp. 995–998. IEEE, 2007.
- [15] Perera, Asanka G, Wei Law, Yee, and Chahl, Javaan. Uav-gesture: A dataset for uav control and gesture recognition. in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.
- [16] Murugeswari, M and Veluchamy, S. Hand gesture recognition system for real-time

- application. in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1220–1225. IEEE, 2014.
- [17] Dang, Tuan Linh, Tran, Sy Dat, Nguyen, Thuy Hang, Kim, Suntae, and Monet, Nicolas. An improved hand gesture recognition system using keypoints and hand bounding boxes. *Array*, 16:100251, 2022.
- [18] Chen, Boxu, Yu, Lixin, Meng, Xiao, and Hua, Yang. Visual gesture recognition based on hand key points. in *Journal of Physics: Conference Series*, vol. 2024, p. 012037. IOP Publishing, 2021.
- [19] Zhang, Fan, Bazarevsky, Valentin, Vakunov, Andrey, Tkachenka, Andrei, Sung, George, Chang, Chuo-Ling, and Grundmann, Matthias. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.
- [20] Waqar, Ahsan, Othman, Idris, Hamah Sor, Nadhim, Alshehri, Abdullah Mohammed, Almuhibah, Hamad R, Alotaibi, Badr Saad, Abuhussain, Mohammed Awad, Bageis, Abdulrahman S, Althoey, Fadi, Hayat, Saleh, et al. Modeling relation among implementing ai-based drones and sustainable construction project success. *Frontiers in Built Environment*, 9:1208807, 2023.
- [21] Ghazi, Georges and Voyer, Julien. Use of a dji tello drone as an educational platform in the field of control engineering. *Proceedings of the Canadian Engineering Education Association (CEEA)*, 2023.
- [22] Clough, Sharice and Duff, Melissa C. The role of gesture in communication and cognition: Implications for understanding and treating neurogenic communication disorders. *Frontiers in Human Neuroscience*, 14:323, 2020.
- [23] Kang, Seokmin and Tversky, Barbara. From hands to minds: Gestures promote understanding. *Cognitive Research: Principles and Implications*, 1:1–15, 2016.

- [24] Introduction to tensorflow. <https://www.tensorflow.org/learn>. (Accessed on 06/02/2024).
- [25] scikit-learn - wikipedia. <https://en.wikipedia.org/wiki/Scikit-learn>. (Accessed on 06/02/2024).
- [26] About keras 3. <https://keras.io/about/>. (Accessed on 06/02/2024).
- [27] Lugaressi, Camillo, Tang, Jiuqiang, Nash, Hadon, McClanahan, Chris, Uboweja, Esha, Hays, Michael, Zhang, Fan, Chang, Chuo-Ling, Yong, Ming, Lee, Juhyun, et al. Mediapipe: A framework for perceiving and processing reality. in *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, vol. 2019, 2019.
- [28] Harris, Moh, Agoes, Ali Suryaperdana, et al. Applying hand gesture recognition for user guide application using mediapipe. in *2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, pp. 101–108. Atlantis Press, 2021.
- [29] What is numpy? — numpy v1.26 manual. <https://numpy.org/doc/stable/user/whatisnumpy.html>. (Accessed on 06/02/2024).
- [30] Introduction to matplotlib - geeksforgeeks. <https://www.geeksforgeeks.org/python-introduction-matplotlib/>. (Accessed on 06/02/2024).
- [31] About - opencv. <https://opencv.org/about/>. (Accessed on 06/02/2024).
- [32] Bhujbal, Kunal and Barahate, Sachin. Custom object detection based on regional convolutional neural network & yolov3 with dji tello programmable drone. in *7th International Conference on Innovation & Research in Technology & Engineering (ICIRTE)*, 2022.
- [33] Accuracy vs. precision vs. recall in machine learning: what's the difference? <https://www.evidentlyai.com/classification-metrics/>

accuracy-precision-recall#:~:text=Accuracy%20is%20a%20metric%20that,often%20the%20model%20is%20right%3F. (Accessed on 06/03/2024).

[34] Precision. <https://c3.ai/glossary/machine-learning/precision/#:~:text=Precision%20is%20one%20indicator%20of, the%20number%20of%20false%20positives>. (Accessed on 06/03/2024).

[35] Understanding precision, recall, f1-score, and support in machine learning evaluation | by nirajan acharya | medium. <https://medium.com/@nirajan.acharya777/understanding-precision-recall-f1-score-and-support-in-machine-learning-evaluati> (Accessed on 06/03/2024).

[36] F1-score in machinaal leren - geeksforgeeks. <https://www.geeksforgeeks.org/f1-score-in-machine-learning/>. (Accessed on 06/03/2024).

[37] Confusion matrix - an overview | sciencedirect topics. <https://www.sciencedirect.com/topics/engineering/confusion-matrix>. (Accessed on 06/03/2024).

[38] Rios-Soria, David J, Schaeffer, Satu E, and Garza-Villarreal, Sara E. Hand-gesture recognition using computer-vision techniques. 2013.

[39] Sharma, Ashish, Mittal, Anmol, Singh, Savitoj, and Awatramani, Vasudev. Hand gesture recognition using image processing and feature extraction techniques. *Procedia Computer Science*, 173:181–190, 2020.

[40] Trigueiros, Paulo, Ribeiro, Fernando, and Reis, Luís Paulo. Hand gesture recognition system based in computer vision and machine learning. *Developments in medical image processing and computational vision*, pp. 355–377, 2015.

[41] Min, Byung-Woo, Yoon, Ho-Sub, Soh, Jung, Yang, Yun-Mo, and Ejima, Toshiaki. Hand gesture recognition using hidden markov models. in *1997 IEEE International*

*Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 5, pp. 4232–4235. IEEE, 1997.

- [42] Li, Yi. Hand gesture recognition using kinect. in *2012 IEEE International Conference on Computer Science and Automation Engineering*, pp. 196–199. IEEE, 2012.
- [43] Murthy, GRS and Jadon, RS. Hand gesture recognition using neural networks. in *2010 IEEE 2nd international advance computing conference (IACC)*, pp. 134–138. IEEE, 2010.