

الامر بالمعروف والنهي عن المنكر



دانشکده مهندسی برق و ریاتیک

پایان نامه کارشناسی ارشد مهندسی مخابرات سیستم

طراحی و ساخت یک بازی ویدئویی تعاملی برای دستگاه‌های اندرویدی با طبقه‌بند آبشاری

نگارنده: محمود محمدزاده

استاد راهنما

دکتر حسین خسروی

بهمن ۱۳۹۸

ب

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای محمود محمدزاده با شماره دانشجویی ۹۶۱۳۱۵۴ رشته مخابرات گرایش (سیستم) تحت عنوان: طراحی و ساخت یک بازی ویدئویی تعاملی برای دستگاه‌های اندرویدی با استفاده از شبکه‌های عصبی که در تاریخ ۹۸/۱۱/۰۷ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می‌گردد:

الف) درجه عالی: نمره ۲۰-۱۹ ب) درجه خیلی خوب: نمره ۱۸-۱۸/۹۹
 ج) درجه خوب: نمره ۱۷/۹۹-۱۶ د) درجه متوسط: نمره ۱۵/۹۹-۱۴
 ه) کمتر از ۱۴ غیر قابل قبول و نیاز به دفاع مجدد دارد
 نوع تحقیق: نظری عملی

اعضاء	مرتبه علمی	نام و نام خانوادگی	عضو هیأت داوران
	استاد	محمد حسن	۱- استاد راهنمای اول
-	-	-	۲- استاد راهنمای دوم
-	-	-	۳- استاد مشاور
	استاد	سید محمود میرزایی	۴- نماینده تحصیلات تکمیلی
	استاد	عروضا احمدزاد	۵- استاد ممتحن اول
	استاد	شیرین حسینی ابراهیمی	۶- استاد ممتحن دوم

نام و نام خانوادگی رئیس دانشکده:
 تاریخ و امضاء و مهر دانشکده:
 تیسردر صورتی که کسی مردود شود حداکثر یکبار دیگر (در مدت مجاز تحصیل) می‌تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

تشکر و قدردانی:

با سپاس فراوان از زحمات استاد محترم و گران قدر جناب آقای دکتر خسروی که از ابتدای راه و در طی انجام این پژوهش، با راهنمایی‌های خود مرا در نگارش این پایان‌نامه یاری نمودند.

تعهد نامه

اینجانب محمود محمدزاده دانشجوی دوره کارشناسی ارشد رشته برق- مخابرات دانشکده برق و رباتیک دانشگاه

صنعتی شاهرود نویسنده پایان نامه طراحی و ساخت یک بازی ویدئویی تعاملی برای دستگاه‌های اندرویدی با

طبقه‌بند آبشاری تحت راهنمایی دکتر حسین خسروی متعهد می شوم:

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا «Shahrood University of Technology» به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

امروزه بازی‌های ویدئویی در میان سرگرمی‌ها از جایگاه ویژه‌ای برخوردارند. بعضی از بازی‌ها از آن جایی که تحرّکی در آن‌ها وجود ندارد، مضرّاتی دارند. در این بین، بازی‌هایی که با حرکات بدن کنترل می‌شوند، می‌توانند جایگزین مناسبی برای سایر بازی‌های ویدئویی باشند چرا که باعث ایجاد تحرّک در افراد می‌شوند.

در این پایان‌نامه، قصد توسعه یک بازی ویدئویی برای دستگاه‌های اندرویدی داریم به گونه‌ای که بتوان با استفاده از شناسایی حالت‌های دست کاربر و تشخیص چهره‌ی او به کنترل شخصیت بازی پرداخت.

از جمله مشکلاتی که در این راه وجود دارد، توان پردازشی کم گوشی‌های موبایل نسبت به رایانه‌ها است. به همین منظور برای شناسایی حالت‌های دست و تشخیص چهره کاربر، از روش‌های بلادرنگ استفاده کرده‌ایم. در این پایان‌نامه از طبقه‌بندهای آبخاری به همراه استخراج ویژگی‌های شبه‌هار و الگوی محلی باینری استفاده کرده‌ایم و ایده‌های مختلفی در به کارگیری آن‌ها استفاده شده است تا به ایده‌ال‌ترین حالت ممکن برسیم. با تشخیص صورت، دست‌ها و جهت حرکت آن‌ها، شخصیت بازی را کنترل می‌کنیم. الگوریتم پیشنهادی بر روی تصاویر دریافتی ازدوربین جلوی گوشی موبایل اعمال می‌شود. برای طراحی و پیاده‌سازی بازی ویدئویی از نرم‌افزار یونیتی و کتابخانه‌ی OpenCV برای یونیتی استفاده شده و زبان برنامه‌نویسی مورد استفاده زبان سی‌شارپ است.

این بازی را برای دو سیستم‌عامل ویندوز و موبایل توسعه دادیم که میزان دقت در انجام حرکت صحیح، برای هر دو سیستم‌عامل ۸۶/۴ درصد بوده و در نسخه ویندوز و اندروید به ترتیب به نرخ قاب بر ثانیه ۱۱ و ۸ رسیدیم.

کلیدواژه‌ها: بازی‌های ویدئویی، گوشی‌های موبایل، شناسایی حالت‌های دست، تشخیص چهره، طبقه‌بندهای آبخاری.

فهرست مطالب

صفحه

عنوان

۱ فصل اول

۲ ۱-۱ مقدمه

۴ ۲-۱ ضرورت انجام پژوهش

۵ ۱-۲-۱ بازی‌های کنترل شونده با حرکات بدن

۶ ۲-۲-۱ مزایا و ضرورت حضور در صنعت بازی‌سازی

۹ ۳-۲-۱ مزایای توسعه بازی برای دستگاه‌های موبایل

۱۱ ۳-۱ اهداف پایان‌نامه

۱۱ ۴-۱ پیش‌نیازهای پژوهش

۱۲ ۵-۱ ساختار پایان‌نامه

۱۳ فصل دوم

۱۴ ۱-۲ تحقیقات انجام شده با کینکت

۲۳ 2-2 شناسایی چهره، حالت‌های دست و شناسایی اشیا

۲۳ ۱-۲-۲ شناسایی چهره

۲۸ ۲-۲-۲ شناسایی حالت‌های دست

۳۰ ۳-۲-۲ شناسایی اشیا

۳۹ ۳-۲ نتیجه‌گیری

۴۱ فصل سوم

۴۲ ۱-۳ الگوریتم ویولا-جونز

۴۳ ۱-۱-۳ انتگرال تصویر

۴۶ ۲-۱-۳ آموزش طبقه‌بند

۴۹ ۳-۱-۳ طبقه‌بند آبخاری

۵۲ ۴-۱-۳ الگوی محلی باینری

۵۳ فصل چهارم

۵۵ ۱-۴ استفاده از روش ویولا-جونز

۵۵ ۱-۱-۴ انتخاب روش

۵۶ ۲-۱-۴ مشخصات اجرایی پژوهش

۵۷ ۳-۱-۴ الگوریتم پیشنهادی

۶۰ ۴-۱-۴ آموزش طبقه‌بندها و ایده استفاده‌شده در طبقه‌بندها

۲-۴ نرم افزارها، کتابخانه‌ها و زبان‌های برنامه‌نویسی مورد استفاده ۶۴

۳-۴ ارتباط بین کاربر و شخصیت درون بازی ۶۶

۴-۴ طراحی بازی ویدئویی ۶۶

۵-۴ نسخه‌های مختلف بازی ویدئویی ۶۹

۷۱ فصل پنجم

۱-۵ آزمایش دقت ۷۲

۱-۱-۵ سنجش میزان دقت حرکات در حالت ایده‌ال ۷۲

۲-۱-۵ سنجش میزان دقت در شرایط نوری مختلف ۷۳

۳-۱-۵ مقایسه میزان دقت در رایانه و گوشی موبایل ۷۴

۴-۱-۵ مقایسه میزان دقت با سایر کارهای انجام شده ۷۴

۲-۵ آزمایش سرعت ۷۶

۱-۲-۵ مقایسه سرعت اجرا در رایانه و گوشی موبایل ۷۶

۲-۲-۵ مقایسه سرعت اجرا در نسخه ویندوز با سایر کارهای انجام شده ۷۷

۳-۵ جمع‌بندی ۷۹

۴-۵ پیشنهادها و کارهای آینده ۷۹

عنوان

صفحه

مراج ۸۱

فهرست شکل‌ها

عنوان	صفحه
شکل ۱-۱ استفاده هم‌زمان از عینک واقعیت مجازی و کنترل کننده پلی استیشن	۳
شکل ۲-۱ دوربین نسخه ۲ پلی استیشن	۳
شکل ۳-۱ حسگر کینکت ایکس باکس وان	۴
شکل ۴-۱ کنسول بازی ایکس باکس وان	۴
شکل ۵-۱ گردش مالی سالانه فنلاند در صنعت بازی سازی	۸
شکل ۶-۱ تعداد شاغلان صنعت بازی در فنلاند	۸
شکل ۷-۱ تعداد شرکت های فعال در صنعت بازی سازی در فنلاند	۹
شکل ۸-۱ محبوبیت سیستم عامل های مختلف در شرکت های بازی سازی فنلاند	۱۰
شکل ۹-۱ تعداد اهالی بازی در موبایل، رایانه و کنسول ها در ایران	۱۰
شکل ۱-۲ تصویری از محیط بازی	۱۵
شکل ۲-۲ نمودار شناسایی حالت دست. شناسایی حالت یک دست (شاخه سمت چپ) و شناسایی حالت دو دست (شاخه سمت راست)	۱۷
شکل ۳-۲ تاثیر متعادل سازی رنگ. (الف) تصویر اصلی، (ب) شناسایی پوست تصویر اصلی، (پ) تصویر متعادل شده، (ت) شناسایی پوست تصویر متعادل شده	۱۸
شکل ۴-۲ فرآیند شناسایی دست اول	۱۹

- شکل ۲-۵ (الف) تمرین سایه به روش سنتی. (ب) تمرین سایه به روش پیشنهادی..... ۲۱
- شکل ۲-۶ فیلم نحوه‌ی صحیح انجام فن روی دست..... ۲۲
- شکل ۲-۷ دامنه حرکتی روی دست..... ۲۲
- شکل ۲-۸ روند کلی الگوریتم شناسایی چهره با استفاده از ویژگی‌های عمیق و دستگاه‌های موبایل.... ۲۵
- شکل ۲-۹ روند کلی سامانه پیشنهادی..... ۲۶
- شکل ۲-۱۰ فرآیند تشخیص و ردیابی..... ۲۷
- شکل ۲-۱۱ حالت‌های مختلف دست در این پژوهش..... ۲۸
- شکل ۲-۱۲ مراحل عملیات پردازش تصویر..... ۲۹
- شکل ۲-۱۳ یک نمایی از Mini-YOLOv3. این شبکه از ساختار YOLOv3 و شبکه‌ی ویژگی هرمی چند مقیاس برای استخراج ویژگی از تصویر ورودی استفاده می‌کند ۳۲
- شکل ۲-۱۴ مقایسه شبکه‌های YOLOv3، YOLOv3-tiny و Mini-YOLOv3 ۳۳
- شکل ۲-۱۵ دستگاه‌های نهفته مورد استفاده در این مقاله ۳۵
- شکل ۲-۱۶ شبکه‌های سلولی پیشنهادی این مقاله ۳۷
- شکل ۲-۱۷ ساختار شبکه Tinier-YOLO ۳۹
- شکل ۳-۱ مثالی از ویژگی‌های مستطیل. مجموع پیکسل‌هایی که در مستطیل‌های سفید قرار دارند از مجموع پیکسل‌های مستطیل‌های خاکستری کم می‌شوند. ویژگی‌های دو مستطیل در شکل (A) و (B)

نشان داده شده است. شکل (C) ویژگی‌های سه مستطیل و شکل (D) ویژگی‌های چهار مستطیل را نشان

می‌دهند..... ۴۴

شکل ۲-۳ مجموع پیکسل‌های درون مستطیل D با چهار آرایه مرجع می‌تواند محاسبه شود. مقدار

انتگرال تصویر در موقعیت ۱ برابر مجموع پیکسل‌های درون مستطیل A است. این مقدار در موقعیت ۲

برابر با A+B و در موقعیت ۳ برابر A+C و در موقعیت ۴ برابر A+B+C+D است. مجموع درون مستطیل

D می‌تواند توسط $4+1-(2+3)$ محاسبه شود..... ۴۵

شکل ۳-۳ الگوریتم آدابوست برای یادگیری طبقه‌بند..... ۴۷

شکل ۴-۳ ویژگی اول و دوم آدابوست. هر دو ویژگی در سطر بالایی نشان داده شده‌اند و سپس بر روی

یک چهره تمرینی در سطر پایینی پوشانده شده است..... ۴۹

شکل ۵-۳ طرح کلی یک شناساگر آبشاری. مجموعه‌ای از طبقه‌بندهای سری شده به هر زیرپنجره

اعمال می‌شود. طبقه‌بند ابتدایی تعداد زیادی از نمونه‌های منفی را با یک پردازش کوچک کنار می‌گذارد.

پردازش‌های بیشتر می‌تواند هر شکلی داشته باشد به عنوان مثال چندین مرحله دیگر از طبقه‌بندها..... ۵۰

شکل ۶-۳ محاسبات الگوی محلی باینری..... ۵۱

شکل ۷-۳ نماینده تصویر براساس الگوی محلی باینری..... ۵۲

شکل ۱-۴ روند کلی پژوهش..... ۵۴

شکل ۲-۴ الگوریتم ویولا-جونز..... ۵۵

شکل ۳-۴ حالت‌های مختلف دست کاربر در این پژوهش..... ۵۶

شکل ۴-۴ نحوه‌ی قرارگیری گوشی موبایل و کاربر در مقابل موبایل..... ۵۷

- شکل ۴-۵ روندنمای برنامه‌نویسی تشخیص چهره و حالت‌های دست..... ۵۸
- شکل ۴-۶ نمایش نواحی مورد علاقه و متغیرهای کادر مرزی چهره..... ۵۹
- شکل ۴-۷ تشخیص چهره و حرکات کاربر. (الف) و (ب) حرکت بالا. (پ) حرکت چپ از نگاه دوربین.
(ت) حرکت راست..... ۶۰
- شکل ۴-۸ روندنمای فرضی در حالت استفاده از یک طبقه‌بند برای شناسایی هر ۳ حرکت دست..... ۶۱
- شکل ۴-۹ تشخیص چهره و تعریف نواحی مختلف..... ۶۲
- شکل ۴-۱۰ نمایی از محیط نرم افزار یونیتی..... ۶۶
- شکل ۴-۱۱ جان‌دهنده طراحی شده برای اجرای انیمیشن‌های مختلف..... ۶۷
- شکل ۴-۱۲ اجزای مختلف بسته PolyDesert که در محیط بازی از آن‌ها استفاده شده است..... ۶۸
- شکل ۴-۱۳ محیط بازی ویدئویی..... ۶۸
- شکل ۵-۱ مقایسه میزان دقت..... ۷۶
- شکل ۵-۲ مقایسه میزان سرعت با سایر پژوهش‌ها..... ۷۷
- شکل ۵-۳ اجرای بازی توسط کاربر (در نمایش تصویر کاربر، سمت چپ و راست از نگاه دوربین در نظر گرفته می‌شود). (الف) تشخیص چهره و انجام حرکت راه رفتن. (ب) تشخیص سمت چپ و انجام چرخش به چپ. (پ) تشخیص سمت راست و انجام چرخش به راست..... ۷۸

فهرست جدول‌ها

صفحه	عنوان
۲۰.....	جدول ۱-۲ حالت های مختلف کنترل بازی ویدئویی.....
۲۰.....	جدول ۲-۲ عملکرد شناسایی حالت‌های مختلف دست.....
۲۳.....	جدول ۳-۲ بازخوردهای داده شده به بازیکن براساس دفعات تکرار صحیح.....
۲۶.....	جدول ۴-۲ زمان اجرای الگوریتم DFFDM بر روی گوشی‌های مختلف.....
۲۸.....	جدول ۵-۲ نتیجه شناسایی.....
۳۰.....	جدول ۶-۲ نتایج ارزیابی پژوهش.....
۳۵.....	جدول ۷-۲ مقایسه مشخصات دستگاه‌های مورد استفاده در این مقاله.....
۳۷.....	جدول ۸-۲ نتایج اجرای الگوریتم پیشنهادی بر روی دستگاه‌های نهفته‌ی مختلف.....
۳۷.....	جدول ۹-۲ نتایج ارزیابی سرعت مقاله.....
۳۸.....	جدول ۱۰-۲ مقایسه میزان سرعت و دقت دو شبکه Tiny-YOLO-v3 و Tinier-YOLO.....
۷۳.....	جدول ۱-۵ میزان دقت شناسایی.....
۷۴.....	جدول ۲-۵ میزان دقت شناسایی در شرایط نوری ضعیف.....
۷۵.....	جدول ۳-۵ نتایج پژوهش چن و همکارانش.....

فصل اول

مقدمه

بازی‌های ویدئویی یکی از مواردی است که امروزه مورد توجه افراد زیادی در سراسر دنیا قرار می‌گیرد. و تفریحی است که افراد زیادی در اوقات فراغت خود از آن‌ها بهره‌مند می‌شوند. بازی‌های ویدئویی برخلاف کتاب‌ها و فیلم‌ها که به عنوان اشکال تکامل یافته‌ی رسانه محسوب می‌شوند، نسبتاً جدید هستند و اولین کنسول بازی در اوایل دهه‌ی ۱۹۷۰ ساخته شده‌است و هنوز هم تقریباً از هر لحاظ در حال رشد و تحول هستند. هر چند سال، کنسول‌های بازی جدیدی ارائه می‌شود که نوید گرافیک واقعی‌تر، کیفیت صدای واقعی‌تر و مجموعه ویژگی‌های جدیدی را می‌دهد. در زمینه‌ی رایانه‌های شخصی، به لطف سخت‌افزار جدیدتر و بهتری که هر چند ماه ارائه می‌شود، این تغییرات سریع‌تر انجام می‌گیرد. اما سخت‌افزار مخصوص بازی، تنها چیزی نیست که تغییر می‌کند بلکه خود بازی‌ها نیز در حال تحول هستند مانند ایجاد طرح‌های کنترل جدید، ایجاد ارتباط جدید بین المان‌ها [۱].

بازی‌های ویدئویی برای انواع سامانه‌ها ارائه شده‌است مانند بازی‌های پلی‌استیشن، ایکس‌باکس، رایانه، گوشی‌های موبایل و غیره. امروزه بازی‌های ویدئویی فقط مخصوص خردسالان نیست و گاهی بازی‌ها دارای رده‌ی سنی هستند که فقط بزرگسالان می‌توانند از آن‌ها استفاده کنند. به همین منظور طراحی و ساخت این بازی‌ها، باید به گونه‌ای باشد که برای مخاطب بزرگسال دارای جاذبه باشد. روش‌های مختلفی برای این امر وجود دارد به‌عنوان مثال طراحی‌های متفاوت و سبک‌های مختلف بازی. یکی از مواردی که می‌توان به منظور جذب کردن بازی انجام داد، نحوه‌ی ارتباط با بازی است. همان‌طور که در کنسول پلی‌استیشن و ایکس‌باکس، این کار انجام شده‌است و کاربر می‌تواند با حرکات دست و سر خود به کنترل شخصیت بازی بپردازد.

در کنسول پلی‌استیشن ۴، کاربر می‌تواند علاوه بر این کنسول، دوربین پلی‌استیشن را نیز تهیه نماید. کاربر زمانی که یک بازی واقعیت مجازی را انجام می‌دهد، به منظور ارسال فرامین به شخصیت درون بازی مجبور است همچنان که از عینک واقعیت مجازی استفاده می‌کند، از کنترل‌کننده‌های پلی‌استیشن

هم استفاده نماید که کمی سخت به نظر می‌رسد چرا که کاربر به هنگام استفاده از عینک واقعیت مجازی، امکان دیدن کلیدهای کنترل کننده بازی را ندارد، همان‌طور که در شکل ۱-۱ مشاهده می‌کنید. اما زمانی که از دوربین پلی‌استیشن استفاده می‌کند، به لطف ردیابی حرکات سر و ژست‌ها دیگر نیازی به کنترل کننده نیست. همچنین امکان ارسال فرامین صوتی نیز فراهم شده است. البته این موضوع فقط برای بازی‌های خاصی امکان اجرا دارد!



شکل ۱-۱ استفاده هم‌زمان از عینک واقعیت مجازی و کنترل کننده پلی‌استیشن.^۲



شکل ۲-۱ دوربین نسخه ۲ پلی‌استیشن.^۳

یکی دیگر از شرکت‌هایی که در این زمینه فعالیت می‌کند مایکروسافت است. کنسول بازی ایکس‌باکس به همراه حسگر کینکت نمونه‌ی مشابه پلی‌استیشن و دوربین آن است. کینکت با تشخیص حرکات بدن انسان و ارسال این فرامین به ایکس‌باکس کاربر را قادر می‌سازد تا بتواند تجربه‌ی جدیدی از بازی‌های

^۱www.playstation.com

^۲www.playstation.com

^۳www.playstation.com

ویدئویی را داشته باشد. بازی‌های مخصوص حسگر کینکت با رنگی متفاوت از سایر بازی‌های ایکس باکس مشخص می‌شوند. امروزه از کینکت نه فقط در زمینه بازی بلکه در زمینه‌های بینایی ماشین و پردازش-تصویر نیز استفاده می‌شود. در شکل ۳-۱ حسگر کینکت و در شکل ۴-۱ کنسول ایکس باکس وان را مشاهده می‌کنید.



شکل ۳-۱ حسگر کینکت ایکس باکس وان!



شکل ۴-۱ کنسول بازی ایکس باکس وان^۲.

۲-۱ ضرورت انجام پژوهش

امروزه تحت تاثیر زندگی ماشینی، بسیاری از مردم فعالیت فیزیکی کمی دارند و عمدتاً ساعات طولانی پشت میز نشسته‌اند. عدم تحرک جسمی، سبب بروز بیماری‌های گوناگونی در بشر شده است. لذا طراحان بازی‌های کامپیوتری به دنبال راه حلی برای فعال کردن عضلات بازیگران هستند. در این بخش به

^۱www.amazon.com

^۲www.xbox.com

بررسی بازی‌های کنترل شونده با حرکات دست و چهره می‌پردازیم که می‌توانند نقش موثری در ایجاد تحرک روزانه ایفا کنند. موضوع دیگری که بررسی خواهیم کرد مزایا و ضرورت حضور در صنعت بازی-سازی است. در انتها، لزوم توسعه بازی‌ها برای دستگاه‌های موبایل را بررسی خواهیم کرد.

۱-۲-۱ بازی‌های کنترل شونده با حرکات بدن

در این بخش به بررسی اهمیت بازی‌های کنترل شونده با حرکات بدن خواهیم پرداخت. همان طور که گفته شد امروزه افراد زمان زیادی را صرف نشستن در پشت میزها و یا انجام کارهایی می‌کنند که می‌تواند باعث ایجاد خستگی و فشار در اندامهای مختلف شود. یک عامل مهم در این مسئله وضعیت قرارگیری بدن^۱ در انجام امور روزانه است. منظور، وضعیتی است که ما بدن خود را نگه می‌داریم؛ ایستاده، نشسته یا دراز کشیده. وضعیت قرارگیری بدن، مستقیماً بر روی سلامتی تاثیر می‌گذارد. علائمی که با عناوین آسیب ناشی از فشار تکراری^۲ و اختلالات اسکلتی-عضلانی مرتبط با کار^۳ شناخته می‌شوند، شامل یک گروه از بیماری‌های شغلی هستند که در سراسر جهان شناخته شده‌اند که اعصاب، عضله‌ها، مفاصل و تاندون‌ها را تحت تاثیر قرار می‌دهند. به طور کلی، این بیماری‌ها از شرایط نامناسب کار ناشی می‌شوند که اغلب در اثر اضافه کاری ایستا یا پویا در موقعیت ارگونومیک نامناسب ایجاد می‌شود [۲].

قسمت‌هایی از بدن که بیشتر تحت تاثیر هستند عبارتند از: دست‌ها، مچ دست‌ها، ساعدها، آرنج‌ها، بازوها، شانه‌ها، گردن و کتف. طرز ایستادن نامناسب اغلب تحت تاثیر تنظیمات نادرست محیط کار است به عنوان مثال ارتفاع صندلی که بر روی موقعیت پاها، بازوها و مچ دست‌ها تاثیر می‌گذارد. این موارد برای افرادی که با رایانه‌ها کار می‌کنند به مراتب بیشتر رخ می‌دهد [۲].

با بهبود و حفظ دامنه حرکتی خوب در مفاصل، می‌توان به سلامت بدن کمک کرد. سطح انعطاف پذیری مفاصل در افراد مختلف، متفاوت است. همچنین سن، جنسیت و ژنتیک می‌توانند عوامل تعیین کننده

¹Posture

²Repetitive Strain Injury

³Work-Related Musculoskeletal Disorders

در این بیماری‌ها باشند. به هر حال تمرینات دوره‌ای مثل کشش، اغلب انعطاف پذیری مفاصل و عضله‌ها را بهبود می‌دهد [۲]. با توجه وقوع مکرر بیماری‌های ذکرشده و کمک به اقدامات پیش‌گیرانه که می‌تواند افراد را نسبت به مشکلات ناشی از بیماری‌های شغلی هشدار دهد [۲]، توجه به بازی‌هایی که در آن‌ها کاربر با حرکات بدن خود به کنترل بازی می‌پردازد یک مسئله مهم بوده و یک اقدام پیش‌گیرانه محسوب می‌شود.

۱-۲-۲ مزایا و ضرورت حضور در صنعت بازی‌سازی

صنعت بازی‌سازی یکی از صنایع پیشرو جهان است که در سال‌های ۲۰۰۹ تا ۲۰۱۲، صنعت بازی‌ها ۴ برابر بیشتر از اقتصاد ایالات متحده رشد داشت. در سال ۲۰۱۶ ارزش کل فروشگاه‌های بازی ۹۹٫۶ میلیارد دلار آمریکا بود و طبق پیش‌بینی‌ها در سال ۲۰۲۰ صنعت بازی‌های ویدئویی در اولین جایگاه قرار خواهد گرفت [۳].

با گذشت سال‌ها و پیشرفت در حوزه‌ی گرافیک و کیفیت بازی‌های ویدئویی، موبایل‌های جدید و سیستم‌عامل‌های برخط، صنعت بازی‌سازی را متحول کرده‌اند. صنعت بازی‌سازی در مقایسه با سایر بخش‌های صنعت سرگرمی، مثل تلویزیون و سینما، کتاب‌ها، فیلم و موسیقی، سریع‌تر رشد می‌کند [۳]. بازی‌های ویدئویی بسیار پویا هستند و به‌لطف دیجیتال‌سازی و پیشرفت صنعت در حال رشد هستند. این گسترش، بازار سرمایه را در صنعت شوکه کرده‌است و رقابت‌هایی شبیه اسکار برای بهترین بازی‌ها ایجاد شده‌است که سالانه هدایای زیادی را به بهترین بازی‌ها اهدا می‌کنند [۳].

امروزه، کشورهای زیادی در این بازار بزرگ فعال هستند. یکی از کشورهای پیشرو در این صنعت فنلاند است. فنلاند یکی از سه کشور موفق در زمینه‌ی توسعه بازی در اروپا است و می‌تواند یک الگویی برای

سایر کشورها همچون ایران باشد. طبق گزارش‌ها، درآمد حاصل از صنعت بازی از بازار دیجیتال تلفن-هوشمند در ایران ۲۶۹ میلیون دلار در سال ۲۰۱۷ تخمین زده شد که در مقایسه با گردش مالی میلیارد دلاری صنعت بازی در جهان، سهم ایران بسیار اندک است. با در نظر گرفتن نوپا بودن این صنعت در ایران، و رشد سریعی که فنلاند در این صنعت در مدت زمان کوتاه داشته است، ارزیابی عوامل توسعه صنعت بازی در فنلاند می‌تواند کمک بزرگی باشد برای توسعه‌دهندگان کشورهای که تجربه پایینی در صنعت بازی‌سازی دارند [۳].

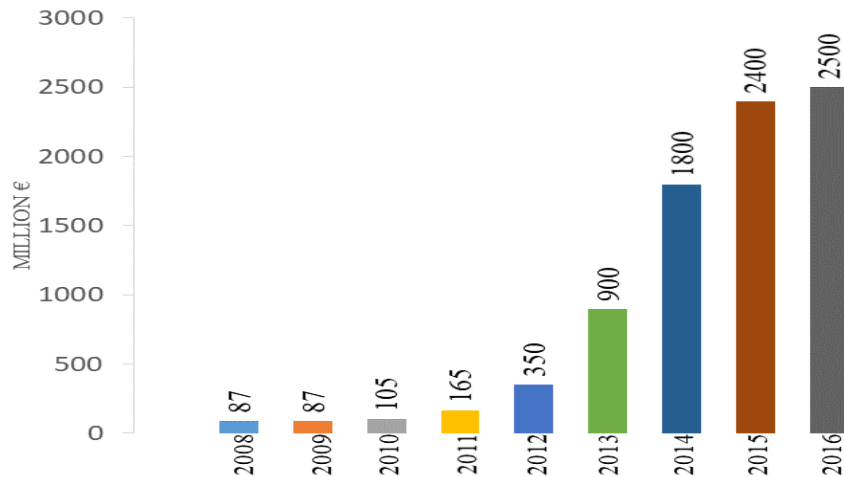
اواخر دهه‌ی ۱۹۹۰ شروع یک رونق حرفه‌ای صنعت بازی‌سازی در فنلاند بوده است. با دستگاه‌های موبایل جدید که در اواخر دهه‌ی ۲۰۰۰ با اولین صفحه نمایش‌های لمسی عرضه شدند صنعت بازی‌سازی با سرعت بیشتری تغییر کرد. موفقیت شرکت سرگرمی Rovio با بازی پرندگان خشمگین^۱ در سال ۲۰۰۹ که تا به امروز شناخته‌شده‌ترین بازی در جهان محسوب می‌شود، مثال‌زدنی است. این بازی بیش از یک میلیارد بار دانلود شده است. موفقیت Rovio باعث شد تا در فنلاند افراد زیادی شروع به فعالیت در این حوزه نمایند. در سال ۲۰۱۲ بازی مسابقه بالا رفتن از تپه^۲ ساخت شرکت Fingersoft در فروشگاه‌های app سیستم‌عامل IOS و play، بیش از ۱۰۰ میلیون بار دانلود شد. در همان سال شرکت Supercell دو بازی Hay Day و جنگ قبایل^۳ را در فروشگاه عرضه کرد که این دو بازی برای اندروید و IOS تا اواخر سال ۲۰۱۴ به بازی‌هایی تبدیل شدند که بیشترین تعداد دانلود را داشتند [۳].

در سال ۲۰۱۷ شرکت‌های سطح متوسط فنلاند نیز کاملاً تقویت شدند و این سال سومین سال پیاپی بود که گردش مالی این صنعت در فنلاند از ۲ میلیارد یورو گذشت. شکل ۱-۵ گردش مالی سالانه این صنعت را نشان می‌دهد [۳].

^۱Angry Birds

^۲Hill Climb Racing

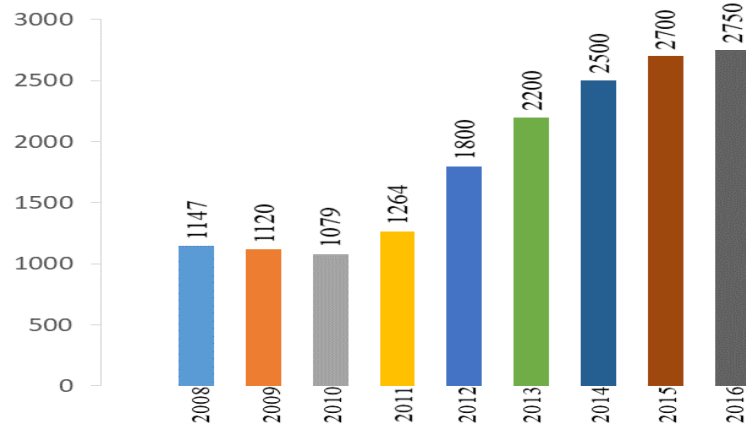
^۳Clash of Clans



شکل ۵-۱ گردش مالی سالانه فنلاند در صنعت بازی سازی [۳].

یک عامل دیگر که کاملاً توسعه بازی را نشان می‌دهد تعداد اشتغال در این صنعت است. تعداد افراد شاغل در این حوزه به ۳۰۰۰ نفر رسیده است. شکل ۶-۱ تعداد شاغلان سالانه در صنعت بازی سازی در

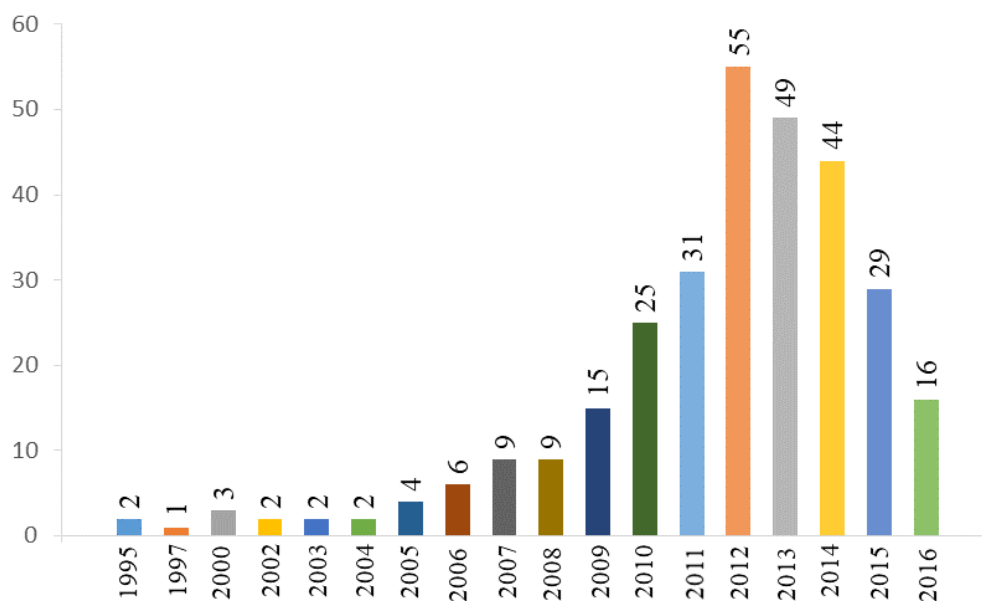
فنلاند را نشان می‌دهد [۳].



شکل ۶-۱ تعداد شاغلان صنعت بازی در فنلاند [۳].

اگرچه توسعه بازی‌ها در فنلاند این کشور را بسیار مشهور کرده است اما رشد آن در سال‌های اخیر کاهش یافته است. شکل ۷-۱ کاهش شرکت‌های فعال در صنعت بازی سازی در چندسال اخیر فنلاند را نشان

می‌دهد.

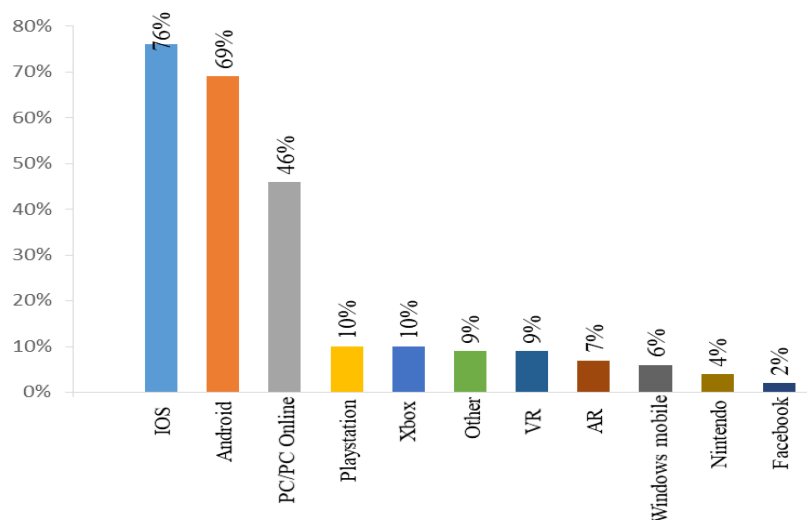


شکل ۱-۷ تعداد شرکت‌های فعال در صنعت بازی‌سازی در فنلاند [۳].

۱-۲-۳ مزایای توسعه بازی برای دستگاه‌های موبایل

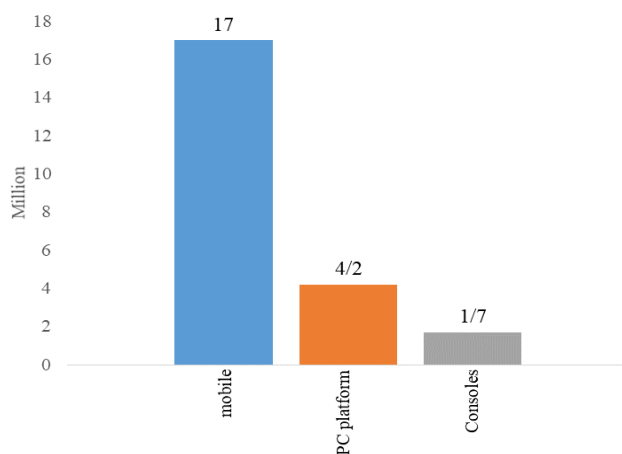
به منظور بررسی عامل لزوم توسعه بازی برای دستگاه‌های موبایل از تاریخ کشور فنلاند در این حوزه استفاده خواهیم کرد.

در حال حاضر، بیشتر تمرکز شرکت‌های بازی‌سازی فنلاند بر روی دستگاه‌های با سیستم عامل IOS و اندروید و رایانه و بازی‌های برخط رایانه است. شکل ۱-۸ محبوبیت سیستم‌عامل‌های مختلف در استودیوهای بازی‌سازی فنلاند را نشان می‌دهد [۳].



شکل ۸-۱ محبوبیت سیستم‌عامل‌های مختلف در شرکت‌های بازی‌سازی فنلاند [۳].

به دلیل اینکه ۶۵ درصد از سهم بازار در صنعت بازی‌سازی در ایران در اختیار بازی‌های موبایل است، متوجه می‌شویم بازیگران ایرانی، به بازی‌های موبایل علاقمند هستند. شکل ۹-۱ تعداد بازیگران برای موبایل، رایانه و دیگر کنسول‌ها را نشان می‌دهد. با وجود نبود قوانین حق‌چاپ، ساخت برنامه در ایران، یکی از بهترین راه‌های کسب درآمد است [۳].



شکل ۹-۱ تعداد بازیگران موبایل، رایانه و کنسول‌ها در ایران [۳].

حال می‌خواهیم یک مقایسه از لحاظ هزینه تهیه دستگاه‌های مختلف داشته باشیم. به منظور انجام یک بازی که با حرکات دست بتوان شخصیت درون بازی را کنترل کرد چند انتخاب وجود دارد، می‌توان از کنسول بازی پلی استیشن ۴ استفاده نمود که طبیعتاً نیاز به دوربین پلی استیشن نیز هست و هزینه تهیه این مجموعه حدود ۱۴ میلیون تومان است. انتخاب دیگر کنسول ایکس‌باکس وان به همراه حسگر کینکت است که هزینه تهیه آن‌ها حدود ۹ میلیون تومان است [۴]. انتخاب دیگر، گوشی موبایل اندرویدی است که با قیمت‌های متنوعی موجود است که طبیعتاً انتخاب‌هایی وجود دارد که از هر دو گزینه‌ی ذکر شده، مناسب‌تر است.

به عنوان عامل دیگر در توسعه بازی‌ها برای موبایل می‌توان قابل حمل بودن دستگاه‌های موبایل را نیز در نظر گرفت.

۱-۳ اهداف پایان‌نامه

در قسمت‌های قبل با فواید بازی‌های کنترل‌شونده با حرکات بدن آشنا شدیم. همچنین از بررسی‌های انجام شده متوجه شدیم که طرفداران بازی‌های موبایلی بیشتر هستند. ایده توسعه بازی موبایلی که با حرکات بدن کنترل شود ایده جذابی است که در این پایان‌نامه به آن پرداخته ایم. در این پژوهش قصد داریم یک بازی طراحی کنیم که کاربر بتواند با گوشی موبایل خود بازی کرده و تمام فرامین را از طریق حرکات سر و دست خود، به محیط بازی انتقال دهد. در این بازی شخصیت درون بازی باید یک مسیر را که شامل موانع مختلفی است بپیماید و به انتها برسد.

۱-۴ پیش‌نیازهای پژوهش

به منظور انجام این بازی لازم است که یک تلفن هوشمند اندرویدی میان‌رده داشته باشیم که از دوربین جلو بهره‌مند بوده و حداقل نسخه اندروید ۴/۱ باشد.

۵-۱ ساختار پایان نامه

در فصل ۲ به مرور کارهای پیشین، روش‌ها و نتایج آن‌ها خواهیم پرداخت. فصل ۳ شامل مباحث نظری این پایان نامه می‌شود. مراحل انجام و روش‌های مورد استفاده در این پایان نامه در فصل ۴ بررسی می‌شود و نهایتاً در فصل ۵ نتایج این پژوهش، پیشنهادات و کارهای آینده ذکر شده‌اند.

فصل دوم

مستشرقین و روش

از جستجوهای به عمل آمده در مراجع معتبر و سایت‌های انتشار مقالات، متوجه شدیم که تاکنون تحقیقات چندانی در زمینه‌ی توسعه بازی‌های کنترل شونده با حرکات بدن برای دستگاه‌های موبایل، انجام نگرفته است. به همین منظور، سعی کردیم که از تحقیقات مرتبط با این موضوع کمک بگیریم. در ابتدا مقالاتی را بررسی خواهیم کرد که اینگونه بازی‌ها را با استفاده از کینکت طراحی کرده‌اند. و سپس به بررسی روش‌های شناسایی چهره، حالات دست و شناسایی اشیاء خواهیم پرداخت. در پایان نیز به نتیجه‌گیری می‌پردازیم.

۱-۲ تحقیقات انجام شده با کینکت

چینگ‌تانگ و همکارانش [۵]، با استفاده از تمام مزایای حسگر کینکت یک بازی آموزشی طراحی کرده‌اند. به منظور طراحی و پیاده‌سازی بازی ابتدا به سراغ انتخاب محتوای آموزش و سازمان‌دهی آن مطابق با اصول طراحی رفته‌اند. این بازی مخصوص دانش‌آموزان ابتدایی کلاس دوم و سوم توسعه داده شده‌است. این دانش‌آموزان به تصاویر فکر می‌کنند، فعال هستند، علاقمند به رنگ‌های روشن، بازی کردن، ایفای نقش و ستایش دیگران هستند. مطابق این ویژگی‌ها، از رنگ سبز روشن به عنوان پس‌زمینه و از رنگ زرد روشن به عنوان رنگ اشکال به منظور جلب توجه استفاده شده‌است. بازیکن با برخورد به اشکال مختلف امتیاز دریافت می‌کند. اگر موفق شود یک صدای تشویق پخش خواهد شد. این بازی آموزشی به منظور درک رنگ‌ها و شکل‌ها توسط کاربر انجام می‌شود [۵].

قسمت طراحی، اساساً به طراحی داستان بازی، نقش‌ها، حرکات مربوط می‌شود. هدف این تحقیق درک بهتر رنگ‌ها، شکل‌ها و خلاص‌شدن از مضرات رایانه است. تصویری از صحنه بازی را در شکل ۱-۲ مشاهده می‌کنید [۵].



شکل ۱-۲ تصویری از محیط بازی [۵].

با استفاده از فرمان صوتی شروع مجدد^۱ یک بازی جدید شروع می‌شود. با استفاده از کج کردن بدن به طرفین می‌توان جهت حرکت شخصیت بازی را تعیین کرد. خم شدن به جلو سرعت را افزایش و به عقب خم شدن سرعت را کاهش می‌دهد. در قسمت توسعه بازی از بسته توسعه نرم‌افزاری^۲ کینکت برای ویندوز، زبان سی شارپ در محیط ویژوال استودیو ۲۰۱۰، نرم‌افزار ۳DMax و فوتوشاپ استفاده شده است [۵].

در [۶]، یک سامانه‌ی شناسایی حالت دست طراحی شده است که می‌تواند حرکات هر دو دست را ردیابی و حالت‌های آن‌ها را شناسایی کند. زمانی که حرکات هر دو دست شناسایی می‌شود، کاربر می‌تواند بازی ویدئویی مسابقه‌ای مشهور نیاز به سرعت^۳ را کنترل کند. ویژگی‌های برجسته‌ای که این سامانه را نسبت به دیگر برنامه‌های شناسایی حالات دست متمایز می‌سازد شامل موارد زیر است [۶]:

- این سامانه می‌تواند در پس زمینه‌های شلوغ مثل حضور چندین نفر نیز پاسخ‌گو باشد و نیازی به پوشیدن دستگاه‌های علامت‌گذاری توسط کاربر نیست.
- رویکرد این پژوهش پیچیدگی محاسباتی کمی دارد که باعث می‌شود سرعت پاسخ‌گویی خوبی داشته باشد و در عمل بتوان در بازی‌های ویدئویی از آن استفاده کرد.

^۱Restart

^۲SDK

^۳Need for Speed

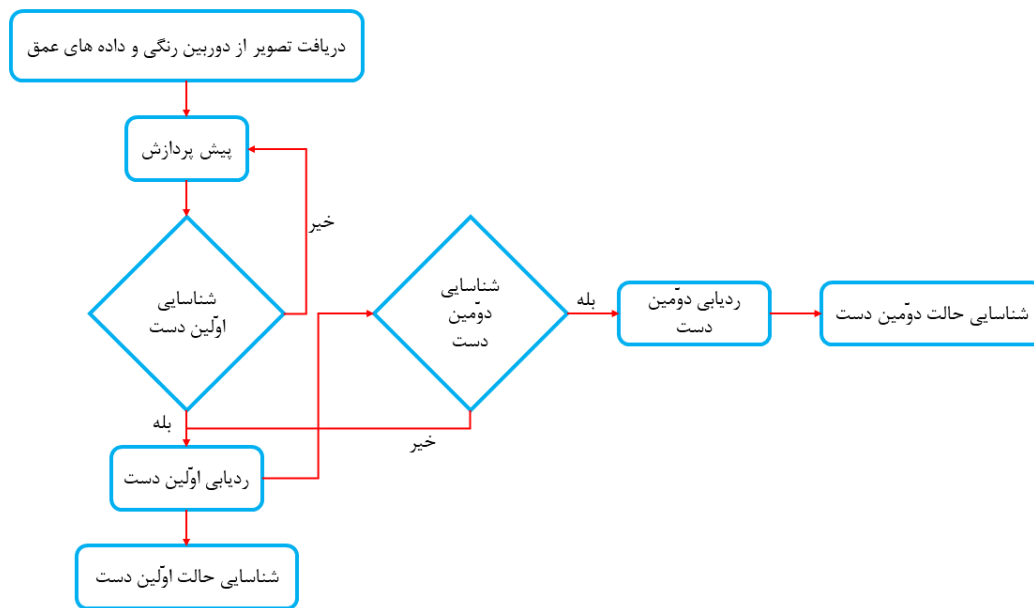
اجزای مورد استفاده در این پژوهش شامل موارد زیر هستند [۶]:

- استفاده از دوربین رنگی سه کاناله به منظور دریافت جریان ویدئو^۱ و دوربین عمق برای ایجاد نقشه عمق در حسگر کینکت با تفکیک پذیری ۴۸۰*۶۴۰ و نرخ قاب ۳۰^۲ قاب بر ثانیه.
- یک رایانه با پردازنده Intel Core i5-2320 و فرکانس ۳ گیگاهرتز و حافظه ۸ گیگابایت.
- سیستم عامل ویندوز ۷، ۶۴ بیتی.
- زبان برنامه نویسی C++ و کتابخانه OpenCV نسخه ۲،۴،۵.
- بازی ویدئویی نیاز به سرعت نسخه ۲۰۱۲ به منظور آزمایش این سامانه.

در شکل ۲-۲ الگوریتم پیشنهادی سامانه شناسایی حالت‌های دست نمایش داده شده است. شناسایی حالت یک دست از پیدا کردن موقعیت دست با استفاده از دوربین رنگی و داده‌های عمق، شروع می‌شود. سپس به منظور دستیابی به مسیر دست، حرکات آن را دنبال می‌کند. و سپس شناسایی حالت آن انجام می‌گیرد. برای شناسایی حالت دو دست، ابتدا موقعیت یک دست پیدا می‌شود و به منظور شناسایی حالت آن، ردیابی آن صورت می‌گیرد. سپس با استفاده از اطلاعات شکل و موقعیت دست شناسایی شده، موقعیت دست دیگر مشخص می‌شود و تشخیص حالت دو دست صورت می‌گیرد [۶].

^۱Video Stream

^۲Frame

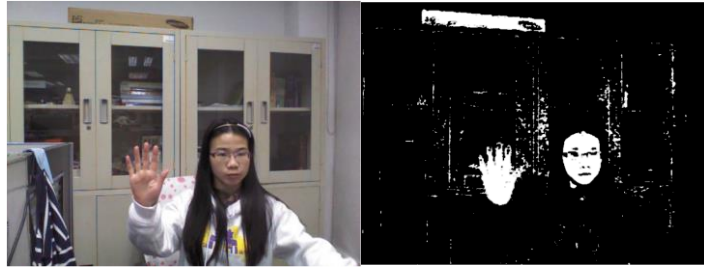


شکل ۲-۲ نمودار شناسایی حالت دست. شناسایی حالت یک دست (شاخه سمت چپ) و شناسایی حالت دو دست (شاخه سمت راست) [۶].

قسمت پیش پردازش مربوط به اعمال الگوریتم متعادل سازی رنگ می شود. که این الگوریتمها به منظور شناسایی در شرایط نوری ضعیف استفاده می شوند. در این پژوهش از روش جهان خاکستری استفاده شده است. که رابطه ۱-۲ این روش را تشریح می کند [۶].

$$r_i = \frac{avg(g)}{avg(r)} r_i \cdot g_i = g_i \cdot b_i = \frac{avg(g)}{avg(b)} b_i \quad (1-2)$$

در این رابطه، متغیرهایی که با avg مشخص شده اند برابر میانگین همان کانال رنگ هستند. و متغیرهایی که با اندیس i مشخص شده اند، برابر مقدار پیکسل همان کانال هستند. شکل ۲-۳ نتیجه ی اعمال پیش پردازش را نشان می دهد [۶].



(الف)

(ب)

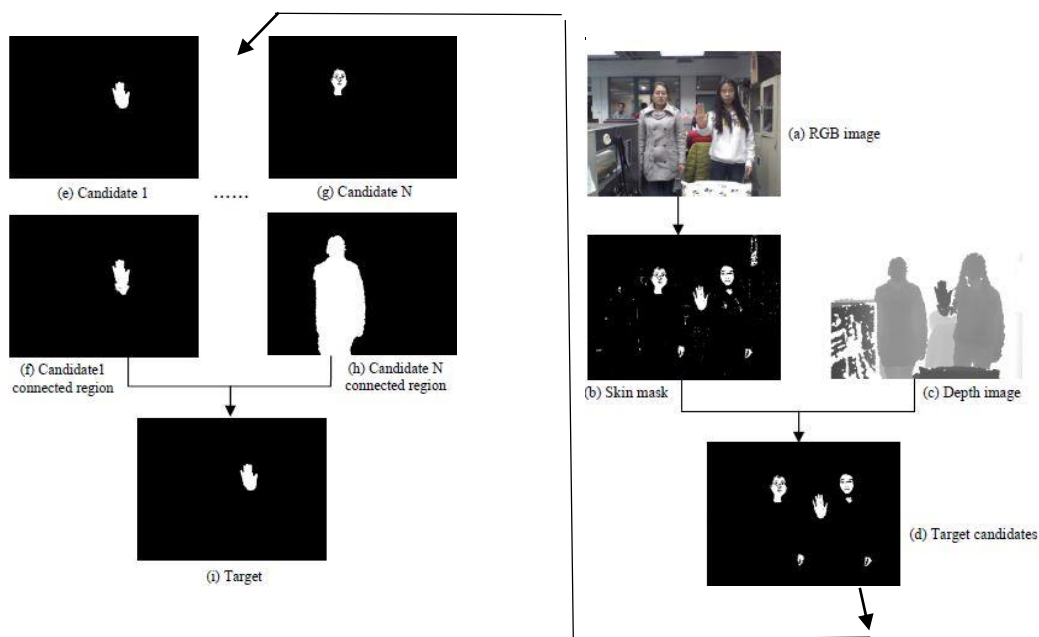


(پ)

(ت)

شکل ۳-۲ تاثیر متعادل سازی رنگ. (الف) تصویر اصلی، (ب) شناسایی پوست تصویر اصلی، (پ) تصویر متعادل شده، (ت) شناسایی پوست تصویر متعادل شده [۶].

در فرآیند شناسایی اولین دست، پس از شناسایی نواحی نامزد برای پوست (تصویر d از شکل ۲-۴)، از سنسور عمق برای انتخاب فرد مورد نظر استفاده می شود (تصویر e و g از شکل ۲-۴). سپس دوباره با استفاده از سنسور عمق اجزای متصل صورت و دست بدست می آیند (تصویر f و h از شکل ۲-۴). سپس برای تشخیص دست از این منطق استفاده می شود: نسبت شی شناسایی شده به نسبت اجزای متصل همان شی باید مقداری نزدیک به یک باشد. یعنی نسبت ناحیه ی شی تصویر e به ناحیه شی تصویر f، با نسبت ناحیه شی تصویر g، به ناحیه شی تصویر h، مقایسه می شود. این فرآیند در شکل ۲-۴ نشان داده شده است [۶].









شکل ۴-۲ فرآیند شناسایی دست اول [۶].

در فرآیند شناسایی دست دوم از همان عمق و موقعیت دست اول استفاده می‌شود. که به سادگی دست دوم مشخص می‌شود. برای ردیابی دست، پس از شناسایی آن در یک فریم ناحیه‌ی آن تحت عنوان ناحیه‌ی مورد علاقه در فریم بعدی در نظر گرفته می‌شود و فرآیند شناسایی فقط در همان ناحیه‌ی مورد علاقه انجام می‌گیرد و این باعث افزایش سرعت پردازش خواهد شد [۶].

پس از شناسایی دست، به طبقه‌بندی پرداخته‌اند. برای تمایز بین کف دست و مشت، یک مجموعه داده شامل ۵۰۰ تصویر کف دست و مشت جمع‌آوری کرده‌اند و به صورت دستی آن‌ها را از پس زمینه جدا کرده‌اند. سپس از ماشین بردار پشتیبان به منظور طبقه‌بندی استفاده شده‌است. در جدول ۱-۲ حالت‌های مختلف کنترل بازی ویدئویی نشان داده شده‌است. عملکرد شناسایی این سامانه در جدول ۲-۲ آورده شده‌است [۶].

جدول ۱-۲ حالت های مختلف کنترل بازی ویدئویی [۶].

Gesture	Meaning/Function
both palms 	speed up
both fists 	back off
left fist, right palm 	free run
left palm, right fist 	hand brake
left hand above right hand 	tum right
right hand above left hand 	tum left

جدول ۲-۲ عملکرد شناسایی حالت های مختلف دست [۶].

Gesture	Recognition Rate (%)
both palm	99
both fist	97
left palm, right fist	98
left fist, right palm	98
right hand above left hand	99
left hand above right hand	99

در [۷] فرزین نژاد و راستی، یک بازی رایانه‌ای-ورزشی طراحی کرده‌اند که با استفاده از حسگر کینکت تمرین صحیح سایه‌زنی را بدون حضور مربی ممکن می‌کند و علاوه بر تغییر طبیعت کسالت‌آور این تمرینات، با ثبت دقیق اطلاعات بازیکن، به مربی برای برنامه‌ریزی دقیق برای پوشش نقاط ضعف بازیکن کمک می‌کند. هدف از ساخت این بازی ویدئویی آموزش اجرای صحیح فن روی دست توسط ورزشکار راست دست است. در روش سنتی تمرین سایه برای انجام فن روی دست، بازیکن باید در مقابل آینه بایستد و تمرین سایه فن روی دست را انجام دهد مطابق شکل ۲-۵ (الف) که تمرینی خسته کننده محسوب می‌شود به همین منظور این بازی ویدئویی را توسعه داده‌اند [۷].



(ب)

(الف)

شکل ۲-۵ (الف) تمرین سایه به روش سنتی. (ب) تمرین سایه به روش پیشنهادی [۷].

در این بازی برای تشخیص بازیکن، از کینکت استفاده می‌شود. این فناوری یک محیط تعاملی برای ثبت حرکات بدن بدون نیاز به کنترل کننده را فراهم می‌کند. کینکت می‌تواند در حالت محدوده پیش فرض، افراد را در فاصله‌ی ۰,۸ تا ۴ متر تشخیص دهد؛ کاربر این توانایی را خواهد داشت که بازوهای خود را در دامنه ۱,۲ تا ۳,۵ متر حرکت دهد. مشخصات این بازی به شرح زیر است [۷]:

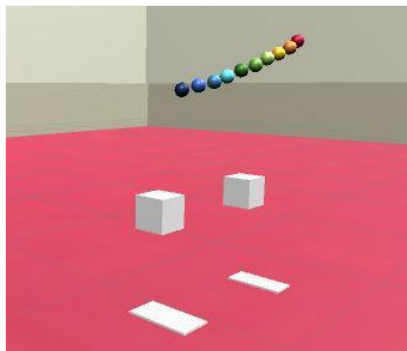
- برای ایجاد این بازی از موتور بازی سازی یونیتی نسخه‌ی ۵,۶,۰ استفاده شده‌است.
- توسط بسته توسعه نرم‌افزاری کینکت برای ویندوز داده‌های دریافت شده توسط کینکت (حرکات کاربر)، با سنسور ردیابی شده و به مدل انسانی طراحی شده در موتور بازی سازی یونیتی منتقل می‌شود که باعث تطابق مفاصل انسان واقعی با مفاصل انسان داخل بازی می‌شود. با توجه به ابعاد بدن کاربر ردگیری شده، ابعاد بدن مدل انسان محاسبه شده و مورد استفاده قرار می‌گیرد.
- فیلمی از نحوه‌ی صحیح انجام فن روی دست در شکل ۲-۶ نشان داده شده‌است که توسط مربی سطح یک جهانی و بازیکن مطرح ایران، گرفته شده‌است و مبنای این بازی خواهد بود. با ضبط و ایجاد انیمیشن از حرکت حرفه‌ای انجام شده، مختصات (x,y,z) نقاط تشکیل دهنده‌ی مسیر

حرکت راکت و دست در فن روی دست، استخراج می‌شود تا مسیر و دامنه‌ی صحیحی که بازیکن باید انجام دهد، به وسیله این نقاط رسم شود.



شکل ۶-۲ فیلم نحوه‌ی صحیح انجام فن روی دست [۷].

برای ساخت مسیر حرکت ۱۰ گلوله کروی شکل ایجاد می‌شود و آن‌ها در مختصات بدست آمده جایگذاری خواهند شد که هر کدام از گلوله‌ها ۱۰ امتیاز دارد. بازیکن باید برای برخورد راکت با این گلوله‌ها هر دو بازوی خود را مقابل بدن خود قرار دهد و آرنج خود را ۹۰ تا ۱۱۰ درجه خم کند همان‌طور که در شکل ۷-۲ نشان داده شده‌است [۷].



شکل ۷-۲ دامنه حرکتی روی دست [۷].

در طول بازی، بازیکن باید تلاش کند که بین راکت و گلوله‌ها برخورد ایجاد کند که بر مبنای تعداد گلوله که راکت به آن‌ها برخورد می‌کند از ۰ تا ۱۰۰ به بازیکن امتیاز داده می‌شود و در صورت برخورد به ترتیب راکت با همه گلوله‌ها و کسب امتیاز ۱۰۰، بازیکن تشویق صوتی و تصویری خواهد شد. به علاوه

به ازای هر ۱۰ بار تمرین سایه، تعداد حرکات کاملاً درست (دریافت ۱۰۰ امتیاز است) به بازیکن بازخورد صوتی و تصویری مطابق جدول ۲-۳ داده خواهد شد [۷].

جدول ۲-۳ بازخوردهای داده شده به بازیکن براساس دفعات تکرار صحیح [۷].

بازخورد	تعداد دفعات صحیح تمرین (از ۱۰ مرتبه)
نیاز به تلاش بیشتر	صفر تا ۲
متوسط	۳ تا ۵
خوب	۶ تا ۹
بسیار خوب	۱۰

2-2 شناسایی چهره، حالت‌های دست و شناسایی اشیا

برای شناسایی چهره و حالت‌های مختلف دست و شناسایی اشیا راه‌کارهای مختلفی وجود دارد که در این بخش به بررسی برخی از این روش‌ها خواهیم پرداخت. روش‌های مبتنی بر شبکه‌های عمیق، طبقه‌بندی‌های آبشاری و جریان نوری^۱ ماشین بردار پشتیبان^۲ و پردازش تصویر بررسی خواهند شد. در ادامه، ابتدا روش‌های شناسایی چهره و سپس شناسایی حالت‌های دست را بررسی خواهیم کرد.

۱-۲-۲ شناسایی چهره

در [۸]، با استفاده از روش مبتنی بر شبکه‌های عمیق به انجام شناسایی چهره با استفاده از گوشی موبایل پرداخته‌اند. واضح است که شناساگرهای قدرتمند چهره مبتنی بر شبکه‌های عمیق که برای اجرا بر روی رایانه‌های رومیزی طراحی شده‌اند، گزینه‌ی مناسبی برای دستگاه‌های موبایل نیستند. دو اختلاف بین این دو مسئله، در زیر ذکر شده‌است [۸]:

(۱) اختلاف در سخت افزار و نرم افزار

^۱Optical Flow

^۲SVM

- سخت افزار لازم برای شبکه‌های عمیق گرافیک‌های قدرتمند انویدیا^۱ هستند. تراشه گرافیکی موبایل از قدرت کمتری برخوردار است و از این رو، الگوریتم‌ها باید ساده‌تر باشند.

- بیشتر شبکه‌های عمیق از بستر کودا^۲ استفاده می‌کنند، اما تا به امروز انویدیا اقدام به ساخت تراشه گرافیک موبایل نکرده‌است. بنابراین، از کودا پشتیبانی نمی‌کنند و به یک بستر دیگری احتیاج است.

۲) اختلاف در مجموعه داده

- مجموعه‌ی داده‌های چهره، شامل تصاویری هستند که چندین چهره کوچک در تصاویر وجود دارند، اما دوربین جلوی گوشی زمانی استفاده می‌شود که کاربر از گوشی استفاده می‌کند، بنابراین یک چهره با اندازه بزرگ خواهد داشت.

در این پژوهش اقدامات زیر صورت گرفته است [۸]:

- یک الگوریتم مبتنی بر شبکه‌های عمیق برای دستگاه‌های موبایل توسعه داده شده که مانند شناساگرهای چهره‌ی عمومی قدرتمند نیست. اما به خوبی تنظیم شده‌است تا بر روی دستگاه‌های موبایل قابل اجرا باشد.
- بیشتر پیاده‌سازی‌های شبکه‌های عمیق در بستر کودا انجام می‌شود که در دستگاه‌های موبایل همچین قابلیت‌ی وجود ندارد. در این پژوهش، به منظور اجرای الگوریتم‌های شبکه عمیق با استفاده از تراشه گرافیکی، کتابخانه‌ها را در OpenCL و RenderScript توسعه داده‌اند.

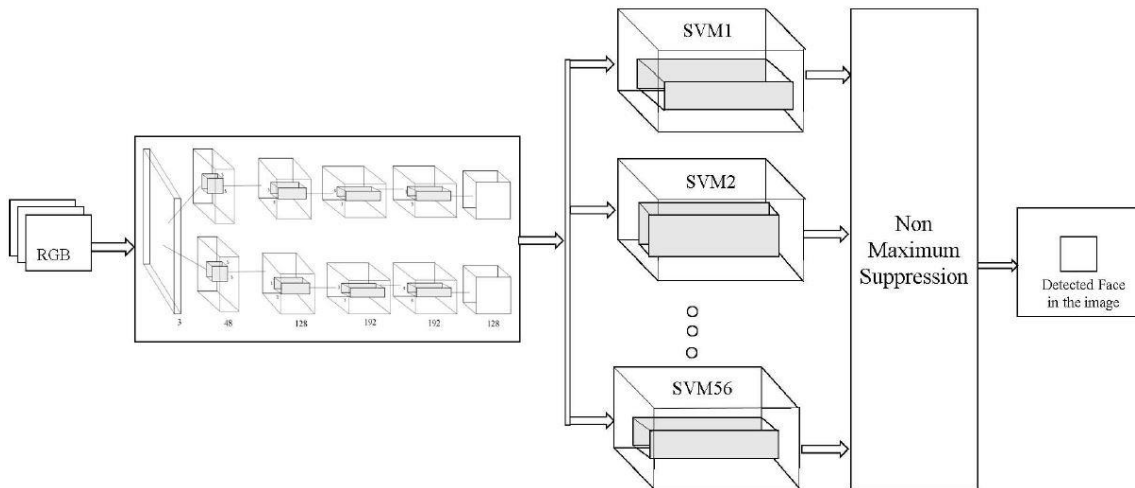
اولین مرحله الگوریتم شناسایی چهره با استفاده از موبایل و ویژگی‌های عمیق^۳ استخراج ویژگی‌های عمیق با استفاده از ۵ لایه‌ی اول شبکه AlexNet است. پنجره‌های متحرک با اندازه‌های مختلف به منظور شناسایی چهره‌هایی با اندازه‌های مختلف در نظر گرفته شده‌است. همچنین برای هر پنجره یک طبقه‌بند ماشین بردار پشتیبان در نظر گرفته شده‌است. سپس شناسایی همه‌ی ماشین بردار پشتیبان‌ها در کنار هم جمع می‌شوند و براساس معیار هم‌پوشانی برخی از نامزدها کنار زده می‌شوند. روند کلی این پژوهش در شکل ۲-۸ نشان داده شده‌است [۸].

^۱Nvidia

^۲CUDA

^۳Deep Features based Face Detection on Mobiles (DFFDM)

^۴Sliding Windows



شکل ۲-۸ روند کلی الگوریتم شناسایی چهره با استفاده از ویژگی‌های عمیق و دستگاه‌های موبایل [۸].

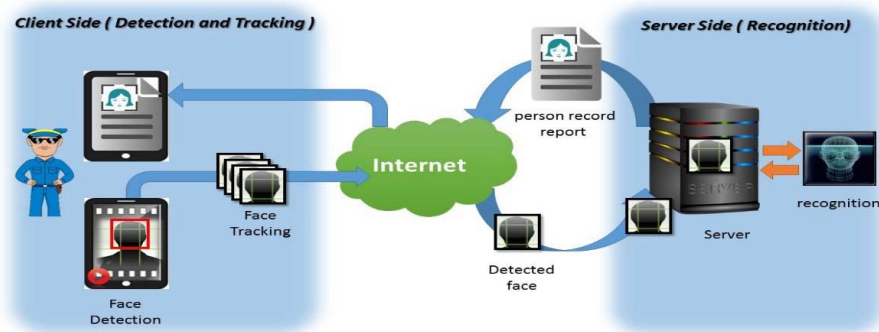
از مجموعه داده‌ی UMD-AA برای آموزش استفاده شده است که مجموعه‌ای از فیلم‌ها را شامل می‌شود و این فیلم‌ها از ۵۰ فرد مختلف در شرایط نوری مختلف تهیه شده‌اند. از ۵۶ پنجره‌ی متحرک با اندازه‌های مختلف برای شناسایی چهره‌هایی با اندازه‌های مختلف استفاده شده‌است. به همین منظور ۵۶ ماشین بردار پشتیبان برای شناسایی به کار رفته‌است [۸].

در یکبار عبور یک تصویر 360×640 از این شبکه، زمانی که از یک رایانه با پردازنده Intel Core i7 و پردازنده گرافیکی Nvidia GeForce GTX 850M استفاده می‌شود، ۱ ثانیه زمان لازم است در صورتی که برای گوشی‌های موبایل مختلف این مقدار بسیار بیشتر است همان‌طور که در جدول ۲-۴ مشاهده می‌کنید. به علاوه مصرف انرژی در گوشی‌های موبایل به شدت افزایش می‌یابد به عنوان مثال برای گوشی موبایل Nexus 5 زمانی که به طور پیوسته این الگوریتم اجرا می‌شود 0.45 درصد در هر دقیقه از شارژ باتری کاسته می‌شود. در حالی که در حالت معمولی این مقدار 0.16 درصد در دقیقه است [۸].

جدول ۴-۲ زمان اجرای الگوریتم DFFDM بر روی گوشی‌های مختلف [۸].

Phone	Runtime	GPU	CPU
Moto G	36.7 s	Adreno 305	Qualcomm Snapdragon 400
HTC One (M7)	31.2 s	Adreno 320	Qualcomm Snapdragon 600
Samsung Galaxy S4	28.0 s	Adreno 320	Qualcomm Snapdragon 600
Nexus 5	11.9 s	Adreno 330	Qualcomm Snapdragon 800
LG G3	10.3 s	Adreno 330	Qualcomm Snapdragon 801
Nexus 6	5.7 s	Adreno 420	Qualcomm Snapdragon 805

الرفاعی و همکارانش [۹]، یک شناساگر جنایی را پیشنهاد داده‌اند. که یک سامانه شناسایی چهره بلادرنگ است که به منظور نظارت و براساس خدمت‌گذار-مشتری^۱ توسعه یافته‌است. روند کلی این پژوهش در شکل ۹-۲ نشان داده شده‌است [۹].



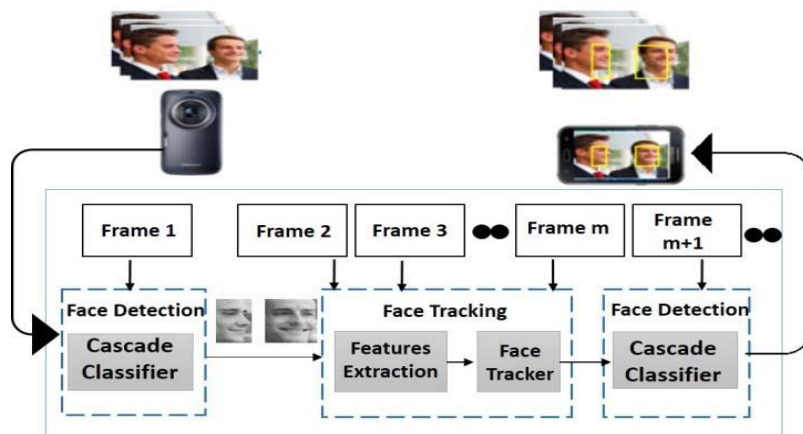
شکل ۹-۲ روند کلی سامانه پیشنهادی [۹].

در این سامانه ابتدا پلیس با استفاده از دوربین تلفن هوشمند خود یک ویدئو از فرد مشکوک ضبط می‌کند. عملیات تشخیص و ردیابی در سمت کاربر انجام می‌شود. سپس قاب‌های ویدئویی شامل چهره‌ی تشخیص داده شده و ردیابی شده به سرور ارسال می‌شود. یک سامانه شناسایی چهره در سرور وجود

^۱Client-server

دارد. سپس اطلاعات شخصی فرد شناسایی شده به تلفن هوشمند پلیس ارسال می‌شود. این پژوهش بر روی قسمت تشخیص و ردیابی چهره در قسمت مشتری تمرکز دارد [۹].

در سمت کاربر، هدف اصلی تشخیص و ردیابی چهره و نمایش آن به صورت بلادرنگ در زمان عملیات ضبط است. الگوریتم ویولا-جونز برای تشخیص چهره استفاده شده است. برای ردیابی از الگوریتم جریان نوری استفاده شده است چرا که می‌تواند درصد اشتباه مثبت را کاهش دهد. شکل ۲-۱۰ الگوریتم پیشنهادی را نشان می‌دهد. در اولین قاب تشخیص انجام می‌شود. و این عملیات هر m بار تکرار می‌شود، که این پارامتر قابل تنظیم است [۹].

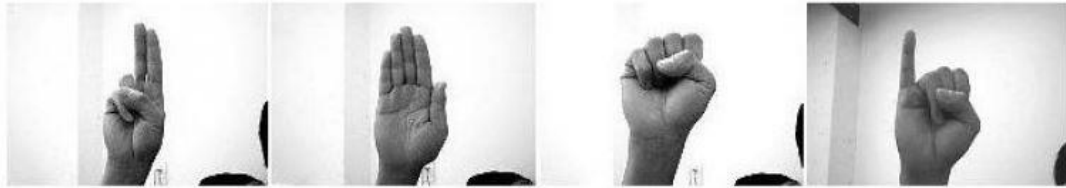


شکل ۲-۱۰ فرآیند تشخیص و ردیابی [۹].

در قسمت تشخیص چهره همان طور که گفته شد از الگوریتم ویولا-جونز استفاده می‌شود. در این الگوریتم به دلیل استفاده از انتگرال تصویر، تشخیص بلادرنگ خواهد داشت. الگوریتم جریان نوری برای ردیابی، از موقعیت یک شی در دو قاب متوالی استفاده می‌کند و یک روش حل معادلات آن، Lucas-Kanade است. در این سامانه در بهترین حالت به تعداد ۷ قاب بر ثانیه دست یافته‌اند [۹].

۲-۲-۲ شناسایی حالت‌های دست

در [۱۰]، یک رویکرد برای مسئله طبقه‌بندی بلادرنگ حالت دست پیشنهاد شده‌است. در این پژوهش به شناسایی ۴ حالت دست پرداخته‌اند. حالت دو انگشت، حالت کف دست، حالت مشت و حالت انگشت کوچک که این حالت‌ها را در شکل ۲-۱۱ مشاهده می‌کنید [۱۰].



شکل ۲-۱۱ حالت‌های مختلف دست در این پژوهش [۱۰].

الگوریتم مورد استفاده در شناسایی روش ویولا-جونز است. نمونه‌های مثبت از دست یک فرد تهیه شده‌است. ۴۵۰ نمونه برای هر حرکت دست با مقیاس‌های مختلف تهیه شده‌است. همچنین ۵۰۰ تصویر مختلف که شامل حرکت دست نیستند، جمع‌آوری شده‌است. نتیجه‌ی شناسایی طبقه‌بندها مطابق جدول ۲-۵ گزارش شده‌است [۱۰].

جدول ۲-۵ نتیجه شناسایی [۱۰].

Posture Name	Hits	Missed	False	Detection time (second)
Two-finger	100	0	29	3.049000
Palm	90	10	0	1.869000
Fist	100	0	1	2.829000
Little	93	7	2	2.452000

ستون آخر جدول ۲-۵ زمان لازم برای شناسایی حدود ۱۰۰ حرکت از هر حالت دست را نشان می‌دهد. در [۱۱] خدابخش و همکارانش، ۱۱ نوع، ژست دست را که شامل اعداد از ۱ تا ۵ و حالت‌های مشت، تایید، بالا، پایین، راست و چپ می‌شود را با استفاده از تکنیک‌های پردازش تصویر طبقه‌بندی کرده‌اند. در شکل ۲-۱۲ روند کلی این پژوهش نشان داده شده‌است. در اولین گام اقدام به تبدیل فضای رنگی

تصویر کرده‌اند. مدل رنگی انتخاب شده فضای YCbCr است که دلیل آن اهمیت رنگ‌ها در تصویر جهت تشخیص رنگ دست می‌باشد. در گام بعدی از طریق روش تفریق پس‌زمینه و تلفیق آن با روش ناحیه‌بندی رنگی، محدوده‌ای که تصویر دست در آن قرار دارد را استخراج می‌کنند. سپس به یافتن قسمت‌های جزئی‌تر دست می‌پردازند [۱۱].



شکل ۲-۱۲ مراحل عملیات پردازش تصویر [۱۱].

استفاده از روش تفریق پس‌زمینه، برای تعیین محل دست زمانی که چندین شی متحرک در تصویر وجود دارند، احتمال خطا را افزایش می‌دهد. به همین دلیل از ترکیب محدوده‌ی رنگ با تفریق پس‌زمینه استفاده شده‌است. همچنین برای کاهش احتمال خطا سه اقدام انجام شده‌است [۱۱]:

- نقاط و اشکال کوچک در تصویر که به عنوان نویز در نظر گرفته می‌شوند، حذف شده‌اند.
- نقاط داخلی پر می‌گردد.
- بهبود لبه تصویر با استفاده از روش گاوسی

برای یافتن نقطه مرکز دست، از روش تبدیل فاصله استفاده شده‌است. در این روش فاصله‌ی تمام نقاط ناحیه‌ی دست از نقاط مرزی بدست می‌آید و نقطه‌ای که بیشترین فاصله را داشته باشد مرکز دست خواهد بود. برای یافتن کف دست، از دایره‌ای با افزایش تدریجی شعاع، تا زمانی که به لبه‌ی دست برسند، استفاده شده‌است. به منظور تشخیص انگشتان، از زاویه‌ی قرار گیری آن‌ها با نقاط اساسی دست

استفاده شده است. پس از تشخیص انگشتان، با توجه به زوایای ایجاد شده‌ی میان انگشتان، حالت‌های مختلف دست را طبقه‌بندی کرده‌اند. نتایج حاصل از ارزیابی در جدول ۲-۶ نشان داده شده است [۱۱].

جدول ۲-۶ نتایج ارزیابی پژوهش [۱۱].

چپ/راست	بالا/پایین	نمایش اعداد توسط ژست دست ۱ لی ۵	تایید OK	مشت Fist	ژست دست
۲۵	۲۵	۲۵	۱۵	۱۵	تعداد کل تصاویر تست شده
۲۴	۲۳	۲۰	۱۲	۱۳	تعداد تشخیص صحیح
۹۵ %	۹۲ %	۸۰ %	۸۰ %	۸۶ %	درصد صحت عملکرد

زمان پردازش هر یک تصویر ۰/۵ ثانیه گزارش شده است [۱۱].

۳-۲-۲ شناسایی اشیا

در ادامه به بررسی چندین مقاله خواهیم پرداخت که با استفاده از شبکه‌های عمیق به شناسایی یک شی پرداخته‌اند. تمرکز ما در بررسی این مقالات، بر روی الگوریتم‌های مورد استفاده و نتایج آن‌هاست و اینکه به شناسایی چه شیئی پرداخته‌اند برای ما اهمیت چندانی ندارد.

در [۱۵]، مائو و همکارانش با استفاده از شبکه‌های عمیق، به شناسایی اشیا پرداخته‌اند به گونه‌ای که این شبکه برای برنامه‌های نهفته^۱ قابلیت اجرای بلادرنگ داشته باشد. با ظهور شبکه‌های عصبی بعضی از روش‌های شناسایی اشیا قدیمی به تدریج کنار گذاشته شدند. به دلیل دقت پایین و جوابگو نبودن در محیط‌های مختلف. به دلیل رشد توان محاسباتی تراشه‌های یادگیری عمیق و وجود مجموعه‌های بزرگ برچسب‌گذاری شده مثل مجموعه‌های ImageNet، PASCAL VOC و MS-COCO، شبکه‌های پیشگی عمیق به طور گسترده مورد مطالعه قرار گرفته‌اند [۱۵].

برخی از روش‌های شناسایی اشیا مبتنی بر شبکه‌های پیشنهاد ناحیه^۲، مثل R-CNN، Fast R-CNN، Mask R-CNN و Faster R-CNN شناسایی اشیا را با دقت بالا انجام دادند. اما این روش‌ها برای

^۱Embedded Applications

^۲Region proposal networks (RPN)

دستیابی به دقت بالا، به قدرت محاسباتی بالای پردازنده‌های گرافیکی وابسته بودند. اما بهبود میزان دقت با محاسبات سنگین، برای کاربردهای بلادرنگ در دنیای واقعی، کمکی نخواهد کرد. به این دلیل که در بسیاری از کاربردها، توانایی مدل‌های یادگیری عمیق در اجرای بلادرنگ، یک عامل ضروری است. اخیراً شناساگرهای اشیا سریعی پیشنهاد شده‌است مثل RetinaNet، SSD، نسخه‌های مختلف^۱ YOLO. در بین این شناساگرهای اشیا سریع، شاید YOLO نسخه سوم، مشهورترین شناساگر شی در برنامه‌های کاربردی باشد که میزان دقت و سرعت در آن به خوبی تنظیم شده‌است. اما این نسخه از YOLO، همچنان از لحاظ محاسبات بسیار سنگین است که باعث می‌شود در دستگاه‌های نهفته، محاسبات بیش از حد باشد و توان مصرفی افزایش یابد. تنها راه اجرای شناساگر شی بر روی دستگاه‌های نهفته، کاهش اندازه‌ی مدل یادگیری عمیق است. در نسخه‌های مختلف YOLO، SSD و دیگر شبکه‌ها، نسخه‌های سبک‌تری نیز وجود دارند که YOLO-tiny، YOLO-Lite و tiny SSD نامیده می‌شوند. اما دقت این شبکه‌ها شدیداً کاهش می‌یابد. بنابراین، اینکه چطور اندازه‌ی مدل و عملیات ممیز شناور بدون کاهش میزان دقت، کاهش پیدا کند؛ یک مسئله‌ی مهم است که به هنگام پیاده‌سازی بر روی دستگاه‌های نهفته اهمیت می‌یابد [۱۵].

مشهورترین روش، کاهش مولفه‌ها و اندازه‌ی شبکه با استفاده از طراحی یک شبکه کارآمدتر است. به عنوان مثال شبکه‌های SqueezeNet، MobileNet، ShffuleNet با کاهش اندازه‌ی مدل و عملیات ممیز شناور به دقت خوبی دست یافته‌اند [۱۵].

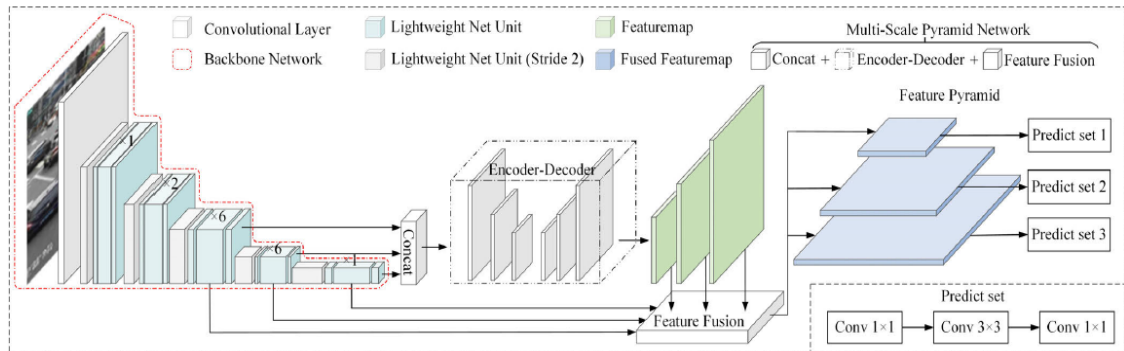
در این مقاله، بدون کاهش چشمگیر دقت، به طراحی مجدد یک شبکه‌ی سبک بر اساس YOLO نسخه سوم پرداخته‌اند و آن را Mini-YOLOv3 نامیده‌اند. با توجه با اینکه کاهش لایه‌های اصلی شبکه، در میزان دقت تاثیر منفی دارد، به عنوان مکمل، یک شبکه‌ی ویژگی هر می چند مقیاس^۳ را برای افزایش

^۱You only look once

^۲Floating point operations (FLOPs)

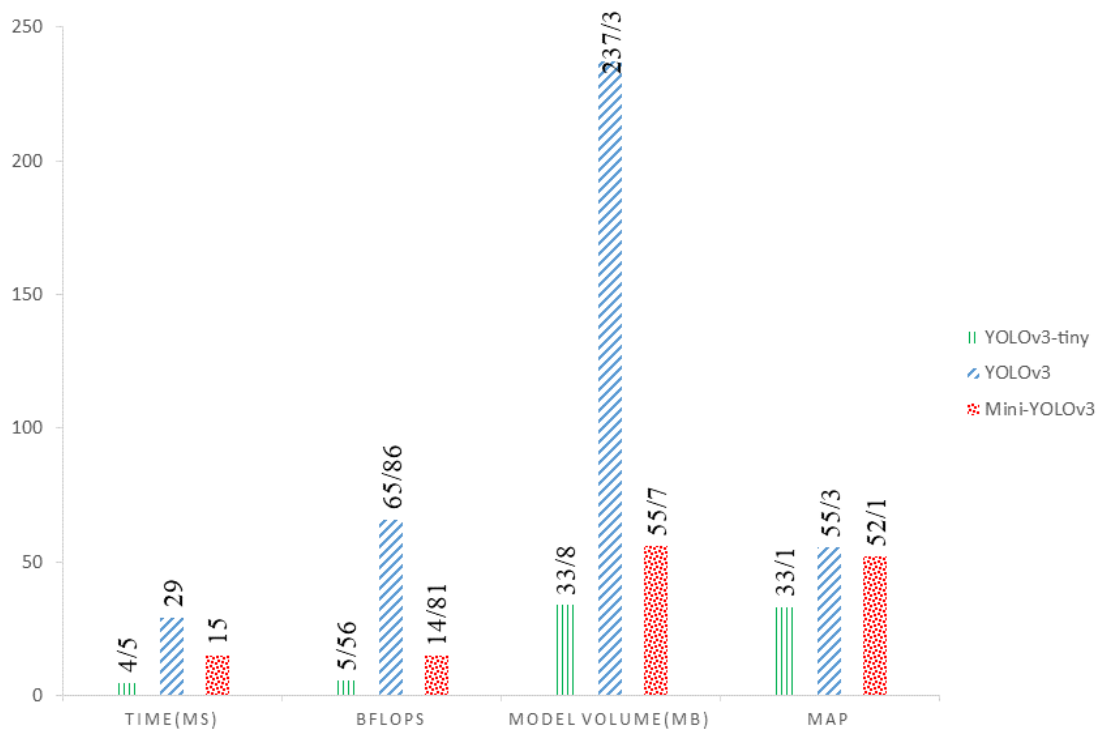
^۳Multi-Scale Feature Pyramid Network (MSFPN)

کارایی استخراج ویژگی، پیشنهاد داده‌اند. در شکل ۲-۱۳ جزئیات شبکه‌ی Mini-YOLOv3 را مشاهده می‌کنید [۱۵].



شکل ۲-۱۳ یک نمایی از Mini-YOLOv3. این شبکه از ساختار YOLOv3 و شبکه‌ی ویژگی هر می چند مقیاس برای استخراج ویژگی از تصویر ورودی استفاده می‌کند [۱۵].

در الگوریتم پیشنهادی این مقاله، برای استخراج ویژگی از تصویر ورودی، از مبنای اصلی شبکه YOLOv3 و شبکه‌ی ویژگی هر می چند مقیاس، استفاده می‌کنند. سپس براساس ویژگی‌های آموخته شده، کادرهای مرزی پیش‌بینی شده را تولید می‌کند و براساس نگاه داشتن پیش‌بینی بیشینه، نتیجه را تولید می‌کنند. در MSFPN، سه ماژول وجود دارد. ماژول Concat وظیفه‌ی اتصال هر ویژگی شبکه‌ی backbone را برعهده دارد. ماژول رمزگذار-رمزگشا وظیفه‌ی تولید ویژگی‌های چند مقیاسی را برعهده دارد و ماژول تجمع ویژگی وظیفه‌ی تجمع ویژگی‌ها را برعهده دارد. در پایان نتایج این مقاله در شکل ۱۴-۲ ذکر شده‌است [۱۵].



شکل ۲-۱۴ مقایسه شبکه‌های YOLOv3، YOLOv3-tiny و Mini-YOLOv3 [۱۵].

همان‌طور که در شکل بالا مشاهده می‌کنید، شبکه‌ی پیشنهادی Mini-YOLOv3 نصف شبکه‌ی YOLOv3 زمان برای شناسایی نیاز دارد و میانگین دقتی تقریباً مشابه YOLOv3 دارد [۱۵].

ماتزیا و همکارانش [۱۶]، نظارت به مزارع کشاورزی را با استفاده از یادگیری عمیق انجام داده‌اند. دو عامل مهم که کشاورزان را قادر به تخمین بار محصول و نقشه‌برداری عملکرد در درختان میوه، می‌کند، شمارش میوه و تخمین اندازه است. در این مقاله تمرکز روی شناسایی میوه‌ی سیب است. این الگوریتم‌ها باعث می‌شوند که ربات‌های برداشت محصول، سریع و دقیق، میوه‌ها را به صورت بلادرنگ شناسایی کنند [۱۶].

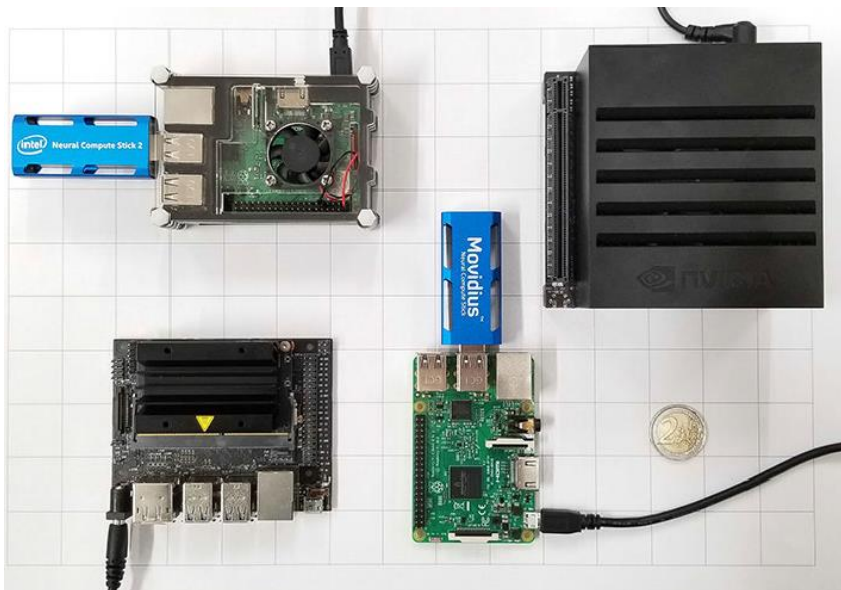
شبکه Faster R-CNN با استفاده از روش شبکه پیشنهاد ناحیه در تصاویر، نواحی مورد علاقه را شناسایی می‌کند. این شبکه به همراه شبکه VGG 16 یک روش پیشرفته برای شناسایی میوه هستند اما از لحاظ سرعت شناسایی ضعیف است. شبکه‌ی YOLO به مسئله‌ی طبقه‌بندی و تعیین موقعیت به

عنوان یک مسئله‌ی تخمین تابع می‌نگرد. شبکه‌ی YOLO، مستقیماً تخمین تابع را بر روی تصویر برای شناسایی اهداف اعمال می‌کند بدون استفاده از روش شبکه پیشنهاد ناحیه. از این رو این شبکه سریع است و می‌تواند در کاربردهای بلادرنگ استفاده شود. با این حال، همچنان نسخه‌ی پیشرفته‌ی YOLO یعنی YOLO نسخه سوم، برای کاربردهای بلادرنگ مورد استفاده در دستگاه‌های نهفته ربات‌ها مناسب نیست [۱۶].

بهینه‌سازی پارامترهای این مدل پیچیدگی محاسباتی آنرا کاهش می‌دهد که برای استقرار بر روی دستگاه‌های نهفته‌ای مانند رسپیری پای و انویدیا جتسون لازم است [۱۶].

به منظور آموزش مدل، از کارت گرافیک قدرتمند Nvidia RTX 2080Ti به همراه کودا نسخه ۱۰ و ۶۴ گیگابایت حافظه DDR4 استفاده شده‌است. این پردازنده گرافیکی قدرتمند باعث می‌شود که به فرآیند آموزش سرعت بخشد. تعداد عملیات ممیز شناور این کارت گرافیک ۲۶/۹ ترافلاپس است [۱۶].

در این مقاله به منظور پیاده‌سازی نهفته از دستگاه‌های مختلفی استفاده شده‌است. که در شکل ۲-۱۵ این دستگاه‌ها را مشاهده می‌کنید. یک رسپیری پای مدل B+ ۳، به همراه دو نسل شتاب دهنده‌های محاسباتی یادگیری عمیق Intel Movidius Neural Compute Stick، یک بُرد انویدیا جتسون نانو و یک انویدیا جتسون AGX Xavier. در جدول ۲-۷ مقایسه مشخصات این دستگاه‌ها را مشاهده می‌کنید.



شکل ۲-۱۵ دستگاه‌های نهفته مورد استفاده در این مقاله [۱۶].

جدول ۲-۷ مقایسه مشخصات دستگاه‌های مورد استفاده در این مقاله [۱۶].

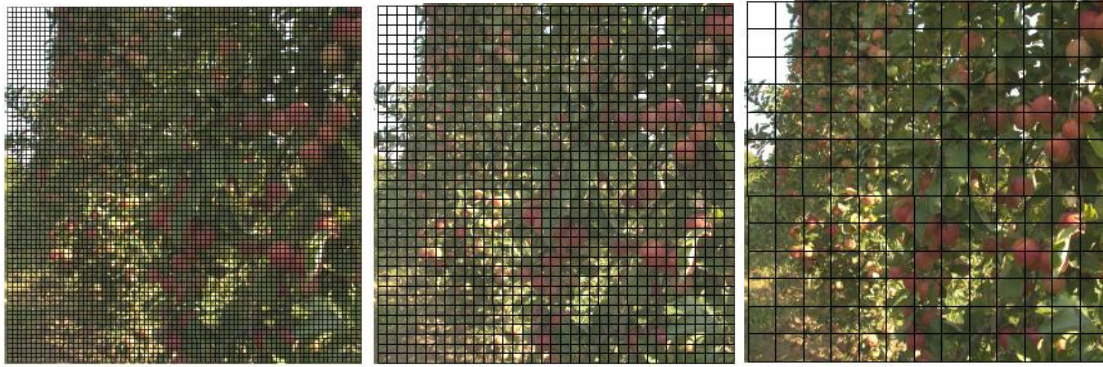
	Intel Movidius NCS	Intel NCS 2	Jetson Nano	Jetson AGX Xavier
Features size	73 x 26 mm	73 x 26 mm	70 x 45 mm	100 x 87 mm
HW Accelerator	Myriad 2 VPU	Myriad X VPU	128-core NVIDIA Maxwell GPU	512-core NVIDIA Volta GPU with 64 Tensor Cores and 2x NVDLA Engines
CPU	N.A.	N.A.	Quad-core Arm A57 @ 1.43 GHz	Octa-core NVIDIA Carmel Arm
Memory	4GB LPDDR3	4GB LPDDR3	4GB LPDDR4	16GB LPDDR4x
Storage	N.A.	N.A.	Micro SD card slot or 16 GB eMMC flash	32 GB eMMC 5.1
Peak performance	100 GFLOPs	150 GFLOPs	472 GFLOPs	16 TFLOPs
Native precision support	FP16	FP16	FP16 / FP32	FP16 / FP32
Nominal power	1 W	1.5 W	5 / 10 W	10 / 15 / 30 W
Weight	18 g	19 g	140 g	280 g
Price	\$ 70	\$ 74	\$ 99	\$ 800

در بین نسخه‌های مختلف YOLO، نسخه‌هایی که با عنوان tiny معرفی می‌شوند برای سامانه‌های نهفته مناسب هستند. در این مقاله از نسخه YOLOv3-tiny استفاده می‌شود. این نسخه یک شبکه تمام پیچشی است که می‌تواند در زمان آموزش و بعد از آن ورودی‌هایی با اندازه‌های مختلف را بپذیرد. این شبکه می‌تواند به دو قسمت اصلی تقسیم شود [۱۶]:

- مورد اول استخراج کننده ویژگی ملقب به Darknet-19 است. این استخراج کننده ویژگی بسیار سبک و کارآمد است اما می‌تواند با سایر مبناهای مثل ResNet و DenseNet جایگزین شود. این معماری از شبکه VGGNet الهام گرفته شده است.

- مورد دوم معماری YOLOv3-tiny، تجزیه و تحلیل خروجی‌های شبکه‌ی مبنا و پیش‌بینی موقعیت و کلاس اشیا مختلفی که در تصاویر ورودی قرار دارند، است.

دو نسخه ابتدایی YOLO در شناسایی اشیا کوچک در تصویر ورودی خیلی خوب عمل می‌کردند. این اساساً به این خاطر است که YOLO یک محدودیت مکانی قوی را در پیش‌بینی کادرهای مرزی اعمال می‌کند. هر شبکه‌ی سلولی فقط می‌تواند سه کادر مرزی پیش‌بینی نماید. به علاوه، کاهش ابعاد ورودی اعمال شده در شبکه‌ی مبنا، باعث حذف ویژگی‌های سطح پایین می‌شود که برای شناسایی اشیا کوچک ضروری است. به همین دلیل نویسندگان، سومین نسخه tiny را پیشنهاد دادند که در دو مقیاس مختلف پیش‌بینی را انجام می‌دهد. در این مقاله این روند را جلوتر برده‌اند و معماری YOLOv3-tiny را به گونه‌ای تغییر داده‌اند که پیش‌بینی را در سه مقیاس مختلف انجام می‌دهد. همچنین اندازه‌ی ابعاد ورودی از $416 * 416$ به $608 * 608$ افزایش یافته است. لایه‌های شناسایی عملیات شناسایی را در سه اندازه‌ی مختلف که از گام‌هایی با مقادیر ۳۲، ۱۶ و ۸ حاصل می‌شوند، انجام می‌دهند. بنابراین در این مدل جدید، عملیات شناسایی بر روی تصاویری که دارای $19 * 19$ ، $38 * 38$ و $76 * 76$ سلول هستند، انجام می‌شود. در شکل ۲-۱۶ این موارد را مشاهده می‌کنید [۱۶].



۷۶*۷۶

۳۸*۳۸

۱۹*۱۹

شکل ۱۶-۲ شبکه‌های سلولی پیشنهادی این مقاله [۱۶].

نتایج ارزیابی سرعت این مقاله را در جدول ۸-۲ مشاهده می‌کنید [۱۶].

جدول ۸-۲ نتایج اجرای الگوریتم پیشنهادی بر روی دستگاه‌های نهفته‌ی مختلف [۱۶].

Device	Mode	$V_{al}[V]$	$I_{mean}[A]$	$P[W]$	fps
Jetson AGX Xavier	IDLE 30W	19	0.35	6.65	N.A.
	RUNNING 30W	19	1	19	30
	RUNNING 15W	19	0.88	16.72	25
	RUNNING 10W	19	0.66	12.54	13
Jetson Nano	IDLE 10W	5	0.32	1.6	N.A.
	RUNNING 10W	5	2.04	10.20	8
	RUNNING 5W	5	1.42	7.1	6
Raspberry Pi 3B+	IDLE	5	0.61	3.075	N.A.
RP3 + Neural Stick 1	RUNNING	5	1.2	6	4
RP3 + Neural Stick 2	RUNNING	5	1.12	5.6	5

نتایج سنجش دقت در این مقاله مطابق با جدول ۹-۲ است [۱۶].

جدول ۹-۲ نتایج ارزیابی سرعت مقاله [۱۶].

IOU_{target}	Proposed architecture			Original Yolov3-Tiny	Gain
	Recall ($c = 0.25$)	Precision ($c = 0.25$)	mAP	mAP	
0.50	0.83	0.69	83.64%	77.02 %	6.62 %
0.75	0.55	0.46	47.97%	42.50 %	5.47 %

در [۱۷]، به دنبال راهی برای بهبود سرعت و دقت YOLOv3-tiny هستند. میزان دقت YOLOv3-tiny خیلی زیاد نیست و عملکرد بلادرنگ آن نیز برای دستگاه‌های با توان پایین مناسب نیست. به منظور دستیابی به یک مدل شناساگر شی کارآمد برای محیط‌هایی که دارای محدودیت هستند، با نشات گرفتن از YOLOv3-tiny، مدل Tinier-YOLO در این مقاله پیشنهاد شده است. مدلی که در آن اندازه‌ی مدل کاهش می‌یابد در حالی که نرخ دقت افزایش یافته‌است و عملکرد بلادرنگ خواهد داشت. در ابتدا، ساختار شبکه YOLOv3-tiny توسط کاهش تعداد مولفه‌ها بهینه می‌شود. در این بین از ماژول آتش^۱ شبکه SqueezeNet استفاده می‌شود تا شبکه جدید کوچکتر و در عین حال سریع‌تر شود. سپس به منظور دستیابی به دقت بیشتر در شبکه‌ی Tinier-YOLO، از لایه‌های متراکم^۲ بین ماژول آتش و لایه‌ی گذرگاه^۳ استفاده می‌شود. ساختار شبکه پیشنهادی را در شکل ۲-۱۷ مشاهده می‌کنید. بعد از پیاده‌سازی این شبکه بر روی بُرد جتسون TX1 با پشتیبانی از کودا به نتایج زیر دست یافته‌اند [۱۷].

جدول ۲-۱۰ مقایسه میزان سرعت و دقت دو شبکه Tiny-YOLO-v3 و Tinier-YOLO [۱۷].

	Tiny-YOLO-V3		Tinier-YOLO	
Fire Module		✓	✓	✓
Dense Connection			✓	✓
Passthrough Layer				✓
Model Size(MB)	34.9	6.9	8.3	8.9
BFLOP/s	5.474	2.350	2.552	2.563
mAP(%)	61.3	60.6	64.8	65.7
FPS	21.6	25.5	24.9	25.1

^۱Fire

^۲Dense

^۳Passthrough

با این حال در بهترین حالت، زمان لازم برای عبور یک تصویر از شبکه‌ی پیشنهادی ۵/۷ ثانیه است. که این زمان پردازش زیاد باعث می‌شود که این شبکه‌ی پیشنهادی برای این پایان‌نامه قابل اجرا نباشد.

مقاله [۹] و [۱۰]، فرآیند شناسایی را با استفاده از الگوریتم ویولا-جونز به ترتیب بر روی گوشی موبایل و رایانه انجام داده‌اند که به نتایج قابل قبولی دست یافته‌اند. سرعت پردازش این دو مقاله باعث شد که روش ویولا-جونز به عنوان یک روش قابل اجرا برای این پایان‌نامه نیز در نظر گرفته شود.

در [۱۱] عملیات شناسایی حالت‌های دست را با استفاده از اندازه‌گیری زوایای بین انگشتان و محل قرارگیری آن‌ها، انجام داده‌اند که نتایج ذکر شده از لحاظ سرعت انجام، روش پیشنهادی را برای این پایان‌نامه مناسب نمی‌کند.

مقاله‌های [۱۵]، [۱۶] و [۱۷]، به دنبال بهبود سرعت و دقت شبکه‌ی معروف YOLOv3-tiny هستند. و طبق نتایج ذکر شده پیاده‌سازی روش‌های پیشنهادی این مقاله‌ها بر روی بُردهای نهفته‌ی ساخت شرکت انویدیا که از کودا پشتیبانی می‌کنند، معقول است. اما با بررسی برخی از کارهای انجام‌شده با شبکه یاد شده و پیاده‌سازی YOLOv3-tiny بر روی گوشی موبایل، همچنین انجام آزمایش این شبکه بر روی بُرد نهفته فاقد پردازنده گرافیکی انویدیا و کسب نرخ قاب بر ثانیه بسیار پایین، باعث شد تا از پیاده‌سازی این شبکه بر روی گوشی موبایل، صرف نظر شود.

شبکه‌ی معروف SSD Mobilenet نیز بر روی بُرد نهفته فاقد پردازنده گرافیکی انویدیا، آزمایش شد و در ترکیب با روش تفریق پس‌زمینه، نرخ قاب بر ثانیه‌ی پایینی حاصل شد که طبیعتاً پس از اجرای قسمت گرافیکی این نرخ پایین‌تر نیز خواهد آمد.

از بررسی‌های صورت گرفته، روش ویولا-جونز را با ترکیب ایده‌های مختلف، برای انجام قسمت‌های پردازشی استفاده کرده‌ایم که در فصل‌های بعدی به توضیح این مراحل خواهیم پرداخت.

فصل سوّم

مباحث نظری

در این پژوهش به منظور شناسایی و ردیابی حرکات دست و چهره کاربر از طبقه‌بندهای آبخاری استفاده شده‌است. این روش که به الگوریتم ویولا-جونز معروف است در سال ۲۰۰۱ مطرح شده‌است و این مطالب از مرجع [۱۲] است.

۳-۱ الگوریتم ویولا-جونز

در این بخش یک رویکرد یادگیری ماشین برای شناسایی شی را تشریح می‌کنیم که قادر به پردازش تصاویر با سرعت بسیار بالا بوده و نرخ شناسایی خوبی دارد. از این سامانه در موارد شناسایی مختلف می‌توان استفاده کرد که در این بخش شناسایی چهره از روبه‌رو را تشریح خواهیم کرد.

عملکرد شناسایی چهره این روش، روی تصاویر با اندازه ۳۸۴ در ۲۸۸ پیکسل، نرخ ۱۵ قاب بر ثانیه بر روی یک رایانه با پردازنده اینتل پنتیوم III با فرکانس پردازنده ۷۰۰ مگاهرتز، بوده است. در دیگر سامانه‌های شناسایی چهره، با کمک گرفتن از تفاضل تصاویر در دنباله‌های ویدئویی یا رنگ پیکسل‌ها در تصاویر رنگی، به نرخ قاب بالا دست یافته‌اند. این سیستم فقط با اطلاعات موجود در یک تک تصویر خاکستری به نرخ قاب بالا دست یافته‌است. سه مؤلفه در این سامانه شناسایی شی دخیل است. ابتدا به اختصار هریک از این مؤلفه‌ها را معرفی خواهیم کرد سپس با جزئیات بیشتری هر قسمت را توضیح خواهیم داد.

مؤلفه‌ی اول انتگرال تصویر است که امکان ارزیابی بسیار سریع ویژگی را فراهم می‌کند. این سامانه شناسایی، مستقیماً با شدت روشنایی تصویر سروکار ندارد بلکه مجموعه‌ای از ویژگی‌ها که براساس توابع هار هستند را استفاده می‌کند. به منظور محاسبه سریع این ویژگی‌ها در مقیاس‌های مختلف از انتگرال تصویر که نماینده‌ی هر تصویر است استفاده می‌شود. انتگرال تصویر می‌تواند از انجام چندین عملیات بر روی هر پیکسل یک تصویر محاسبه شود. پس از اولین محاسبه، هر کدام از این ویژگی‌های شبه هار، می‌تواند در هر مقیاس یا موقعیتی در زمان ثابت محاسبه شود.

مؤلفه‌ی دوم روشی است که با استفاده از الگوریتم آدابوست تعداد کمی از ویژگی‌های مهم را به منظور ساخت یک طبقه‌بند، انتخاب می‌کند. در هر پنجره زیرمجموعه یک تصویر تعداد کل ویژگی‌های شبه‌ها را بسیار بیشتر از تعداد پیکسل‌ها است. به منظور اطمینان از طبقه‌بندی سریع، فرایند آموزش باید تعداد زیادی از ویژگی‌های موجود را حذف کند و بر روی مجموعه‌ی کوچک ویژگی‌های مهم متمرکز باشد. این مورد در فرآیند تقویت اتفاق می‌افتد [۱۲].

تقویت کردن یک روش گروهی است که از تعدادی طبقه‌بند ضعیف، یک طبقه‌بند قوی می‌سازد. این کار توسط ساخت یک مدل از مجموعه آموزش انجام می‌شود، سپس ساخت مدل دوم که تلاش آن تصحیح خطاهای مدل اول است. مدل‌ها تا زمانی اضافه می‌شوند که مجموعه آموزش به خوبی پیش‌بینی شود یا اینکه حداکثر تعداد مدل‌ها اضافه شوند. آدابوست اولین الگوریتم تقویت کردنی بود که با موفقیت کامل برای طبقه‌بندی باینری توسعه داده شد [۱۳].

مؤلفه سوم روشی است برای ترکیب پی‌درپی پیشرفته‌ی طبقه‌بندها در یک ساختار آبشاری که سرعت شناساگر را توسط تمرکز بر روی نواحی محتمل، افزایش می‌دهد.

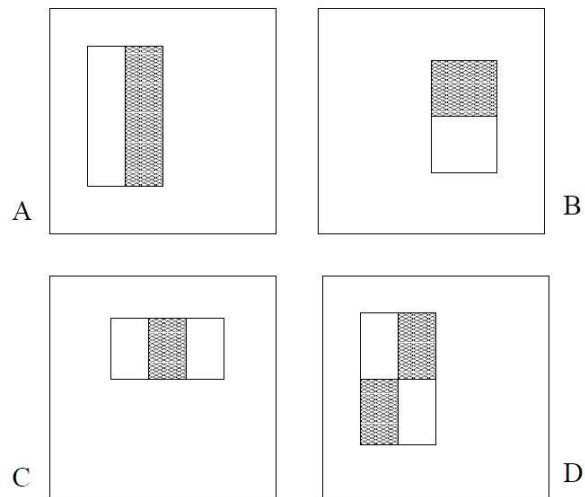
۳-۱-۱-۱ انتگرال تصویر

این شناساگر تصاویر را براساس مقادیر ویژگی‌های ساده طبقه‌بندی می‌کند. دلایل زیادی وجود دارد که به جای استفاده‌ی مستقیم از پیکسل‌ها، از ویژگی‌ها استفاده شود. یکی از دلایل سرعت بیشتر سامانه‌ی مبتنی بر ویژگی نسبت به سامانه مبتنی بر پیکسل است. در این سامانه از سه نوع ویژگی استفاده می‌شود:

۱- مقدار ویژگی دو مستطیل؛ اختلاف بین مجموع پیکسل‌های ناحیه بین دو مستطیل است. مناطق اندازه و شکل یکسانی دارند و از نظر افقی یا عمودی بودن مجاور یکدیگرند. همان‌طور که در شکل ۳-۱ مشاهده می‌کنید.

۲- ویژگی سه مستطیل^۲ مجموع درون دو مستطیل خارج را محاسبه و از مجموع موجود در مستطیل مرکز کم می‌کند.

۳- ویژگی چهار مستطیل^۳ اختلاف بین جفت‌های مورب مستطیل‌ها را محاسبه می‌کند.



شکل ۳-۱ مثالی از ویژگی‌های مستطیل. مجموع پیکسل‌هایی که در مستطیل‌های سفید قرار دارند از مجموع پیکسل‌های مستطیل‌های خاکستری کم می‌شوند. ویژگی‌های دو مستطیل در شکل (A) و (B) نشان داده شده‌است. شکل (C) ویژگی‌های سه مستطیل و شکل (D) ویژگی‌های چهار مستطیل را نشان می‌دهند.

^۱Two Rectangle Feature

^۲Three Rectangle Feature

^۳Four Rectangle Feature

تفکیک پذیری شناساگر ۲۴ در ۲۴ پیکسل است. مجموعه ویژگی‌های مستطیل بسیار زیاد است بیش از ۱۸۰۰۰۰. ویژگی‌های مستطیل می‌تواند با سرعت بالا توسط اصطلاح انتگرال تصویر محاسبه شود.

انتگرال تصویر در موقعیت y و x طبق رابطه زیر بدست می‌آید:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1-3)$$

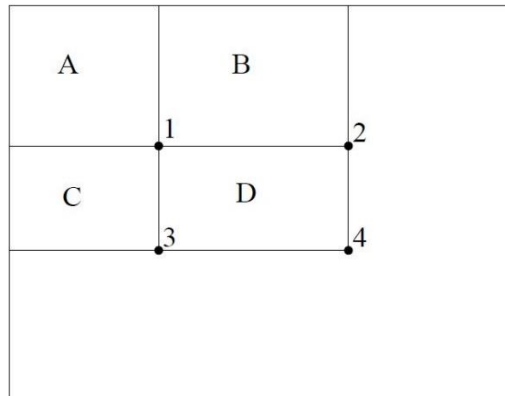
که $ii(x, y)$ انتگرال تصویر است و $i(x, y)$ تصویر اصلی است. تعریف دیگری از آن به شکل زیر است:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2-3)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (3-3)$$

که $s(x, y)$ برابر با مجموع سطری و $s(x, -1) = 0$ و $ii(-1, y) = 0$ است. بنابراین انتگرال تصویر را می‌توان با یکبار پویش تصویر اصلی محاسبه کرد.

با استفاده از انتگرال تصویر مجموع هر مستطیلی با استفاده از چهار آرایه مرجع، محاسبه می‌شود همان‌طور که در شکل ۲-۳ مشاهده می‌کنید.



شکل ۲-۳ مجموع پیکسل‌های درون مستطیل D با چهار آرایه مرجع می‌تواند محاسبه شود. مقدار انتگرال تصویر در موقعیت ۱ برابر مجموع پیکسل‌های درون مستطیل A است. این مقدار در موقعیت ۲ برابر با $A+B$ و در موقعیت ۳ برابر $A+C$ و در موقعیت ۴ برابر $A+B+C+D$ است. مجموع درون مستطیل D می‌تواند توسط $(2+3) - (4+1)$ محاسبه شود.

واضح است که اختلاف بین دو جمع مستطیلی می‌تواند با هشت مرجع محاسبه شود. از آنجایی که ویژگی‌های دو مستطیل شامل مجموع مستطیلی هم‌جوار هستند، می‌توانند توسط شش آرایه مرجع محاسبه شوند. در مورد ویژگی‌های سه مستطیل این مقدار به هشت و در ویژگی‌های چهار مستطیل این مقدار به نه می‌رسد.

۳-۱-۲ آموزش طبقه‌بند

با توجه به مجموعه ویژگی و تصاویر مثبت و منفی مجموعه آموزش، به منظور آموزش یک طبقه‌بند از هر رویکرد یادگیری ماشین می‌توان استفاده کرد. در این سامانه از یک نوع آدابوست برای انتخاب یک مجموعه کوچکی از ویژگی و همچنین آموزش طبقه‌بند استفاده می‌شود. در حالت معمولی، الگوریتم یادگیری آدابوست به منظور تقویت عملکرد، الگوریتم یادگیری ساده (ضعیف) را استفاده می‌کند.

همان‌طور که گفته شد بیش از ۱۸۰۰۰۰ ویژگی مستطیل وجود دارد که به هر زیرپنجره تصویر مرتبط هستند که این تعداد بسیار بیشتر از تعداد پیکسل‌ها است. با وجود اینکه هر ویژگی می‌تواند بسیار کارآمد، محاسبه شود، محاسبه تمام این مجموعه ضروری نیست. فرضیه‌ی این سامانه که با استفاده از آزمایش به آن رسیده‌اند، این است که تعداد بسیار کمی از این ویژگی‌ها می‌توانند با یکدیگر ترکیب شوند به منظور ایجاد یک طبقه‌بند مؤثر. چالش اصلی یافتن این ویژگی‌ها است.

به منظور رسیدن به این هدف، الگوریتم یادگیری ضعیف^۱ به‌گونه‌ای طراحی شده است تا یک ویژگی مستطیلی را انتخاب کند که به بهترین شکل نمونه‌های مثبت و منفی را از یکدیگر جدا کند. برای هر ویژگی، طبقه‌بند ضعیف بهینه‌ترین آستانه‌ی طبقه‌بندی را مشخص می‌کند مثل تعداد نمونه‌هایی که نادرست طبقه‌بندی شده‌اند، کمترین باشد. یک طبقه‌بند ضعیف $h_j(x)$ شامل یک ویژگی f_j ، یک آستانه‌ی θ_j و یک متغیر p_j که جهت نامساوی را تعیین می‌کند، است:

^۱Weak Learning

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (4-3)$$

که در این رابطه x یک زیرپنجره 24×24 پیکسل از یک تصویر است. فرآیند تقویت کردن در شکل ۳-۳ توضیح داده شده است.

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:

1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that w_t is a probability distribution.

2. For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Choose the classifier, h_t , with the lowest error ϵ_t .
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

شکل ۳-۳ الگوریتم آداپوست برای یادگیری طبقه‌بند.

در این شکل برای n نمونه، n برچسب داده شده است. بعد از مقدار دهی اولیه وزن‌ها، در هر تکرار وزن‌ها نرمالیزه می‌شوند. و مقدار خطا برای هر طبقه‌بند محاسبه می‌شود و طبقه‌بند با کمترین خطا انتخاب می‌شود. به روز رسانی وزن‌ها در جهتی است که وزن‌هایی که کمترین خطا را ایجاد کرده‌اند تقویت شوند. سرانجام طبقه‌بند نهایی طبق رابطه‌ی گفته شده در شکل ۳-۳ بدست می‌آید. در هر بار تقویت، یک ویژگی از ۱۸۰،۰۰۰ ویژگی نامزد، انتخاب می‌شود.

در عمل فقط با یک ویژگی نمی‌توان طبقه‌بندی را با خطای کم انجام داد. ویژگی‌هایی که در مراحل اولیه فرآیند تقویت انتخاب می‌شوند، نرخ خطایی بین ۰/۱ و ۰/۳ دارند در حالی که در مراحل پایانی خطای بین ۰/۴ و ۰/۵ بدست می‌آید.

آزمایش‌های اولیه نشان می‌دهد که یک طبقه‌بند چهره که از ۲۰۰ ویژگی ساخته شده‌است، به میزان تشخیص ۹۵ درصد با نرخ اشتباه مثبت ۱ در ۱۴۰۸۴ دست می‌یابد. این نتیجه متقاعد کننده‌است، اما در بسیاری از پروژه‌های واقعی این مقدار کافی نیست. اضافه کردن ویژگی، ساده‌ترین تکنیک برای بهبود عملکرد شناسایی است که مستقیماً زمان محاسبات را افزایش می‌دهد.

در مثال شناسایی چهره، یک طبقه‌بند آبخاری با ۳۸ لایه آموزش دیده است. مجموعه آموزش مورد استفاده شامل ۴۹۱۶ تصویر چهره و ۹۵۴۴ تصویر غیره چهره می‌شود که تمامی تصاویر به 24×24 تغییر اندازه یافته‌اند.

تعداد ویژگی‌های ۵ لایه اول شناساگر به ترتیب برابر ۱، ۱۰، ۲۵، ۲۵ و ۵۰ است و تعداد ویژگی‌ها در لایه‌های باقی مانده مرتباً افزایش می‌یابد. تعداد کل ویژگی‌ها در تمام لایه‌ها ۶۰۶۱ است.

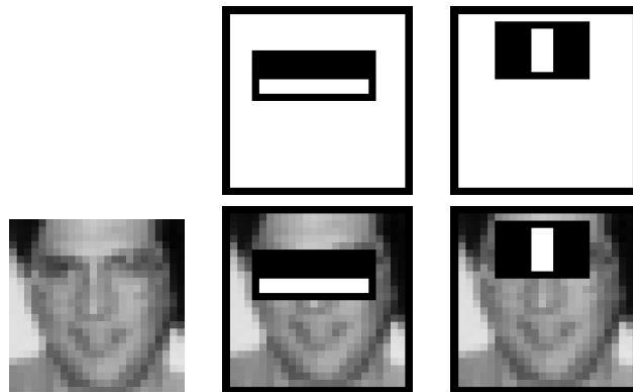
شناساگر نهایی در مقیاس‌ها و مکان‌های مختلفی در سراسر تصویر اعمال می‌شود. مقیاس‌های مختلف به جای اعمال بر روی تصاویر بر روی شناساگر اعمال می‌شود. این فرآیند منطقی‌تر است. زیرا که ویژگی‌ها را می‌توان در هر مقیاسی با صرف یک زمان ثابت محاسبه کرد. نتایج خوبی با استفاده از مجموعه‌ی مقیاس‌ها با ضریب ۱،۲۵ به دست آمده است. برای اعمال شناساگر به مکان‌های مختلف، از

شیفت دادن پنجره به اندازه Δ استفاده می‌شود. فرآیند شیفت دادن از مقیاس شناساگر تاثیر می‌پذیرد. اگر مقیاس فعلی s باشد، پنجره با $[s\Delta]$ شیفت داده می‌شود که کروسه عملگر گرد کردن است. انتخاب Δ هم بر سرعت و هم بر دقت شناساگر تاثیر می‌گذارد. نتایج مثال تشخیص چهره با $\Delta = 1$ بدست آمده‌اند. با افزایش آن به مقدار $1/5$ می‌توان سرعت را با کاهش مقدار کمی از دقت، افزایش داد.

برای مسئله‌ی شناسایی چهره، ویژگی‌های مستطیل اولیه انتخابی توسط آدابوست معنادار است و به راحتی قابل تفسیر است. به نظر می‌رسد ویژگی انتخابی اول بر روی این ویژگی متمرکز است که نواحی چشم‌ها اغلب از نواحی بینی و گونه‌ها تاریک‌تر است. ویژگی انتخابی دوم بر این ویژگی متمرکز است که چشم‌ها تاریک‌تر از استخوان بینی هستند. این مسئله در شکل ۳-۴ نشان داده شده‌است.

۳-۱-۳ طبقه‌بند آبشاری

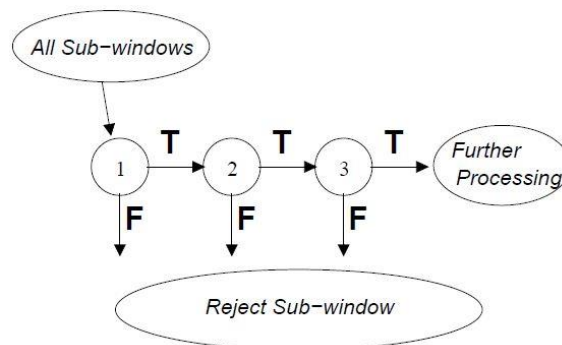
این قسمت یک الگوریتمی برای ساختن آبشاری از طبقه‌بندها را تشریح می‌کند که عملکرد شناسایی را افزایش می‌دهد. طبقه‌بندهای تقویت شده طوری ساخته شده‌اند که تعداد زیادی از زیرپنجره‌های منفی را کنار می‌گذارند در حالی که تقریباً همه‌ی موارد مثبت را شناسایی می‌کنند (آستانه‌ی یک طبقه‌بند تقویت شده می‌تواند به گونه‌ای تنظیم شود که نرخ اشتباه منفی نزدیک صفر شود). به منظور دستیابی به نرخ اشتباه منفی پایین، از طبقه‌بندهای ساده‌تر استفاده می‌شود تا بیشتر زیرپنجره‌ها را قبل از استفاده از طبقه‌بندهای پیچیده‌تر کنار بگذارند.



شکل ۳-۴ ویژگی اول و دوم آدابوست. هر دو ویژگی در سطر بالایی نشان داده شده‌اند و سپس بر روی یک چهره تمرینی در سطر پایینی پوشانده شده‌است.

شکل کلی فرایند شناسایی یک درخت تصمیم‌گیری است که در این سامانه آبشار نامیده شده‌است که در شکل ۳-۵ مشاهده می‌کنید. یک نتیجه مثبت از طبقه‌بند اول، ارزیابی طبقه‌بند دوم را انجام می‌دهد که طوری تنظیم شده‌است که نرخ شناسایی بالایی داشته‌باشد. یک نتیجه مثبت از طبقه‌بند دوم، ارزیابی طبقه‌بند سوم را انجام می‌دهد و این روند به همین شکل ادامه دارد. یک خروجی منفی در هر نقطه باعث می‌شود تا آن زیرپنجره کنار گذاشته‌شود.

مراحل درون این آبشار توسط آموزش طبقه‌بندها با استفاده از آدابوست و سپس تنظیم آستانه برای کمینه کردن اشتباه منفی، انجام می‌شود. یادآوری می‌شود که آستانه آدابوست پیش‌فرض، طوری طراحی شده‌است که به نرخ خطای پایین بر روی مجموعه‌ی آموزش دست یابد. به طور کلی یک آستانه پایین‌تر، یک نرخ شناسایی بالاتر و نرخ اشتباه مثبت بالاتر را حاصل می‌شود.



شکل ۳-۵ طرح کلی یک شناساگر آبشاری. مجموعه‌ای از طبقه‌بندهای سری شده به هر زیرپنجره اعمال می‌شود. طبقه‌بند ابتدایی تعداد زیادی از نمونه‌های منفی را با یک پردازش کوچک کنار می‌گذارد. پردازش‌های بیشتر می‌تواند هر شکلی داشته باشد به عنوان مثال چندین مرحله دیگر از طبقه‌بندها.

فرآیند آموزش طبقه‌بند آبخاری شامل دو نوع مصالحه است. در بیشتر طبقه‌بندها با ویژگی‌های بیشتر، به نرخ شناسایی بالاتر و نرخ اشتباه مثبت کمتری دست یافته‌اند. در طبقه‌بندهایی با ویژگی‌های بیشتر به زمان بیشتری برای محاسبات نیاز است. در اصل می‌توان سامانه‌ی بهینه‌ای را تعریف کرد که به منظور کمینه کردن تعداد ویژگی‌های ارزیابی‌شده، ویژگی‌های زیر را کنترل نماید:

۱- تعداد مراحل طبقه‌بند

۲- تعداد ویژگی‌های هر مرحله

۳- آستانه هر مرحله

یافتن این موارد بهینه مسئله فوق‌العاده سختی است.

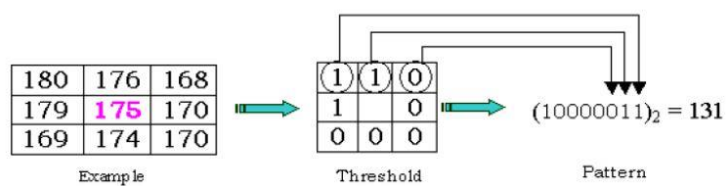
در بعضی موارد به جای استفاده از ویژگی‌های شبه‌هار از ویژگی‌های الگوی محلی باینری استفاده می‌شود. در ادامه به توضیح مختصر این ویژگی‌ها خواهیم پرداخت.

۳-۱-۴ الگوی محلی باینری

علاوه بر استخراج ویژگی‌های شبه‌هار که بحث شد، الگوی محلی باینری، یک انتخاب دیگر به منظور آموزش طبقه‌بند است که در کاربردهای مختلف بسیار خوب عمل کرده‌است مثل طبقه‌بندی بافت و تقسیم‌بندی^۲. عملکرد الگوی محلی باینری معمولی اینگونه است که، پیکسل‌های یک تصویر را توسط مقایسه‌ی پیکسل مرکزی تصویر با همسایگی‌های ۳ در ۳ این پیکسل، برچسب‌زنی می‌کند و نتیجه را به عنوان یک عدد باینری در نظر می‌گیرد. شکل ۳-۶ محاسبات الگوی محلی باینری را نشان می‌دهد [۱۴].

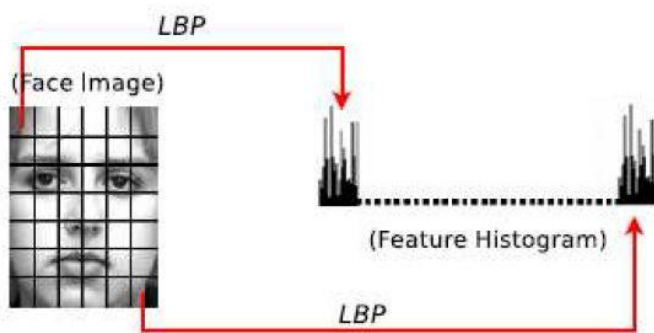
^۱Tradeoff

^۲Segmentation



شکل ۶-۳ محاسبات الگوی محلی باینری [۱۴].

سپس این اعداد به عنوان نماینده‌ی پنجره در نظر گرفته می‌شوند و یک هیستوگرامی ایجاد می‌شود همان‌طور که در شکل ۷-۳ مشاهده می‌کنید [۱۴].



شکل ۷-۳ نماینده تصویر براساس الگوی محلی باینری [۱۴].

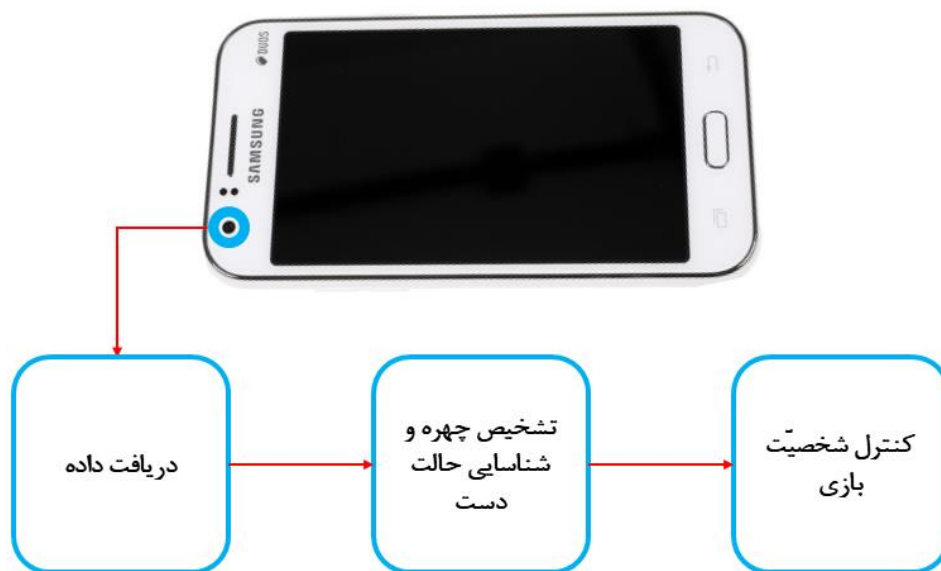
فصل چهارم

روش پیشنهادی

همان‌طور که گفته شد، در این پژوهش قصد توسعه یک بازی اندرویدی برای گوشی‌های موبایل را داریم که در آن شخصیت درون بازی از طریق حرکات دست و سر کاربر کنترل می‌شود. این بازی اندرویدی شامل دو بخش اصلی می‌شود: قسمت پردازشی و قسمت گرافیکی بازی.

قسمت پردازشی، یک سامانه‌ی مبتنی بر روش‌های پردازش تصویر و بینایی‌ماشین است که اقدامات پردازشی بر روی تصاویر دریافتی از دوربین جلوی گوشی موبایل، انجام می‌گیرد. در این قسمت از کتابخانه‌ی OpenCV استفاده می‌کنیم.

روند کلی این پژوهش در شکل ۴-۱ نشان داده شده‌است.



شکل ۴-۱ روند کلی پژوهش.

همان‌طور که گفته شد در قسمت پردازش این پژوهش از روش‌های پردازش تصویر و بینایی ماشین استفاده شده‌است. به دلیل اینکه در این پژوهش هدف اصلی اجرای بازی ویدئویی بر روی گوشی موبایل است، بایستی که قسمت‌های پردازشی به اندازه‌ی کافی سریع باشند تا قابل اجرا بر روی گوشی موبایل باشد. به همین منظور تمرکز بر روی روش‌های بلادرنگ است. مهم‌ترین قسمت بخش پردازشی، بلوک

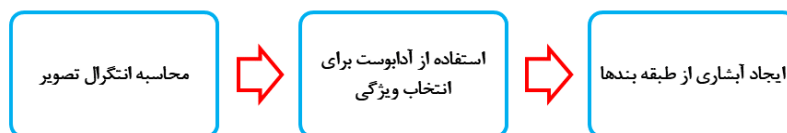
تشخیص چهره و شناسایی حالت دست از شکل ۴-۱ است. به منظور انجام این قسمت از روش طبقه‌بند آبخاری استفاده کرده‌ایم. در قسمت‌های بعدی به تشریح این مراحل خواهیم پرداخت.

۴-۱ استفاده از روش ویولا-جونز

در این پژوهش، به منظور توسعه بازی ویدئویی برای دستگاه‌های موبایل، باید تمام روش‌های مورد استفاده در بخش‌های مختلف این فرآیند، از سرعت بالایی برخوردار باشند تا هنگامی که این بخش‌ها با یکدیگر ادغام می‌شوند همچنان بر روی دستگاه موبایل به صورت بلادرنگ اجرا شود و افت سرعت مشاهده نشود. این موضوع زمانی اهمیت می‌یابد که گوشی موبایل، علاوه بر قسمت پردازشی، وظیفه‌ی اجرای قسمت‌های گرافیکی مربوط به صحنه‌ی بازی ویدئویی را نیز برعهده دارد.

۴-۱-۱ انتخاب روش

با بررسی مقاله‌های مختلف در حوزه‌ی تشخیص و شناسایی اشیا به صورت بلادرنگ و اهمیت سرعت در این پایان نامه، الگوریتم ویولا-جونز را انتخاب کردیم. این الگوریتم شامل ۳ مرحله است. که این مراحل را در شکل ۴-۲ مشاهده می‌کنید.



شکل ۴-۲ الگوریتم ویولا جونز.

- استفاده از انتگرال تصویر و ویژگی‌های مستطیل به جای استفاده از پیکسل‌ها، سرعت محاسبات را افزایش می‌دهد.
- بهره‌گیری از روش آدابوست به عنوان روشی که موثرترین ویژگی‌ها را انتخاب می‌کند، باعث حذف بسیاری از ویژگی‌های غیر مهم می‌شود که این امر باعث کاهش تعداد محاسبات و در نتیجه افزایش سرعت خواهد شد.

• در قسمت طبقه‌بندی، سری کردن مجموعه‌ای از طبقه‌بندها به صورت آبشاری، طوریکه در هر مرحله، فقط نمونه‌های مثبت (شی مورد نظر) حق عبور مجموعه را دارند. هر زمان که یکی از این طبقه‌بندها یک نمونه را به عنوان نمونه‌ی منفی تشخیص دهد، بلافاصله فرآیند خاتمه می‌یابد که این بینش خود باعث افزایش سرعت الگوریتم می‌شود. به مجموعه‌ی این طبقه‌بندهای سری شده، طبقه‌بند آبشاری گفته می‌شود.

جزئیات بیشتر این الگوریتم در فصل ۳ بیان شده‌است.

۴-۱-۲ مشخصات اجرایی پژوهش

در این پژوهش، شخصیت درون بازی ویدئویی توسط چهره و ۳ حرکت دست کاربر کنترل می‌شود. حرکات دست کاربر شامل موارد زیر هستند:

- کف دست به سمت بالا
- کف دست به سمت راست
- کف دست به سمت چپ

در شکل ۴-۳ نمونه‌ای از این حرکات را مشاهده می‌کنید.

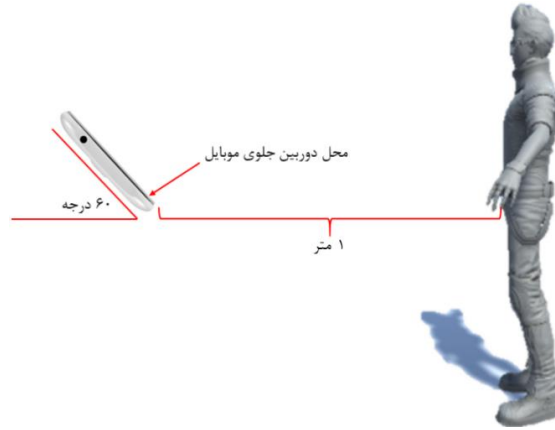


شکل ۴-۳ حالت‌های مختلف دست کاربر در این پژوهش.

همچنین تصویر چهره‌ی کاربر باید از روبه‌رو باشد.

کاربر باید در مقابل گوشی موبایل به صورت ایستاده قرار گیرد و فاصله‌ی کاربر از گوشی موبایل در بهترین حالت ۱ متر است. ارتفاع قرارگیری گوشی موبایل و مقدار زاویه آن با سطح افق در شکل ۴-۴ نشان داده شده‌است. حداقل نسخه اندروید، ۴/۱ است و گوشی موبایل از لحاظ سخت‌افزاری حداقل یک

گوشی میان‌رده باشد. منظور از میان رده یعنی ۴ هسته‌ای با فرکانس ۱/۵ گیگاهرتز و ۳ گیگابایت رم باشد.



شکل ۴-۴ نحوه‌ی قرارگیری گوشی موبایل و کاربر در مقابل موبایل.

۳-۱-۴ الگوریتم پیشنهادی

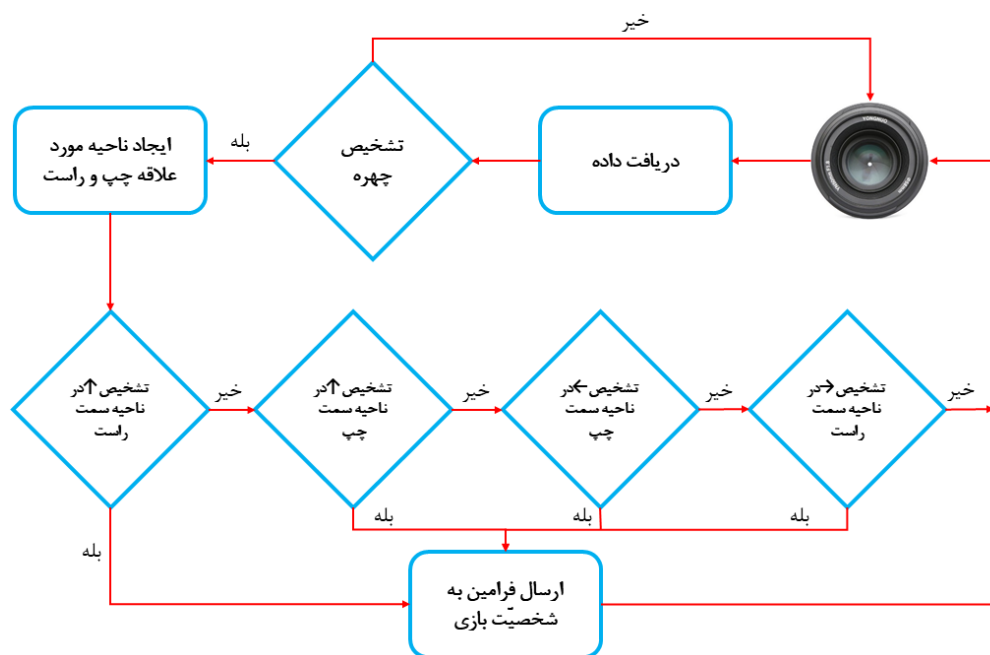
به منظور تشخیص چهره و حالت‌های دست از روشی که در شکل ۴-۵ نشان داده شده‌است در برنامه-نویسی استفاده کرده‌ایم تا بتوانیم سرعت انجام پردازش را افزایش دهیم. همان‌طور که در این روندنا مشاهده می‌کنید پس از دریافت داده از دوربین جلوی گوشی موبایل، تشخیص چهره انجام می‌گیرد. این تشخیص توسط جستجو در کل تصویر انجام می‌شود. بعد از تشخیص چهره نوبت به تشخیص حالت دست می‌رسد؛ در این مرحله به منظور کاهش محاسبات از ایده زیر استفاده کرده‌ایم:

کاربر در زمان اجرای بازی ویدئویی باید در مقابل گوشی موبایل قرار گیرد و با چهره و حرکات دست خود بازی را انجام دهد. طبیعی است که همیشه موقعیت دست‌ها، پایین چهره و در دو طرف آن قرار می‌گیرند. در این پژوهش از همین موضوع استفاده کرده‌ایم و پس از تشخیص چهره با توجه به موقعیت چهره به تعریف دو ناحیه‌ی مورد علاقه از تصویر می‌پردازیم و فقط در این نواحی عملیات تشخیص حالت دست را انجام می‌دهیم. این نواحی با رنگ قرمز در شکل ۴-۶ مشخص شده‌اند. به منظور تعریف ناحیه مورد علاقه سمت چپ و راست از مقادیر X_{min} ، X_{max} ، Y_{min} و Y_{max} که در شکل ۴-۶

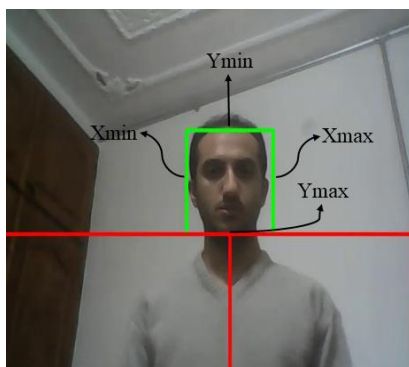
مشخص شده‌اند کمک می‌گیریم. نحوه تعریف محدوده X و Y ناحیه سمت راست از نگاه دوربین، به ترتیب در روابط ۱-۴ و ۲-۴ ذکر شده‌اند. همچنین محدوده‌ی X و Y ناحیه سمت چپ از نگاه دوربین، به ترتیب در روابط ۳-۴ و ۴-۴ ذکر شده‌است. منظور از ROI ناحیه مورد علاقه است.

با استفاده از تعریف این دو ناحیه و انجام عملیات تشخیص حالت دست در این دو ناحیه، می‌توانیم سرعت پردازش را نسبت به زمانی که در کل تصویر به دنبال موقعیت دست هستیم، افزایش دهیم.

بعد از تعریف نواحی مورد علاقه، تشخیص دست به سمت بالا به ناحیه سمت راست اعمال می‌شود، در صورت تشخیص، این حرکت به شخصیت بازی منتقل می‌شود و به سراغ دریافت قاب بعدی می‌رود. در غیر این صورت به ناحیه سمت چپ اعمال می‌شود و این مراحل تا زمانی ادامه می‌یابد تا تمامی شناساگرها به نواحی اعمال شوند. لازم به ذکر است که برای حرکت سمت چپ فقط ناحیه چپ بررسی می‌شود و برعکس.



شکل ۴-۵ روندنمای تشخیص چهره و حرکت‌های دست.



شکل ۴-۶ نمایش نواحی مورد علاقه و متغیرهای کادر مرکزی چهره.

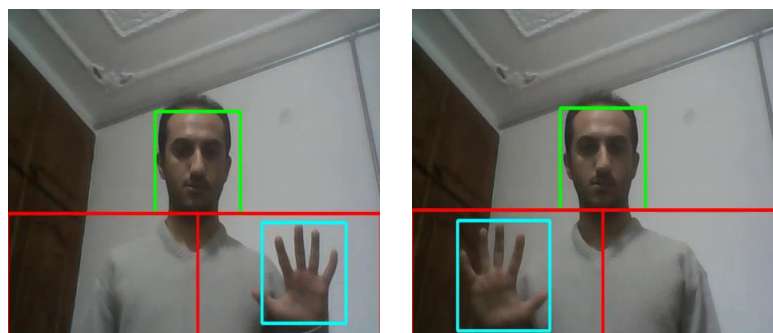
$$X_{right_roi} = \left[0, \frac{Xmin+Xmax}{2} \right] \quad (1-4)$$

$$Y_{right_roi} = \left[Ymax, \text{ارتفاع تصویر} \right] \quad (2-4)$$

$$X_{left_roi} = \left[\frac{Xmin+Xmax}{2}, \text{عرض تصویر} \right] \quad (3-4)$$

$$Y_{left_roi} = \left[Ymax, \text{ارتفاع تصویر} \right] \quad (4-4)$$

در شکل ۴-۷ اعمال الگوریتم ذکر شده بر روی چند قاب، که کاربر حرکات مختلفی انجام داده است را نمایش می دهد.



(ب)

(الف)



(ت)

(پ)

شکل ۴-۷ تشخیص چهره و حرکات کاربر. (الف) و (ب) حرکت بالا. (پ) حرکت چپ از نگاه دوربین. (ت) حرکت راست.

۴-۱-۴ آموزش طبقه‌بندها و ایده استفاده‌شده در طبقه‌بندها

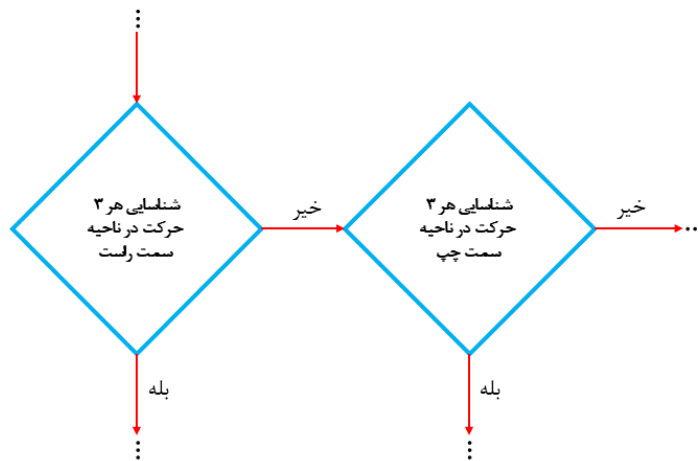
همان‌طور که گفته شد از الگوریتم ویولا-جونز در تشخیص اجزای مختلف بدن استفاده شده‌است. در مجموع از ۴ طبقه‌بند آبخاری در این پژوهش استفاده کرده‌ایم:

- طبقه‌بند آبخاری تشخیص چهره
- طبقه‌بند آبخاری تشخیص حالت کف دست به سمت بالا
- طبقه‌بند آبخاری تشخیص حالت کف دست به سمت چپ
- طبقه‌بند آبخاری تشخیص حالت کف دست به سمت راست

در حالت‌های چند کلاسه معمولاً یک طبقه‌بند آموزش داده می‌شود که تمامی کلاس‌ها را طبقه‌بندی می‌کند، با توجه به آزمایشات انجام شده، در طبقه‌بندهای آبخاری هرچه تعداد تصاویر مورد استفاده در

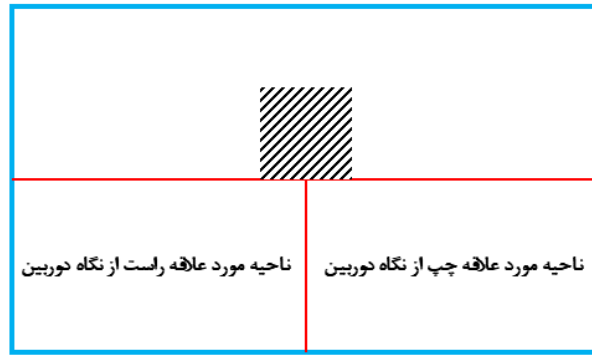
آموزش طبقه‌بند بیشتر شود، طبقه‌بند از لحاظ سرعت انجام تشخیص در یک قاب تصویر، کندتر خواهد شد، به همین دلیل تصمیم گرفتیم برای هر یک از حرکات دست، یک طبقه‌بند آموزش دهیم. در ادامه به توضیح این مطلب می‌پردازیم.

فرض کنید که یک طبقه‌بند آموزش داده‌ایم که سه حرکت بالا، چپ و راست دست را شناسایی می‌کند. همچنین فرض کنید که زمان لازم برای شناسایی هر یک از این حرکات با یکدیگر برابر باشد. بنابراین زمان لازم برای شناسایی هر یک از این حرکات، $\frac{1}{3}$ زمان شناسایی کل است. برای حالتی که یک طبقه‌بند سه کلاسه وجود دارد فرآیند شناسایی طبق روال فرضی شکل ۴-۸ خواهد بود.



شکل ۴-۸ روندنمای فرضی در حالت استفاده از یک طبقه‌بند برای شناسایی هر ۳ حرکت دست.

زمان لازم برای شناسایی حرکت‌های مختلف در حالتی که یک طبقه‌بند ۳ کلاسه داریم در نواحی مختلف طبق روابط زیر خواهد بود. برای انجام محاسبات می‌توان از شکل ۴-۹ کمک گرفت که مربع هاشور خورده چهره کاربر را نشان می‌دهد.



شکل ۹-۴ تشخیص چهره و تعریف نواحی مختلف.

زمان لازم برای شناسایی را در دو حالت طبقه‌بندی مانند زیر در نظر می‌گیریم:

Computation time for 3 classes classifier : t

Computation time for 1 classes classifier : $\frac{t}{3}$

$$up_{right_roi} = t \quad (۵-۴)$$

$$up_{left_roi} = 2 t \quad (۶-۴)$$

$$left = 2 t \quad (۷-۴)$$

$$right = t \quad (۸-۴)$$

حال همین محاسبات را برای حالتی که ۳ طبقه‌بند یک کلاسه داریم، انجام می‌دهیم روند انجام الگوریتم این بار طبق شکل ۵-۴ خواهد بود.

$$up_{right_roi} = \frac{t}{3} \quad (۹-۴)$$

$$up_{left_roi} = 2 \times \frac{t}{3} \quad (۱۰-۴)$$

$$left = 3 \times \frac{t}{3} \quad (۱۱-۴)$$

$$right = 4 \times \frac{t}{3} \quad (12-4)$$

همان طور که ملاحظه می کنید فقط در حالت دست به سمت راست، طبقه بند ۳ کلاسه سرعت پردازش بیشتری خواهد داشت. در بقیه موارد، ۳ طبقه بند یک کلاسه عملکرد بهتری خواهند داشت.

در فرآیند آموزش از تعداد زیادی تصویر مثبت و منفی و همچنین دو نوع ویژگی استفاده شده است. اندازه‌ی تصاویر به گونه‌ای انتخاب شده است که فرآیند آموزش خیلی طولانی نشود. در ادامه به تشریح این موارد خواهیم پرداخت.

- در آموزش شناساگر چهره از ۳۰۰۰ تصویر مثبت (تصویر چهره) و ۱۵۰۰ تصویر منفی (تصاویر غیر چهره) استفاده شده است. در فرآیند آموزش طبقه بند مربوط به چهره، از استخراج ویژگی الگوی محلی باینری استفاده شده است. تمامی نمونه‌ها به اندازه ۲۴*۲۴ تغییر اندازه یافته‌اند. تعداد مراحل آموزش ۲۰ مرحله است. در مرحله ۱ تا ۵ تعداد ویژگی‌ها به ترتیب برابر ۳، ۴، ۴، ۵ و ۵ است و تعداد کل ویژگی‌ها ۱۳۹ است.
- شناساگر مربوط به حالت دست به سمت بالا، با ۲۰۰۰ تصویر مثبت و ۵۰۰۰ تصویر منفی آموزش دیده است. اندازه‌ی تصاویر ۲۴*۲۴ است و ویژگی‌های شبه‌هار از تصاویر استخراج می‌شود. تعداد مراحل آموزش ۲۰ مرحله است. در مرحله ۱ تا ۵ تعداد ویژگی‌ها به ترتیب برابر ۱۳، ۱۴، ۱۵، ۱۰ و ۱۵ است و تعداد کل ویژگی‌ها ۴۲۰ می‌باشد.
- شناساگر مربوط به حالت دست به سمت چپ، با ۲۰۰۰ تصویر مثبت و ۵۰۰۰ تصویر منفی آموزش دیده است. اندازه‌ی تصاویر ۲۴*۲۴ است و ویژگی‌های شبه‌هار از تصاویر استخراج می‌شود. تعداد مراحل آموزش ۱۸ مرحله است. در مرحله ۱ تا ۵ تعداد ویژگی‌ها به ترتیب برابر ۱۰، ۸، ۱۰، ۸ و ۱۲ است. تعداد کل ویژگی‌ها برابر ۳۳۵ می‌باشد.
- شناساگر مربوط به حالت دست به سمت راست، با ۲۰۰۰ تصویر مثبت و ۵۰۰۰ تصویر منفی آموزش دیده است. اندازه‌ی تصاویر ۲۴*۲۴ است و ویژگی‌های شبه‌هار از تصاویر استخراج می‌شود. تعداد مراحل آموزش ۲۰ مرحله است. در مرحله ۱ تا ۵ تعداد ویژگی‌ها به ترتیب برابر ۱۱، ۱۶، ۲۲، ۲۰ و ۱۷ است. تعداد کل ویژگی‌ها برابر ۶۰۶ می‌باشد.
- در فرآیند آزمایش طبقه‌بندها، مقدار مؤلفه‌ای که مشخص می‌کند اندازه‌ی تصویر در هر مقیاس چقدر کاهش یابد (Scale Factor)، برابر ۱/۱ است. پارامتری که مشخص می‌کند که یک

مستطیل نامزد چه تعداد همسایگی برای نگه داشته شدن باید داشته باشد (MinNeighbor)، برابر ۲ است. حداقل اندازه شی شناسایی شده باید ۱۰ پیکسل در ۱۰ پیکسل باشد.

۴-۲ نرم‌افزارها، کتابخانه‌ها و زبان‌های برنامه‌نویسی مورد استفاده

به منظور توسعه یک نرم‌افزار اندرویدی، از محیط‌های برنامه‌نویسی مختلفی می‌توان استفاده کرد مثل نرم‌افزارهای اکیلیپس و اندروید استودیو. در این پژوهش از اندروید استودیو نسخه ۳,۵ استفاده شده است. محیط برنامه‌نویسی این نرم‌افزار شامل ترکیبی از زبان‌های برنامه‌نویسی مختلف است شامل xml، جاوا و کوتلین^۱ که کاربر می‌تواند به انتخاب خود یکی از زبان‌های جاوا یا کوتلین را انتخاب نماید. زبان جاوایی که در اندروید استودیو استفاده می‌شود مخلوطی از زبان جاوای استاندارد و کتابخانه‌های مختلف مخصوص سیستم‌عامل اندروید است. علاوه بر این زبان‌های برنامه‌نویسی، قابلیت استفاده از زبان ++C نیز در این نرم‌افزار وجود دارد که بر طبق تجربه در صورت نوشتن برنامه‌ها به این زبان، سرعت پردازش تا ۴ الی ۵ برابر بیشتر از زبان جاوا خواهد بود.

در قسمت پردازشی این پژوهش سعی بر آن شد تا قسمت شناسایی اجزای مختلف با استفاده از زبان ++C انجام شود. کتابخانه‌ی OpenCV نسخه ۳,۴,۳ در قسمت دریافت قاب‌های تصویر و انجام الگوریتم‌های پردازش تصویر استفاده شده است.

نرم‌افزار اندروید استودیو یک نرم‌افزار صرفاً برنامه‌نویسی است و به منظور طراحی بازی ویدئویی نمی‌توان از آن استفاده کرد. به همین منظور از نرم‌افزار یونیتی ۳ بعدی نسخه ۵,۶,۷ استفاده شده است. یونیتی یک موتور بازی‌سازی معروف است و از زبان‌های برنامه‌نویسی سی‌شارپ و جاوا اسکریپت پشتیبانی می‌کند. برای برقراری ارتباط بین بخش پردازشی و صحنه بازی ویدئویی، باید بین نرم‌افزار اندروید استودیو و یونیتی ارتباط برقرار شود. روش‌های مختلفی برای این کار وجود دارد:

^۱Kotlin

- روش اول ایجاد یک پروژه یونیتی و ایجاد طراحی مورد نیاز و ساخت یک پروژه Gradle و وارد کردن این پروژه در نرم افزار اندروید استودیو است. در این روش یک Activity مربوط به نرم افزار یونیتی خواهیم داشت. برقراری ارتباط بین این Activity با بخش پردازشی، نسبت به برقراری ارتباط بین دو Activity ایجاد شده توسط نرم افزار اندروید استودیو کار بسیار سخت تری است. به علاوه که ناسازگاری نسخه های Gradle در دو نرم افزار همواره مشکل ساز خواهد بود.

- روش دوم ساخت پروژه یونیتی مورد نظر به عنوان یک پروژه اکلیپس و وارد کردن آن در اندروید استودیو به عنوان یک پروژه اکلیپس است. با این روش دیگر مشکل ناسازگاری نسخه های Gradle وجود نخواهد داشت زیرا که در قسمت یونیتی یک پروژه اکلیپس خواهیم داشت که فاقد Gradle است و Gradle در نرم افزار اندروید استودیو اضافه می شود. اما مشکل برقراری ارتباط بین Activity مربوط به صحنه بازی و بخش پردازشی همچنان برقرار است.

- روش سوم که در این پژوهش انجام شده است؛ استفاده از تشابه زبان های جاوا و سی شارپ و تغییر طرز نگرش به مسئله است. یعنی به جای وارد کردن پروژه یونیتی درون اندروید استودیو، پروژه اندروید استودیو را وارد یونیتی نماییم. پس از ایجاد یک برنامه ی سی شارپ در نرم افزار یونیتی و کپی کردن برنامه ای که در اندروید استودیو به زبان جاوا نوشته شده بود و ایجاد تغییرات بسیار اندک توانستیم برنامه ی قسمت پردازشی را درون نرم افزار یونیتی به زبان سی شارپ پیاده سازی نماییم.

برای دریافت قاب های تصویر از کلاس WebCamTexture مربوط به کتابخانه ی یونیتی Engine استفاده شده است و الگوریتم های پردازشی با استفاده از کتابخانه ی OpenCV برای یونیتی نسخه ۲،۱،۳ پیاده سازی شده اند.

یکی از مشکلات نرم افزار یونیتی غیر رایگان بودن کتابخانه های مختلف آن است. آخرین نسخه کتابخانه OpenCV با قیمت حدود ۸۵ یورو در فروشگاه یونیتی قابل دریافت است. به همین دلیل مجبور شدیم از نسخه قدیمی کتابخانه OpenCV که در منابع غیر رسمی به صورت رایگان قابل دریافت است، استفاده کنیم.

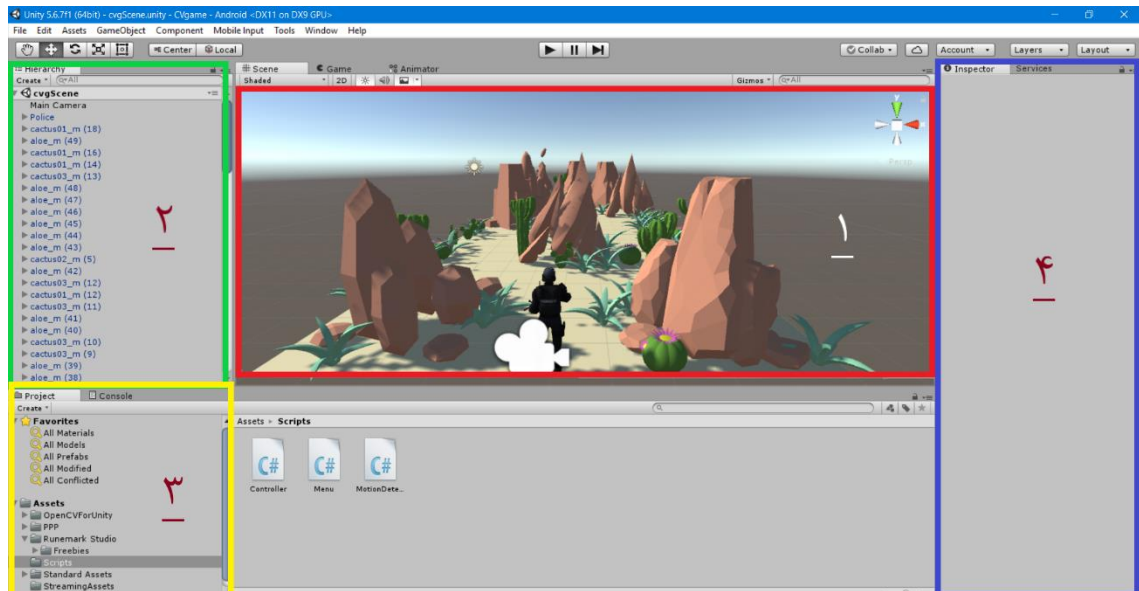
۳-۴ ارتباط بین کاربر و شخصیت درون بازی

پس از تشخیص چهره و حالت‌های دست کاربر، باید حرکات معادل آن‌ها به شخصیت درون بازی ارسال شود. این حرکات شامل موارد زیر هستند:

- تشخیص چهره کاربر معادل حرکت راه رفتن شخصیت بازی است.
- حالتی که هم چهره تشخیص داده شده‌است و هم دست به بالا اشاره می‌کند معادل حرکت پریدن شخصیت بازی است.
- تشخیص همزمان چهره و اشاره دست به سمت راست معادل حرکت چرخش به سمت راست، است.
- تشخیص همزمان چهره و اشاره دست به سمت چپ معادل حرکت چرخش به سمت چپ، است.

۴-۴ طراحی بازی ویدئویی

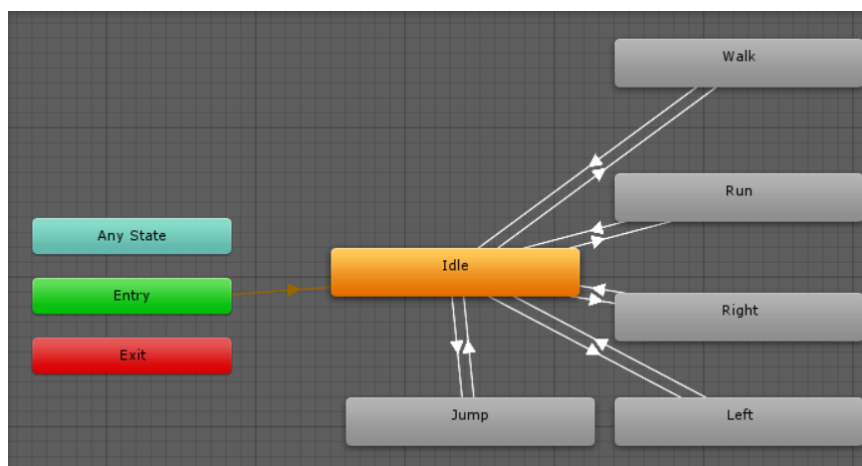
یک پروژه‌ی یونیتی از قسمت‌های مختلفی تشکیل می‌شود. صحنه‌ی بازی، قسمت سلسله مراتب، قسمت پروژه و قسمت بازرسی که در شکل ۴-۱۰ این موارد را به ترتیب با اعداد ۱ تا ۴ مشاهده می‌کنید.



شکل ۴-۱۰ نمایی از محیط نرم افزار یونیتی.

در قسمت صحنه بازی کاربر می‌تواند اشیاء بازی را جایگذاری نماید یا با استفاده از کشیدن و رها کردن ماوس آن‌ها را وارد نماید. قسمت سلسله مراتب تمامی اشیاء اضافه شده به صحنه بازی را نشان می‌دهد. در قسمت پروژه تمامی بسته‌ها و کتابخانه‌هایی که کاربر نیاز دارد، آورده می‌شود. قسمت بازرسی برای دیدن و تغییر دادن ویژگی‌های اشیاء بازی و فایل‌ها مورد استفاده قرار می‌گیرد.

به منظور اجرای حرکات مختلف یک جان‌دهنده^۱ ایجاد کرده‌ایم که با استفاده از توابع بولی^۲ بین انیمیشن‌های مختلف جا به جا می‌شود. این جان‌دهنده را در شکل ۴-۱۱ مشاهده می‌کنید.



شکل ۴-۱۱ جان‌دهنده طراحی شده برای اجرای انیمیشن‌های مختلف.

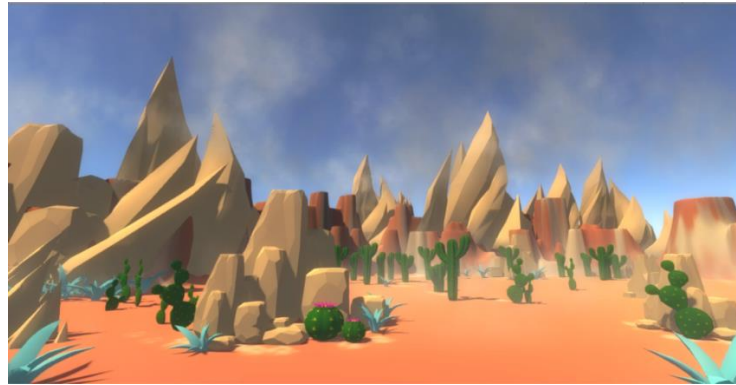
در این جان‌دهنده در حالتی که هیچ حرکتی شناسایی نشده است، کاراکتر در حالت ایده‌آل قرار می‌گیرد و ثابت می‌ایستد. در زمانی که چهره کاربر تشخیص داده می‌شود، توسط یک متغیر بولی این موضوع به برنامه کنترل حرکات، انتقال داده می‌شود و سپس مولفه بولی مربوط به راه رفتن به حالت درست تغییر می‌کند و شخصیت بازی به حالت راه رفتن می‌رود. زمانی که چهره شناسایی نشود، عدم وجود چهره به

^۱Animator

^۲Boolean

برنامه کنترل حرکات انتقال می‌یابد و مولفه بولی مربوط به راه رفتن به حالت اشتباه می‌رود و شخصیت بازی به حالت ایده‌ال برمی‌گردد. برای سایر حرکات نیز همین اتفاق می‌افتد.

برای محیط بازی از محیط‌های بیابانی استفاده شده‌است. تصویری از بسته PolyDesert در شکل ۴-۱۲ نشان داده شده است.



شکل ۴-۱۲ اجزای مختلف بسته PolyDesert که در محیط بازی از آن‌ها استفاده شده‌است.

این بسته شامل اجزای مختلف بیابانی است که هر جزء در دو نسخه مخصوص رایانه و مخصوص موبایل ایجاد شده‌است. اجزای موبایل از لحاظ ابعاد کوچکتر از اجزای رایانه هستند که باعث می‌شود اجرای بازی ویدئویی برای دستگاه موبایل راحت‌تر انجام گیرد. در این پژوهش از اجزای مخصوص موبایل استفاده شده‌است. در شکل ۴-۱۳ نمایی از صحنه‌ی بازی را مشاهده می‌کنید.



شکل ۴-۱۳ محیط بازی ویدئویی.

داستان این بازی ویدئویی، حرکت دادن شخصیت بازی و پیمودن مسیر در نظر گرفته شده است توسط حرکات کاربر، طوری که با موانع برخوردی نداشته باشد و به انتهای مسیر برسد.

۴-۵ نسخه‌های مختلف بازی ویدئویی

یکی از مزایای استفاده از موتور بازی‌سازی یونیتی، قابلیت ایجاد بازی ویدئویی برای دستگاه‌های مختلف است. به همین دلیل این بازی ویدئویی علاوه بر سیستم‌عامل اندروید، برای سیستم‌عامل ویندوز نیز توسعه یافته است.

فصل پنجم

نتایج، جمع‌بندی و پیشنهادها

در این فصل به بررسی نتایج تجربی و ارزیابی دقت و سرعت بازی ویدئویی طراحی شده در شرایط مختلف می پردازیم.

لازم به ذکر است، با توجه به نبود تحقیق مشابه این پایان نامه در زمینه اجرای بازی ویدئویی با حرکات دست، بر روی گوشی موبایل، به منظور مقایسه نتایج این پژوهش با سایر کارهای انجام شده، از نسخه مخصوص سیستم عامل ویندوز استفاده خواهیم کرد. برای سایر مقایسه های صورت گرفته از گوشی موبایل استفاده شده است.

در این پژوهش به دلیل اینکه برای هر حرکت دست یک طبقه بند آموزش دیده است، حالتی که کاربر به عنوان مثال حرکت الف را انجام داده باشد و حرکت ب شناسایی شود، هیچ گاه اتفاق نخواهد افتاد. آزمایش دقت شامل موارد زیر می شود:

- انجام بازی در حالت ایده آل و ثبت دقت شناسایی.
- انجام بازی توسط کاربر، در شرایط نوری مختلف و سنجش میزان دقت شناسایی.
- مقایسه میزان دقت در دو نسخه ویندوز و اندروید.
- مقایسه میزان دقت شناسایی بین نسخه ویندوز و سایر تحقیقات این زمینه.

آزمایش سرعت شامل موارد زیر می شود:

- اجرای بازی بر روی دو سامانه مختلف رایانه و گوشی موبایل و مقایسه سرعت اجرای آن ها.
- مقایسه میزان سرعت شناسایی بین نسخه ویندوز و سایر تحقیقات این زمینه.

۵-۱ آزمایش دقت

۵-۱-۱ سنجش میزان دقت حرکات در حالت ایده آل

در این آزمایش کاربر تعداد ۵۰ حرکت از هر جزء این پژوهش را انجام می دهد تا دقت شناسایی بازی ویدئویی ثبت شود.

جدول ۵-۱ میزان دقت شناسایی

میزان دقت	تعداد حرکت‌های شناسایی نشده	تعداد حرکت‌های شناسایی شده	تعداد حرکت‌های انجام شده	
۷۸٪	۱۱	۳۹	۵۰	اشاره به بالا در ناحیه مورد علاقه چپ
۸۲٪	۹	۴۱	۵۰	اشاره به بالا در ناحیه مورد علاقه راست
۸۶٪	۷	۴۳	۵۰	اشاره به چپ
۸۶٪	۷	۴۳	۵۰	اشاره به راست
۱۰۰٪	۰	۵۰	۵۰	شناسایی چهره
۸۶/۴٪				دقت کل

در این پژوهش، عامل سرعت محاسبات برای ما بیشتر از دقت، اهمیت دارد و به سراغ الگوریتمی رفته‌ایم که بتوانیم تا حد امکان اجرای بلادرنگ داشته باشیم. به همین دلیل میزان دقت ۸۶/۴ درصد برای این پژوهش دقت قابل قبولی است.

۵-۱-۲ سنجش میزان دقت در شرایط نوری مختلف

کاربر تعداد ۱۰ حرکت از هر جزء بازی ویدئویی را در شرایط نوری ضعیف انجام می‌دهد و میزان دقت آن ثبت می‌شود. همان‌طور که در جدول ۵-۲ مشاهده می‌کنید، شرایط نوری یک عامل تعیین کننده در این پژوهش محسوب می‌شود و نسبت به جدول ۵-۱ کاهش محسوس میزان دقت مشاهده می‌شود و برای داشتن خروجی مورد انتظار، شرایط نوری ایده‌ال یک عامل مهم محسوب می‌شود.

جدول ۵-۲ میزان دقت شناسایی در شرایط نوری ضعیف.

میزان دقت	تعداد حرکت‌های شناسایی نشده	تعداد حرکت‌های شناسایی شده	تعداد حرکت‌های انجام شده	
۴۰٪	۶	۴	۱۰	اشاره به بالا در ناحیه مورد علاقه چپ
۶۰٪	۴	۶	۱۰	اشاره به بالا در ناحیه مورد علاقه راست
۵۰٪	۵	۵	۱۰	اشاره به چپ
۶۰٪	۴	۶	۱۰	اشاره به راست

۵-۱-۳ مقایسه میزان دقت در رایانه و گوشی موبایل

پس از انجام آزمایش در دو دستگاه مختلف، از لحاظ دقت شناسایی تفاوتی مشاهده نشد. و میزان دقت شناسایی در رایانه نیز همانند جدول ۵-۱ است.

۵-۱-۴ مقایسه میزان دقت با سایر کارهای انجام شده

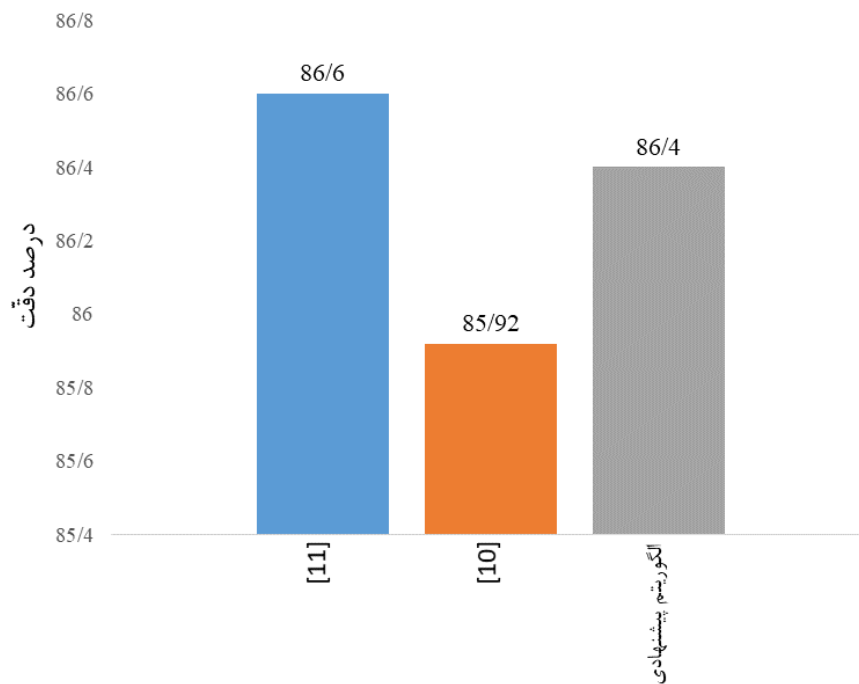
به منظور انجام مقایسه میزان دقت شناسایی از کارهای پیشین استفاده می‌کنیم. در [۱۰] از روش ویولا-جونز استفاده شده‌است. برای اطلاعات بیشتر در مورد کارهای پیشین به فصل ۲ مراجعه نمایید.

جدول ۳-۵ نتایج پژوهش چن و همکارانش [۱۰].

زمان پردازش	میزان دقت	تعداد اشتباه شناسایی شده	تعداد حرکت-های شناسایی نشده	تعداد تشخیص‌های درست	تعداد حرکت-های انجام شده	
3/049	۷۱٪	۲۹	۰	۷۱	۱۰۰	حالت دو انگشت
1/869	88/8%	۰	۱۰	۸۰	۹۰	کف دست
2/829	۹۹٪	۱	۰	۹۹	۱۰۰	مشت
2/452	84/9%	۷	۷	۷۹	۹۳	حالت انگشت کوچک
	85/92%					دقت کل

همان‌طور که در جدول ۳-۵ مشاهده می‌کنید مقدار دقت کل [۱۰] به ازای ۴ حرکت دست 85/92 درصد است که ما در این پژوهش طبق جدول ۳-۵ به ازای ۴ حرکت دست و ۱ تشخیص چهره، به مقدار دقت ۸۶/۴ درصد دست یافته‌ایم که دقت کمی بهبود یافته‌است با اینکه بر روی گوشی موبایل اجرا می‌شود. علاوه بر این، در این پژوهش هیچ‌گاه حرکتی اشتباه شناسایی نمی‌شود، چرا که برای هر حرکت یک طبقه‌بند اختصاص یافته‌است که از این لحاظ نیز نسبت به روش [۱۰] بهبود داریم.

مقاله دیگری که برای مقایسه میزان دقت استفاده خواهیم کرد، [۱۱] است. در [۱۱]، با استفاده از تکنیک‌های پردازش تصویر، ۱۱ حرکت دست را طبقه‌بندی کرده‌اند و به مقدار دقت کل ۸۶/۶ درصد رسیده‌اند. ما نیز در این پژوهش دقت مشابهی کسب کرده‌ایم. در شکل ۳-۵ مقایسه بین این دو مقاله با روش پیشنهادی را مشاهده می‌کنید.



شکل ۵-۱ مقایسه میزان دقت.

۵-۲ آزمایش سرعت

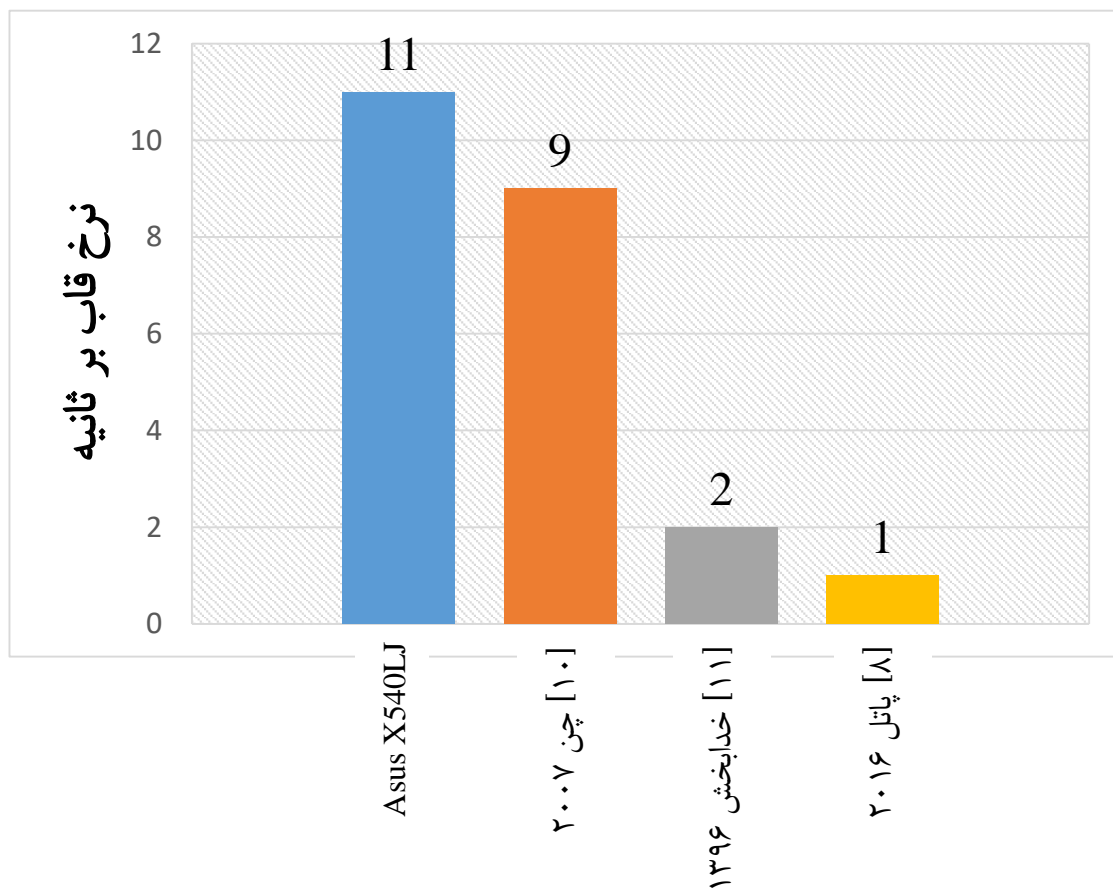
۵-۲-۱ مقایسه سرعت اجرا در رایانه و گوشی موبایل

زمانی که بازی در حال اجرا است، زمان لازم برای پاسخ به حرکت کاربر در رایانه ۲ ثانیه و در گوشی موبایل ۳ ثانیه است که این مورد به دلیل تفاوت در قدرت پردازشی این دو دستگاه اتفاق می‌افتد. پس از شناسایی حرکت کاربر، شخصیت بازی بلافاصله متوقف می‌شود و حرکت معادل حالت دست کاربر را انجام می‌دهد. در حین انجام این فرآیند کاربر باید دست خود را در حالت مورد نظر نگاه‌دارد. در نسخه ویندوز به نرخ قاب ۱۱ و در نسخه اندروید با اجرا بر روی گوشی موبایل سامسونگ گلکسی Note 9 به نرخ ۸ قاب بر ثانیه دست یافتیم.

۲-۲-۵ مقایسه سرعت اجرا در نسخه ویندوز با سایر کارهای انجام شده

برای مقایسه سرعت، از نسخه ویندوز استفاده خواهیم کرد تا قابل مقایسه با سایر کارهای انجام شده باشد. در این مقایسه نیز از کارهای پیشین استفاده می‌کنیم. به منظور انجام مقایسه پژوهش‌های مختلف از مولفه‌ی نرخ قاب بر ثانیه استفاده می‌کنیم.

در اجرای بازی ویدئویی از یک لپ‌تاپ Asus X540LJ استفاده کرده‌ایم.



شکل ۲-۵ مقایسه میزان سرعت با سایر پژوهش‌ها.

همان‌طور که در شکل ۲-۵ مشاهده می‌کنید در مقایسه با سایر مقاله‌ها از سرعت بهتری بهره‌مند هستیم با اینکه قسمت گرافیکی بازی نیز اجرا می‌شود.

در شکل ۳-۵ تصاویری از اجرای بازی توسط کاربر را مشاهده می‌کنید.



(الف)



(ب)



(پ)

شکل ۳-۵ اجرای بازی توسط کاربر (در نمایش تصویر کاربر، سمت چپ و راست از نگاه دوربین در نظر گرفته می‌شود).
 (الف) تشخیص چهره و انجام حرکت راه رفتن. (ب) تشخیص سمت چپ و انجام چرخش به چپ. (پ) تشخیص سمت
 راست و انجام چرخش به راست.

۳-۵ جمع‌بندی

در این پایان‌نامه، یک بازی ویدئوی برای سیستم‌عامل اندروید توسعه دادیم که برای کنترل شخصیت بازی از یک روش نوین استفاده می‌کند. کنترل شخصیت بازی با استفاده از حرکات دست و تشخیص چهره کاربر صورت می‌گیرد. این روش نوین فواید زیادی می‌تواند داشته باشد از جمله اینکه باعث رفع خستگی ناشی از مشغله روزانه در عضلات شود. همچنین توسعه اینگونه بازی‌ها برای دستگاه‌های موبایل می‌تواند برای علاقه‌مندان به توسعه بازی‌های ویدئویی یک انتخاب بسیار خوب از لحاظ کسب درآمد باشد.

در انجام این پایان‌نامه به سراغ نرم‌افزارهای گوناگون و زبان‌های برنامه‌نویسی مختلفی رفتیم که در نهایت تصمیم گرفتیم که از موتور بازی‌سازی یونیتی و زبان برنامه‌نویسی سی‌شارپ استفاده نماییم. در قسمت‌های پردازشی شامل شناسایی حالت‌های مختلف از روش‌های بلادرنگ استفاده کرده‌ایم و ایده‌های مختلفی در اعمال این الگوریتم‌ها به کار بردیم تا بتوانیم یک بازی ویدئویی با سرعت مناسب توسعه دهیم.

سرانجام این بازی را برای دو سیستم عامل ویندوز و اندروید توسعه دادیم. که از لحاظ سرعت و دقت پردازش در مقایسه با سایر کارهای انجام شده به نتایج مطلوبی دست یافتیم.

۴-۵ پیشنهادها و کارهای آینده

در این تحقیق به دلیل توسعه بازی ویدئویی برای دستگاه‌های موبایل باید قسمت‌های مربوط به تشخیص و شناسایی از لحاظ بار محاسباتی، سبک باشند تا سرعت اجرای قابل قبولی در پایان مشاهده شود. برای رسیدن به این هدف ایده‌های مختلفی به کار برده‌ایم.

یک پیشنهاد دیگر برای افزایش سرعت و کاهش خطای عدم شناسایی، تعریف نواحی جستجوی جدید است. یعنی به منظور تشخیص حالت دست در قاب تصویر بعدی از موقعیت دست در قاب تصویر فعلی

استفاده نماییم. به عنوان مثال اگر در قاب تصویر فعلی حالت دست در موقعیت Y_{min} , X_{max} , X_{min} و Y_{max} تشخیص داده شده است؛ در قاب تصویر بعدی این ناحیه را به اندازه‌ی به عنوان مثال ۱۰ پیکسل از هر طرف افزایش دهیم و درون این ناحیه‌ی جدید به دنبال حالت دست باشیم. با این کار می‌توانیم ناحیه‌ی مورد بررسی را کاهش دهیم که باعث افزایش سرعت و کاهش خطا خواهد شد.

پیشنهاد دیگر جدا کردن قسمت شناسایی و ردیابی پژوهش است به گونه‌ای که نیاز نباشد در هر قاب تصویر، شناسایی صورت گیرد و بعد از اولین شناسایی به ردیابی شی بپردازیم. این کار باعث می‌شود که شی شناسایی شده از پایداری بیشتری برخوردار شود و مشکل عدم شناسایی در برخی از قاب‌های تصویر از بین برود.

1-Lebowitz J, and Klug Ch, (2011), “**Interactive Storytelling for Video Games**”, Focal Press, pp.5.

2-Rodrigues M, Macedo D, Pontes H, Serpa Yv, and Serpa Yg, (2016), “A serious game to improve posture and spinal health while having fun”, **2016 IEEE International Conference on Serious Games and Applications for Health (SeGAH)**, pp 5.

3-Goldani M, and Goldani A, (2018), “A review study on effective factors of developing the Finnish gaming industry and some suggestions for Iran’s game industry”, **2018 2nd National and 1st International Digital Games Research Conference: Trends, Technologies, and Applications (DGRC)**, pp 123.

4-www.digikala.com

5-Qingtang L, Yang W, Linjing W, Jingxiu H, and Peng W, (2015), “Design and Implementation of a Serious Game Based on Kinect”, **2015 International Conference of Educational Innovation through Technology (EITT)**, pp 13.

6-Zhu Y, and Yuan B, (2014), “Real-Time Hand Gesture Recognition with Kinect for Playing Racing Video Games”, **2014 International Joint Conference on Neural Networks (IJCNN)**, pp 3240.

۷-فرزین نژاد ف، و راستی ج، (۱۳۹۶)، "یک بازی رایانه‌ای-ورزشی برای تمرین سایه‌زنی درایو فورهند به کمک فناوری کینکت ایکس‌باکس"، سومین کنفرانس ملی و اولین کنفرانس بین‌المللی بازی‌های رایانه‌ای؛ فرصت‌ها و چالش‌ها.

8-Sarkar S, Patel V, and Chellappa R, (2016), “Deep Feature-based Face Detection on Mobile Devices”, **arXiv.org by Cornell University**.

9-Elrefaei L, Alharthi A, Alamoudi H, Almutairi S, and Al-rammah F, (2017), “Real-time face detection and tracking on mobile phones for criminal detection”, **2017 2nd International Conference on Anti-Cyber Crimes (ICACC)**.

10-Chen Q, Georganas N, and Petriu E, (2007), "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features", **2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007**, pp 1.

۱۱-خدابخش و، و اسلامی م، (۱۳۹۶)، "تشخیص حرکات دست برای تعامل انسان-رایانه با استفاده از تکنیک پردازش تصویر"، دومین کنفرانس بین‌المللی پژوهش‌های دانش‌بنیان در مهندسی کامپیوتر و فناوری اطلاعات.

12-Viola P, and Jones M, (2001), "Rapid Object Detection using a Boosted Cascade of Simple Features", **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001**, pp 511.

13-Brownlee J, (2016), "**Master Machine Learning Algorithms**", pp.136.

14-Chang J, (2008), "Face Detection using LBP feature", **CS 229 Final Project Report**.

15-Mao Q, Sun H, Liu Y, and Jia R, (2019), "Mini-Yolov3: Real-Time Object Detector for Embedded Applications", **IEEE Access**, Vol. 7, pp. 133529-133538.

16-Mazzia V, Khaliq A, Salvetti F, and Chiaberge M, (2020), "Real-Time Apple Detection System using Embedded Systems With Hardware Accelerators: An Edge Ai Application", **IEEE Access**, Vol. 8, pp. 9102-9114.

17-Fang W, Wang L, and Ren P, (2020), "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments", **IEEE Access**, Vol. 8, pp. 1935-1944.

Abstract

Video games today have a special place among entertainments. Along with the benefits listed for video games, overuse can also cause a variety of problems. In the meantime, body-controlled games can be a good alternative to other video games because they stimulate people.

In this thesis, we intend to develop a video game for Android devices so that we can control the personality of the game by identifying the user's hand gestures and recognizing his face.

One of the problems with this, was the lower processing power of mobile phones than computers. For this reason, we have used real-time methods to identify hand gestures and face detection. In this thesis we used cascading classifiers with extraction of Haar-like features and local binary pattern and various ideas were used to apply them to the most ideal case. By recognizing their faces, hands, and direction, we control the game's character.

Processing algorithms are applied to the images captured from the front camera of the mobile phone and we use the OpenCVForUnity library. Unity software is used to design the video game and the programming language is C #.

At the end, we developed the game for both Windows and mobile operating systems, and the accuracy for both operating systems was 86.4 percent and in the Windows and Android versions we achieved frame per second rate 11 and 8, respectively.

Keywords: video games, mobile phones, hand gesture recognition, face detection, cascade classifiers.



Faculty of Electrical and Robotics Engineering

M.Sc. Thesis in Telecommunication Systems Engineering

**Design and implementation of an interactive video game, for android
devices using cascading classifier**

By: Mahmood Mohammadzadeh

Supervisor:

Dr Hossein Khosravi

February 2020