دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فن آوری اطلاعات

پایان‌نامه کارشناسی

گرایش نرم افزار

عنوان

طراحی و پیاده سازی واسط کاربری برای کنترل تلفن همراه با اشاره دست

نگارش

مینا قدیمی عتیق

استاد راهنما

دکتر محمد رحمتی

آبان ۹۵

به نام خدا

تعهدنامه اصالت اثر

تاریخ:

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

اینجانب مینا قدیمی عتیق متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

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

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

مینا قدیمی عتیق

امضا

نخستین سپاس و ستایش از آن خداوندی که بنده کوچکش را در دریای بیکران اندیشه، قطره‌ای ساخت تا وسعت آن را از دریچه‌های ناب آموزگارانی بزرگ به تماشا بنشیند.

سپاس از استاد گران‌قدرم، جناب آقای دکتر رحمتی که مرا یاری کردند.

تقدیم به

پدر و مادر عزیز و مهربانم،

که در سختی‌ها و دشواری‌های زندگی همواره یاری دلسوز و فداکار و پشتیبانی محکم و مطمئن برایم بوده‌اند.

چکیده

هدف این پروژه طراحی و پیاده‌سازی نرم‌افزار واسط کاربری برای ارتباط انسان و دستگاه تلفن همراه که با اشاره دست و بدون تماس انجام می‌شود، است. از آنجا که امروزه در این بین تلفن همراه جزو پر استفاده‌ترین وسایل ارتباطی است و هر روزه ساعت زیادی صرف کار با تلفن همراه می‌شود، در این پروژه؛ اشاره دست به منظور راه ارتباطی با کامپیوتر و به ویژه تلفن همراه در نظر گرفته شده است. با استفاده از ابزارهای پردازش تصویر، کتابخانه‌ها و توابع موجود رشته تصاویر ویدئویی به صورت برخط از دوربین اخذ و یا از قبل ذخیره شده دریافت می‌شود و به فریم‌های تشکیل دهنده خود تقسیم می‌شوند و این فریم‌ها به عنوان تصاویر ورودی مورد پردازش و تغییر قرار می‌گیرند. با استفاده از اطلاعات رنگی مربوط به پوست بدن و پس از اعمال پردازش‌هایی ناحیه مربوط به دست از تصویر کلی استخراج می‌شود. در مرحله بعدی یک مدل به دست آمده برای تشخیص ناحیه دست در تصاویر ورودی مورد استفاده قرار می‌گیرد. شبکه‌های عصبی به کار رفته، شبکه پرسپترون سه لایه و شبکه عمیق بر پایه گوگل‌نت[1] است. در نهایت عملی متناظر با اشاره انتسابی انجام می‌شود. عمل تشخیص نوع اشاره با استفاده از شبکه عصبی انجام می‌شود.

در این پروژه، کتابخانه OpenCV برای پردازش تصویر به کاررفته است و زبان برنامه‌ی نوشته شده، پایتون می‌باشد.

واژه‌های کلیدی:

اشاره دست، پردازش تصویر، شبکه عصبی

[1] GoogLeNet

فهرست عناوین

صفحه

فهرست اشکال

صفحه

فهرست جداول

صفحه

1

فصل اول

مقدمه

مقدمه

کامپیوتر با ورود به عرصه مدرن، نقش کلیدی در تغییر روش‌های ارتباطی بازی کرده و به تمامی زوایای زندگی شخصی و اجتماعی ما نفوذ کرده است. جست‌وجو در وب، نوشتن نامه، انجام بازی ویدیویی و ذخیره و یا بازیابی اطلاعات شخصی و یا اداری، تنها مثال‌های کوچکی از استفاده از کامپیوتر و یا دستگاه‌های مبتنی بر کامپیوتر در زندگی شخصی ما هستند. با توجه به افزایش تولید و کاهش قیمت کامپیوترهای شخصی، تاثیر آن‌ها در زندگی روزمره رو به افزایش است. برای استفاده هرچه بهتر از این پدیده، حوزه‌ی تحقیقاتی در زمینه برنامه‌های کامپیوتری و نحوه تعامل با کامپیوترها پایه‌گذاری شده است و به حوزه‌ای زنده و پویا تبدیل شده است .[1]

با گذشت زمان، تعامل انسال و کامپیوتر از ارتباط متنی به حالت گرافیکی تکامل یافته است. اما همچنان معمول‌ترین وسیله‌های ارتباطی ماوس و صفحه کلید هستند. اما متاسفانه این وسیله‌ها قادر به رفع نیازهای امروزی در کاربردهای واقعیت مجازی نیستند. در نتیجه گسترش راه‌های جایگزین و مناسب برای برقراری ارتباط بین انسان و کامپیوتر جزو علایق محققان در سال‌های اخیر قرار گرفته است. دو نوع بخش در این زمینه وجود دارد: انجام پژوهش در زمینه برقراری ارتباط انسان و کامپیوتر توسط راه‌های طبیعی‌تر و برقراری ارتباط انسان و کامپیوتر برای افراد دارای ناتوانی جسمی. در هر دو حالت دو نوع راه حل وجود دارد: سیستم‌هایی به همراه دستگاه خارجی نصب شونده بر روی بدن انسان، سیستم‌های بدون تماس. سیستم‌های بدون تماس راحتی بیشتری برای کاربران تامین می‌کنند. در این بین بهترین سیستم‌ها برای برقراری ارتباط، سیستم‌های مبتنی بر بینایی هستند. واسط کاربری کاربرپسند[۲] باید چندین ویژگی داشته باشد: عدم برقراری تماس، عدم وابستگی به شرایط نوری، قابل اعتماد و کار در زمان واقعی.[2]

تعامل انسان با کامپیوتر دارای کاربردهای متعددی است که به طور عمده در سه دسته ''کنترل ماشین''، ''تشخیص زبان اشاره'' و ''سیستم‌های بازی مبتنی بر پردازش تصویر'' قرار می‌گیرند. دسته کنترل ماشین شامل مواردی از جمله کنترل ربات با استفاده از بینایی ماشین، کنترل پخش‌کننده موسیقی، کنترل کانال و یا صدای تلویزیون، کنترل ارایه و نشانه‌گر ماوس است. دسته شناسایی زبان اشاره شامل پیاده‌سازی سیستم‌های برقراری ارتباط بین افراد دارای ناتوانی در حرف زدن و شنیدن هستند، است. دسته سوم شامل

[2] User Friendly

پیاده‌سازی کنسول‌های بازی برای بازی بدون سخت‌افزارهای متداول قدیمی همانند دسته بازی و یا صفحه کلید، است. در اکثر کاربردهای ذکر شده، تعامل انسان با کامپیوتر به طور عمده شامل تعامل دست و یا انگشت با ماشین است. در نتیجه تشخیص اشاره دست[۳] شاخه‌ای از تعامل انسان با کامپیوتر است که شامل تفسیر اشاره به اطلاعات معنی‌دار در راستای تعامل است. عبارت ''تشخیص اشاره دست'' به فرایند ردیابی و پیگیری اشاره دست و سپس تفسیر اشاره به فرمان‌های معنی‌دار، گفته می‌شود. تکنولوژی‌های موجود در این زمینه با استفاده از تماس و یا با استفاده از بینایی و بدون تماس می‌توانند باشند. تکنولوژی‌های بر پایه تماس، بر اساس مواردی هم‌چون دستکش داده[۴]، صفحه‌های چندلمسی و ... که از تشخیص‌دهنده‌هایی شامل دما، انعکاس، سرعت، زمان و ... استفاده می‌کنند، شکل می‌گیرند. استفاده از سنسورها و سیم‌ها در دستکش‌ها حرکت دست کاربر را محدود می‌کنند، در نتیجه کاربرد این دستکش‌ها در برقراری ارتباط طبیعی دچار مشکل می‌شود. سیستم‌های مبتنی بر بینایی جایگزین‌هایی هستند که از سیستم‌های پردازش تصویر همچون وب‌کم استفاده می‌کنند. بیشترین امتیاز این سیستم‌ها درجه آزادی برای حرکت دست است، که در برقراری ارتباط طبیعی انسان با کامپیوتر نقش مهم و اساسی دارند. در این روش‌ها تصویر اشاره دست به عنوان ورودی با استفاده از دستگاه‌های ورودی دریافت می‌شود. سپس با استفاده از روش‌های ردیابی دست منطقه دست شناسایی می‌شود. سپس منطقه دست جدا شده به روش‌های موجود برای شناسایی اشاره و دسته‌بندی کننده، ارسال می‌شود. به دلیل دقت حاصل، این روش‌ها به صورت جهانی مورد قبول عموم قرار گرفته‌اند.[3]

اشاره دست زبان بدن عملی سطح بالایی است که توسط کف دست انسان، مکان انگشت‌ها و شکل دست، مشخص می‌شود. اشاره‌ها شامل دو نوع استاتیک و پویا هستند. حالت استاتیک یا ساکن، همان‌طور که از نامش مشخص است، حالت ساکن دست انسان است. حالت پویا نیز از مجموعه‌ای از حرکات دست تشکیل شده است. در این پروژه حالت ساکن یا ایستا مورد بررسی قرار می‌گیرد.[4]

بسیاری از شرایط و محدودیت‌های موجود در هنگام تشخیص اشاره باعث افزایش پیچیدگی این مسئله می‌شود. این موارد می‌توانند شامل پیچیدگی اشاره موجود، پس‌زمینه پیچیده، تغییر در نورپردازی، گرفتگی و موارد از این نوع باشند. [5]

---

[3] Hand Gesture Recognition (HGR)

[4] Data Glove

روش به کار رفته در این پروژه برای برقراری ارتباط توسط اشاره دست، به سه مرحله کلی تقسیم می‌شود. در مرحله اول ورودی به صورت برخط و یا به صورت فایل ذخیره شده دریافت می‌شود. سپس ویدئوی ورودی به فریم‌های تشکیل دهنده تقسیم می‌شود. در نهایت پس از تقسیم به فریم‌ها، تعدادی تصویر در اختیار داریم. تصاویر به دست آمده برای استخراج بهتر اطلاعات به فضای رنگی HSV انتقال داده می‌شوند. برای تشخیص محدوده متعلق به دست از بازه متعلق به پوست در فضای رنگی جدید استفاده می‌شود. در نتیجه بر اساس محدوده مربوط به پوست در فضای رنگی جدید، تصویر باینری که شامل دست و غیر دست است، تشکیل داده می‌شود. برای کاهش نویز تصویر به دست آمده، از فیلتر میانه استفاده می‌شود. پس از اعمال فیلتر، ناحیه دست در تصویر به دست آمده، مشخص می‌شود. حفره‌های موجود که به علت تغییر در نورپردازی ویا موارد مشابه ایجاد شده‌اند، پرشده و تصویر دست نهایی به دست می‌آید. حال با در دست داشتن ناحیه متعلق به دست در تصویر، وارد مرحله دوم می‌شویم. در این مرحله از شبکه عصبی برای تشخیص اشاره استفاده می‌شود. تصویر به دست آمده به شکل موردنیاز تبدیل شده و به عنوان داده آزمایش به شبکه عصبی آموزش داده شده، ارسال شده و خروجی یا همان اشاره دریافت می‌شود. در مرحله سوم متناسب با اشاره خروجی، عملی انجام می‌شود.

در ادامه این پایان‌نامه و در فصل دوم به بررسی پیشینه و چند کار مهم انجام شده در زمینه تعامل انسان و کامپیوتر با استفاده از پردازش تصویر می‌پردازیم. در فصل سوم به بررسی روش پیشنهادی ارایه شده در این پروژه می‌پردازیم. در نهایت در فصل چهارم به بررسی نتایج به دست آمده و جمع‌بندی و کارهای پیشنهادی آینده پروژه می‌پردازیم.

2

فصل دوم

پیشینه

پیشینه

همانطور که پیش از این بیان شد، تعامل در ابزارهای تکنولوژی امروزه بسیار مورد توجه است . در این فصل ، پس از تعریف تعامل ، به بیان روش‌های تعامل مبتنی بر پردازش تصویر می‌پردازیم. چند کار مهم موجود در این زمینه را نیز بررسی کرده و روش‌های به کار رفته در آن‌ها را بیان می‌کنیم.

2.1 تعامل چیست؟

تعامل انسان و کامپیوتر[1] به دانش و فن‌آوری مدرن و پرتنوع مطالعه، طراحی، اجراء، و ارزیابی سامانه‌های محاسباتی درگیر در محاورات و تعاملات مابین کاربران انسانی از یک سو، و رایانه‌ها و عامل‌های هوشمند نرم‌افزاری از سوی دیگر گفته می‌شود.

HCI ، نقطه تقاطع علوم رایانه و علوم رفتارشناسی طراحی و چند علم دیگر است. ارتباط و تعامل انسان و رایانه از طریق واسطی اتفاق می‌افتد که شامل نرم‌افزار و سخت‌افزار است. تعریفی دقیق دیگر آن است که علم تعامل انسان و رایانه، یک رشته مرتبط با طراحی، ارزیابی و پیاده‌سازی سیستم‌های محاسباتی متقابل برای استفاده انسان در مطالعه پدیده‌های پیرامون اوست. این رشته شاخه‌هایی از هر دو طرف درگیر را شامل می‌شود مثلاً گرافیک کامپیوتری، سیستم عامل، زبان‌های برنامه‌نویسی، تئوری ارتباطات و طراحی صنعتی برای قسمت کامپیوتری زبان‌شناسی، روانشناسی و کارایی انسان برای قسمت انسانی آن.

هدف آن تقویت تعاملات کاربر و رایانه به وسیله کاربردی ترکردن رایانه‌هاو برنامه‌های کامپیوتری و مطابقت آنها با نیاز کاربران است. به عبارت دیگر این رشته مرتبط است با:

- روش‌شناسی و فرایندهای طراحی واسطه‌ها
- روش‌های پیاده‌سازی واسطه‌ها
- تکنیک‌های ارزیابی و مقایسه واسطه‌ها
- توسعه طراحی و پیاده‌سازی واسطه‌های جدید

[1] Human-Computer Interaction(HCI)

این رشته در اوایل دهه ی نهم قرن بیستم (۱۹۸۰) میلادی به عنوان شاخه ای تخصصی از علوم کامپیوتر که با علوم شناختی، مهندسی عوامل انسانی و طراحی سر و کار دارد ایجاد شد.

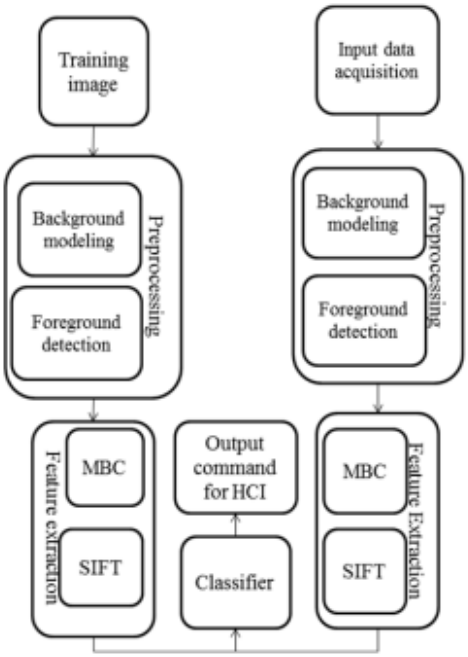
در گذشته،در تعریف تعامل کاربر و کامپیوتر، تمرکز بر واسط‌های کاربری سخت افزاری مانند موس و کیبورد بوده است . این واسط امکان ورودی گرفتن از کاربر و تنظیم خروجی بر اساس آن را فراهم می آوردند. محصولات تعاملی در واقع به صورت دوطرفه عمل می‌کند و هوشمند هستند .یعنی کاربر با کامپیوتر و یا وسایل الکترونیکی و کامپیوتری به صورت تعاملی در ارتباط است.در دهه‌ی هشتم قرن بیستم، اولین جرقه‌ها در تکنولوژی صفحه نمایش لمسی زده شد که بسیار مورد توجه قرار گرفت تا کنون در مرکز توجه کاربران قرار دارد. اما این تکنولوژی نیز مشکلات خاص خود را دارد . از نمونه‌ی این مشکلات می توان به موارد زیر اشاره کرد:

- نیاز به سخت افزار پیچیده و حساس و گران قیمت
- تفاوت نیرو و فشار و حرکت دست کاربران و نیاز به هماهنگ سازی سیستم با کاربر فعلی
- رعایت نشدن بهداشت و امکان انتقال بیماری ها از طریق تماس(خصوصا در سیستم‌های عمومی)

با وجود مشکلاتی از این دست ، به نظر می‌رسد نیاز به روشی دیگر برای تعامل با سیستم‌ها امروزه وجود دارد که در ادامه به بررسی چند نمونه میپردازیم.

2.2 روش‌های تعامل مبتنی بر پردازش تصویر

بلوک دیاگرام روش پیشنهادی در مقاله [5] در شکل زیر نمایش داده شده است.



شکل 1 بلوک دیاگرام روش پیشنهادی در مقاله [5]

در این روش، ابتدا مجموعه داده آموزش و آزمایش از بین داده‌های موجود مشـــخص می‌شـــود. ســـپس عملیات پیش پردازش بر روی داده‌های ورودی انجام می شود. در این عملیات، تصویر دارای فضای رنگی RGB به فضای رنگی سطح خاکستری[1] تبدیل می‌شود. سپس با اعمال فیلتر، نویزهای موجود در تصویر کاهش داده می‌شـــود. با اســـتفاده از آســـتانه‌ســـازی رنگ پوســـت، مدل‌ســـازی پس زمینه انجام می‌شـــود. آستانه‌سازی رنگ پوست روش مناسبی برای جداسازی پیش زمینه شامل دست از پس زمینه است. مدل پوســـتی به کار رفته برای این کار، باید در برابر شـــرایط محیطی و پس‌زمینه پیچیده مقاوم باشـــد. برای تبدیل فضای رنگی عکس از RGB به فضای سطح خاکستری از تبدیل زیر استفاده می‌شود.

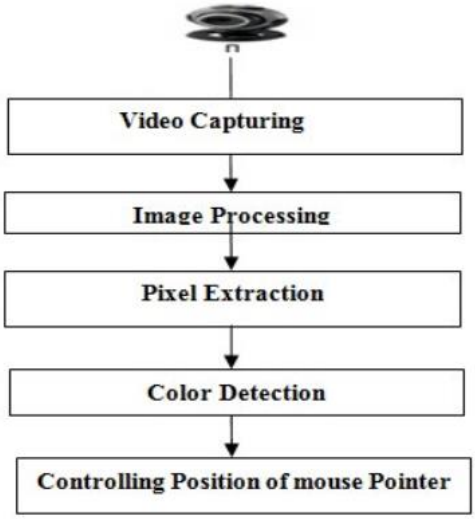
$$(R, G, B) \rightarrow 0.29R + 0.58G + 0.11B$$

سپس با ا ستفاده از SIFT و کدینگ باینری تک ژنی[2] ا ستخراج ویژگی انجام می شود. ا ستخراج ویژگی برای ارسال به دسته‌بندی کننده دارای اهمیت بسیار بالایی است.

بلوک دیاگرام روش پیشنهادی در مقاله [6] در شکل زیر نمایش داده شده است.

[1] Gray Scale

[2] Monogenic binary coding

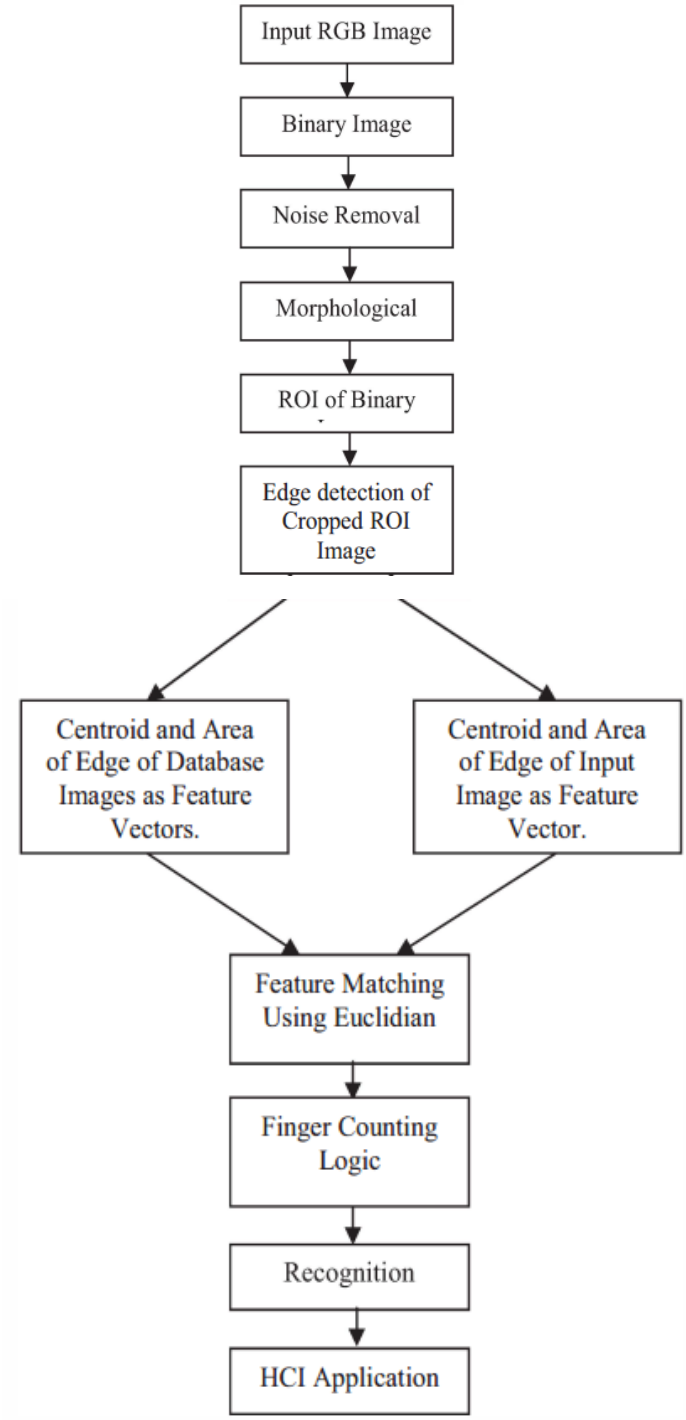


شکل 2 بلوک دیاگرام روش پیشنهادی در مقاله [6]

در این روش ابتدا ویدیوی ورودی دریافت شـــده و وارد بخش پردازش تصـــویر می‌شـــود. در این بخش، قطعه‌بندی تـ صویر[1] در دو مرحله تـ شخیص پو ست و تخمین مدل میانه، انجام می شود. مرحله اول برای تشـــخیص دســـت و مرحله دوم برای حذف پس زمینه به کار می‌رود. در این مقاله از هیچ روش شـــبکه عـ صبی دارای یادگیری ا ستفاده نـ شده و دنباله پیکـ سلی مربوط به حرکت انگـ شت ا ستخراج شده ا ست. سپس مختصات مربوط به دست استخراج شده و مکان اشاره‌گر ماوس با استفاده از آن کنترل می‌شود.

بلوک دیاگرام روش پیشنهادی در مقاله [3] در شکل زیر نمایش داده شده است.

[1] Image Segmetation



شکل 3 بلوک دیاگرام روش پیشنهادی در مقاله [3]

در این روش پس از دریافت ویدیو به عنوان ورودی، قطعه‌بندی دست بر اساس عملیات مورفولوژیکی انجام می‌شود. از محدوده رنگی پوست در فضای رنگی RGB برای جداسازی دست استفاده می‌شود. از پس جداسازی دست از عملیات مورفولوژیکی برای از بین بردن خطا و حفظ لبه‌ها استفاده می‌شود.

سپس در تصویر باینری به دست آمده با استفاده از عملگر Sobel ناحیه مورد علاقه مربوط به دست استخراج می‌شود. در واقع در حال استخراج لبه‌های دست در تصویر هستیم. در ادامه از تعداد انگشت‌های موجود در تصویر و ناحیه قرار گیری انگشت‌ها، برای دسته‌بندی اشاره استفاده می‌شود.

در مقاله [7] از ترکیب اطلاعات حاصل از دست و سر برای تعامل استفاده شده است. در ادامه به بررسی بخش مربوط به دست می‌پردازیم. در این مقاله در مرحله اول حذف پس زمینه با عملیات تفریق فریم و پس‌زمینه انجام می‌گیرد. سپس با استفاده از حد آستانه، بخش مربوط به دست در تصویر تفاضل مشخص می‌شود. پس از تشخیص ناحیه مربوط به دست، زمان استخراج ویژگی فرا می‌رسد. در این مرحله از بزرگترین کانتور در تصویر پیدا می‌شود. سپس با استفاده از تشکیل پوشش محدب[1] تعداد بریدگی‌ها[2] شمرده می‌شود. تعداد بردیدگی‌ها، جهت قرارگیری و حرکت دست، جزو ویژگی‌های به کار رفته در تشخیص اشاره هستند. در انتها نیز از الگوریتم کم‌شیفت[3] برای ردیابی دست استفاده می‌شود.

2.3 جمع‌بندی

با توجه به بررسی‌های صورت گرفته، برای تعامل انسان با کامپیوتر، روش‌ها و ابزارهای زیادی وجود دارد. هرکدام از روش‌ها به اعمال محدودیت‌های خاص خود پرداخته و کاربر را مقید به رعایت شروطی می‌کند. برای مثال در اکثر روش‌های سخت‌افزاری نیاز به تماس وجود دارد. حال ما به دنبال پردازش نرم‌افزاری و کاهش محدودیت‌های ممکن و دستیابی به نتایج بهتر هستیم. از این رو به ارایه روشی برای استخراج بهتر ویژگی‌های دست و دسته‌بندی بهتر اشاره دست می‌پردازیم.

[1] Convex hull

[2] defect

[3] Cam Shift

3

فصل سوم

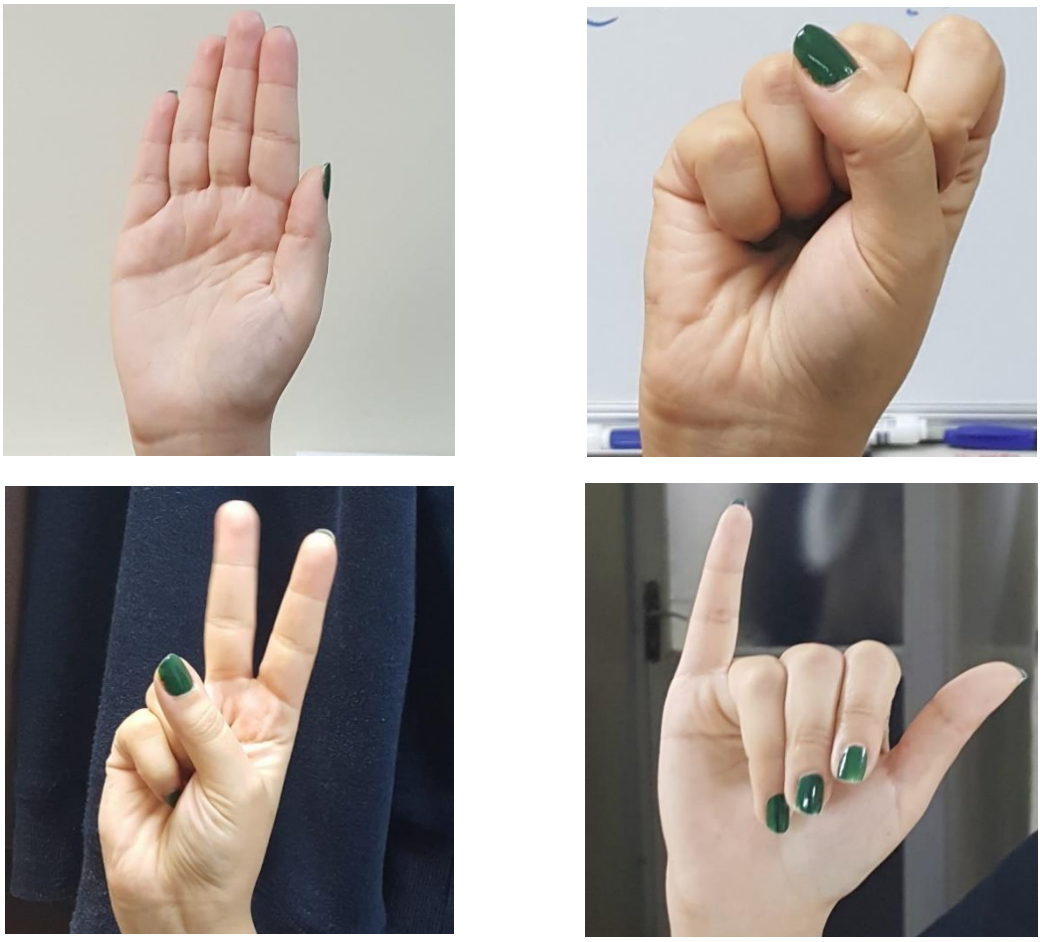
روش پیشنهادی

روش پیشنهادی

در این بخش، پس از ارایه مقدمه، به بررسی روش پیشنهادی می‌پردازیم.

3.1 مقدمه

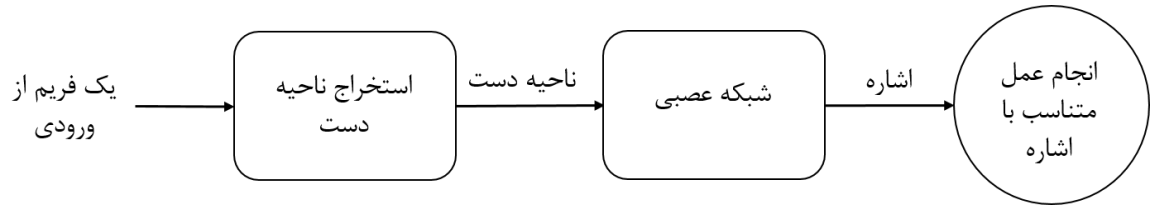
هدف در این پروژه، کنترل تلفن همراه با ا ستفاده از ا شاره د ست ا ست. ا شاره‌های د ست به کار رفته در پروژه به صورت زیر تعریف شده‌اند.



شکل 4 اشاره‌های دست تعریف شده در پروژه

برای کنترل موبایل با استفاده از اشاره‌های در نظر گرفته شده، قدم اول یافتن دست در دنباله ورودی است. در این راستا، از رنگ پوست برای تشخیص ناحیه دست استفاده می‌کنیم. برای استخراج رنگ پوست نیز از دو روش رنگ پوست صورت و مجموعه داده پوست استفاده شده است. پس از مشخص کردن محدوده مربوط به دست و انجام پیش‌پردازش‌های لازم بر روی آن، برشی از تصویر شامل ناحیه دست حاصل می‌شود. حال زمان تشخیص اشاره مربوط به ناحیه برش داده شده می‌رسد. برای این‌کار از دسته‌بند های موجود می‌توان استفاده کرد. در این پروژه از شبکه‌های عصبی استفاده شده است. پس از دریافت اشاره به عنوان خروجی شبکه عصبی، عمل متناظر با آن انجام می‌شود.

بلوک دیاگرام روش پیشنهادی در زیر نمایش داده شده است.

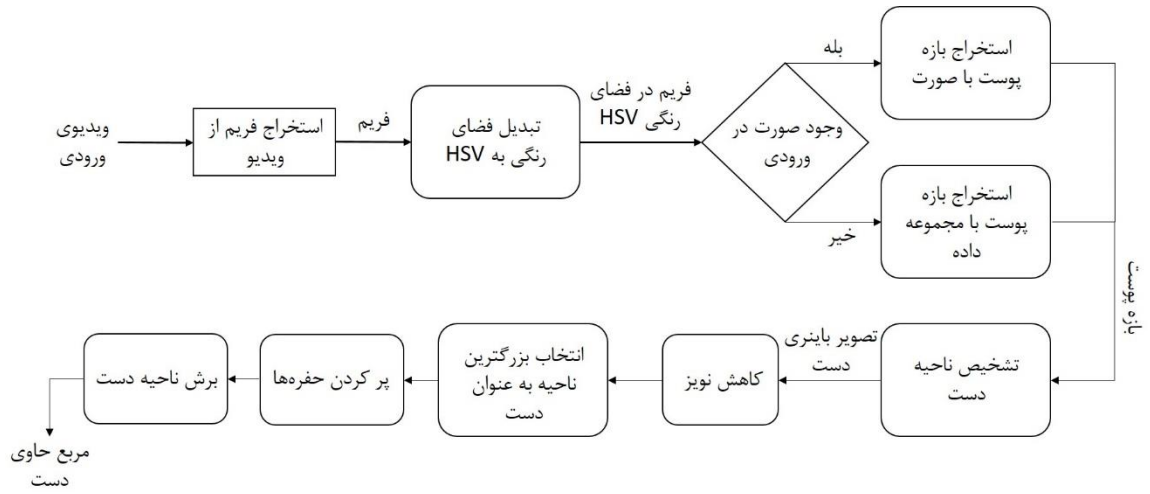


شکل 5 بلوک دیاگرام روش پیشنهادی

در ادامه به توضیح هر کدام از بخش‌ها می‌پردازیم.

3.2 استخراج ناحیه دست

هدف در این بخش دریافت یک فریم از ورودی و مشخص کردن ناحیه دست به صورت ناحیه مربعی است. بخش‌های استفاده شده برای استخراج ناحیه دست در بلوک دیاگرام زیر نمایش داده شده است.



شکل 6 بلوک دیاگرام مربوط به استخراج ناحیه دست

به عنوان اولین قدم در روند کارکرد، ابتدا باید ویدئو به عنوان ورودی دریافت شود. این ورودی می‌تواند به صورت برخط و یا به صورت فایل آماده به برنامه داده شود. برای خواندن ویدئوی ورودی از تابع VideoCapture که در کتابخانه OpenCV موجود است، استفاده شده است. این تابع آدرس ویدئو را، که در واقع شامل نام آن نیز هست، به عنوان ورودی دریافت می‌کند. سپس توسط تابع isOpened عدم خاتمه ویدئو مورد بررسی قرار می‌گیرد و تا زمانی که ویدئو تمام نشده است، تابع read فریم‌های ویدئو را استخراج می‌کند.

نکته قابل ذکر دریافت تصویر تنها از یک دوربین است. برخی از سیستم‌های موجود در تشخیص اشاره دست، توسط چندین دوربین عملیات فیلم‌برداری را انجام داده و در نهایت اطلاعات به دست آمده از هر بخش را با هم ترکیب می‌کنند. مزیت استفاده از چندین دوربین، پوشش زوایای مختلف و برخورداری از حجم اطلاعات بیشتری است. اما در این پروژه تنها از یک دوربین استفاده شده است.

حال فریم‌های ویدئو را در اختیار داریم و قصد داریم تا از فریم‌های موجود، ناحیه مربوط به دست را تشخیص داده و استخراج کنیم. برای این منظور از اعمال فیلتر بر اساس رنگ پوست استفاده شده است.

نکته قابل ذکر تغییر هر اشاره پس از گذشت چندین فریم است. به این صورت که اشاره در هر فریم عوض نشده و طی چندین فریم اشاره جدید وجود ندارد. از این رو به جای پردازش هر فریم، یک فریم از بین چندین انتخاب می‌شود. در بخش‌های زیر به توضیح عملیات انجام داده شده برای استخراج دست می‌پردازیم.

3.2.1 استخراج مولفه‌های مربوط به فام و خلوص رنگ از فریم

ویدئوی دریافت شده به عنوان ورودی و فریم استخراج شده از آن، در فضای رنگی RGB هستند. یعنی دارای سه مولفه رنگی قرمز، سبز و آبی هستند. رنگ پوست انسان از ترکیب دو رنگ قرمز (به دلیل خون) و زرد (به دلیل ملانین[1]) با خلوص رنگ متوسط تشکیل شده است و در کل در سطح پوست تغییرات دامنه‌ای کمی دارد. درنتیجه با استفاده از اطلاعات حاصل از رنگ پوست، می‌توان دست را از تصویر جدا کرد. اما این اطلاعات در فضای رنگی HSV قدرت تمایز بهتری از خود نشان می‌دهند. در نتیجه در ادامه قصد داریم تا فضای رنگی فریم ورودی را به فضای رنگی HSV تبدیل کنیم.

برای استفاده از اطلاعات رنگی فریم‌های موجود، ابتدا توسط فرمول‌های زیر مقادیر $I، R_g, B_y$ به دست می‌آیند.

$$L(x) = 105 * \log_{10}(1 + x + n)$$
$$I = L(g)$$
$$R_g = L(R) - L(G)$$
$$B_y = L(B) - {\left(L(G) - L(R)\right)}/{2b}$$

(1)

مقادیر I، R_g و B_y در واقع مولفه‌های لگاریتمی سه کانال رنگ قرمز و سبز و آبی هستند. برای نشان دادن شدت نور از کانال سبز استفاده شده است، زیرا کانال‌های قرمز و آبی در اکثر دوربین‌ها دارای عملکرد ضعیفی هستند. در فرمول محاسبه L همان‌طور که دیدیم عدد ثابت ۱۰۵ وجود دارد. این عدد برای تبدیل مقیاس خروجی به بازه‌ی $[0,254]$ است. n موجود در فرمول، یک نویز یکنواخت تولید شده در بازه $[0,1)$ است. این نویز برای جلوگیری از تولید لبه‌های مصنوعی ایجاد شده در ناحیه‌های تاریک تصویر افزوده شده است. عدد ثابت ۱ برای جلوگیری از تفاوت زیاد بین رنگ‌های متفاوت موجود، اضافه شده است.

سپس مقادیر فام و خلوص رنگ به صورت زیر استخراج می‌شوند.

[1] melanin

$$Hue = \frac{180}{\pi} \tan^{-2}(R_g, B_y)$$

$$Saturation = \sqrt{R_g^2 + B_y^2}$$

(2)

3.2.2 مشخص کردن بازه‌ی متعلق به پوست و استخراج محدوده پوست با استفاده از بازه به دست آمده

در مرحله قبل، مولفه‌های جدید از تصویر استخراج شد. حال زمان آن فرا رسیده است تا با توجه به بازه مولفه‌های استخراج شده، محدوده مربوط به دست مشخص شود. در نتیجه ابتدا باید بازه‌هایی از مولفه‌ها، که شامل محدوده پوست هستند، مشخص شوند.

پس از مشخص شدن بازه‌های مربوط به پوست، زمان استخراج پوست از تصویر فرا می‌رسد. مولفه‌های فام و خلوص رنگ در تمامی پیکسل‌های فریم به دست می‌آید، سپس با بازه‌بندی ارایه شده مقایسه می‌شود. در صورت صدق در بازه‌بندی پوست، آن پیکسل مقدار ۲۵۵ یعنی رنگ سفید و در غیر این‌صورت، مقدار ۰ یعنی رنگ سیاه به خود می‌گیرد.

حال به موضوع مشخص کردن بازه متعلق به پوست به صورت جزیی نگاه می‌کنیم. برای این‌کار، از دو رویکرد استفاده شده است. از آنجایی که بازه متعلق به پوست دست تا در صورت نیز صدق می‌کند، قصد داریم تا ابتدا به تشخیص صورت بپردازیم. پس از استخراج ناحیه مربوط به صورت و بازه مربوط به صورت، از بازه به دست آمده برای استخراج دست استفاده می‌شود. حال در صورتی که صورت فرد در تصویر موجود نباشد، از بازه به دست آمده از مجموعه داده پوست استفاده می‌شوذ. در زیر به ارایه توضیح دقیق‌تر می‌پردازیم.

استخراج بازه متعلق به پوست با استفاده از داده‌های موجود در مجموعه داده‌ی پوست

در این بخش قصد داریم تا با استفاده از مجموعه داده مربوط به قطعه‌بندی پوست ارایه شده توسط دانشگاه UCI اطلاعاتی در زمینه پوست جمع‌آوری کنیم. برای دانلود و دریافت اطلاعات بیشتر در زمینه این مجموعه داده به لینک (https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation) مراجعه

کنید. این مجموعه داده، دارای ستون ۳ ویژگی و یک ستون کلاس یا دسته است. ۳ ویژگی موجود در مجموعه داده در واقع سه مولفه مربوط به رنگ آبی، سبز و قرمز هستند. کلاس نیز عدد ۰ یا ۱ است که نشان می‌دهد آیا این اطلاعات متعلق به پوست هستند یا نه. نکته قابل توجه در این مجموعه، جمع‌آوری داده‌های پوستی از سه ناحیه اروپا، آسیا و آفریقا انجام شده‌است. در شکل ۳ سطرهایی از مجموعه داده دیده می‌شود.

B	G	R	S
74	85	123	1
73	84	122	1
72	83	121	1
70	81	119	1
70	81	119	1
69	80	118	1
70	81	119	1
70	81	119	1
76	87	125	1
76	87	125	1
77	88	126	1
77	88	126	1
77	88	126	1
78	89	127	1
77	85	125	1

شکل 7 نمایی از مجموعه داده رنگ پوست

عملیات مربوط به این بخش در فایل skin_range.py و در تابع HS_range() انجام می‌شود. فرمت مجموعه داده موجود csv می‌باشد. برای بارگذاری این مجموعه داده از کتابخانه pandas استفاده می‌شود. برای کار با مجموعه داده، ابتدا داده‌های مربوط به پوست که با برچسب یک مشخص شده بودند، جدا و سپس در یک ماتریس ریخته شدند.

حال قصد داریم با استفاده از مجموعه داده، بازه متعلق به پوست را استخراج کنیم. اما همان‌طور که گفتیم، این مجموعه داده مربوط به سه ناحیه مختلف است. از این رو به جای استفاده از کل مجموعه داده برای استخراج بازه، به فکر تقسیم آن به نواحی تشکیل دهنده افتادیم. در واقع ۳ ناحیه تشکیل دهنده را جدا کرده و ناحیه‌ی دارای بیشتری هم‌خوانی با پوست دست کشور ایران برای تعیین بازه استفاده می‌شود. برای تقسیم مجموعه داده به ۳ بخش تشکیل دهنده، از الگوریتم خوشه‌بندی Kmeans استفاده شد.

این الگوریتم با دریافت داده‌ها و مشخص کردن تعداد خوشه‌ها، شروع به خوشه‌بندی کرده و داده‌ها را در سه خوشه جداسازی می‌کند. این الگوریتم در واقع در حال انجام یادگیری بدون نظارت[1] است. زیرا هیچ

[1] Unsupervised Learning

نمونه‌ای از سه خوشه به عنوان ورودی به آن داده نشده است. با مشخص کردن تعداد خوشه‌ها، هربار مراکز خوشه مشخص شده، بر اساس فاصله نقاط تا مراکز خوشه‌ها، خوشه مناسب انتخاب شده و این کار تا رسیدن به دقت و یا تعداد دفعات تکرار موردنظر، ادامه پیدا می‌کند. انتخاب اولیه مراکز خوشه‌ها می‌تواند به روش‌های مختلفی از جمله رندوم، انجام شود. با در نظر داشتن "k-means++" = init الگوریتم در انتخاب مراکز خوشه‌ها هوشمندی به خرج داده و سریع‌تر به جواب می‌رسد. تعداد بیشینه تکرار در کد برابر با ۵۰۰ در نظر گرفته شده است. هم‌چنین تعداد دفعات انتخاب مرکز خوشه جدید نیز برابر با ۱۰۰ در نظر گرفته شده است. در واقع الگوریتم با ۱۰۰ بار انتخاب مرکز خوشه‌های اولیه اجرا شده و در نهایت بهترین نتیجه انتخاب می‌شود. پس از اتمام کار الگوریتم، داده‌های سه خوشه در متغیرهای مختلف ریخته می‌شود. به ازای داده‌های متعلق به هر خوشه، تبدیل به فضای رنگی موردنظر انجام می‌شود. در نهایت محدوده حاصل از هر سه خوشه به عنوان خروجی تابع در نظر گرفته می‌شوند.

در انتها هر پیکسل از فریم، اگر در بازه‌ی مربوط به یکی از خوشه‌ها صدق کند، شرط وجود پوست برقرار است.

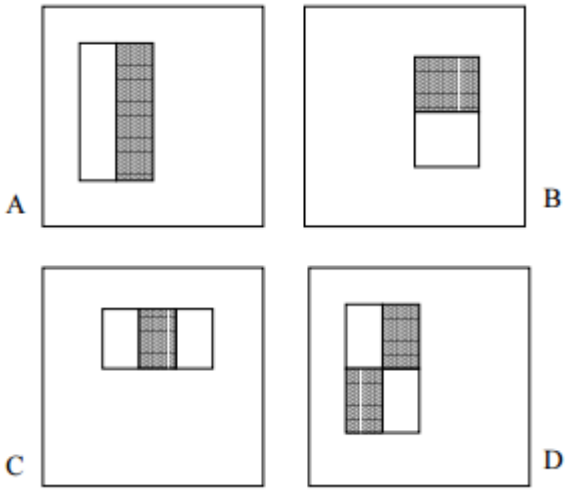
استخراج بازه متعلق به پوست با استفاده از صورت

تغییرات در نورپردازی تصویر باعث ایجاد تغییر در رنگ استخراج شده از پوست می‌شود. درنتیجه محدوده رنگی پوست به راحتی قابل تغییر است و این مسئله باعث ایجاد دشواری در تعیین ناحیه مربوط به پوست می‌شود. برای حل این مشکل، سعی در یافتن روش‌هایی شده است که رنگ را به صورت کاراتری در زمان اجرای برنامه و دریافت ورودی، استخراج نمایند. ایده‌ای که در این زمینه به ذهن می‌آید، استفاده از رنگ پوست صورت است. در نتیجه ابتدا باید محدوده متعلق به صورت از تصویر استخراج شود. برای استخراج صورت از الگوریتم ویولاجونز استفاده می‌شود که در زیر توضیح داده می‌شود.

در این روش برای تشخیص چهره از روش‌های یادگیری ماشین بهره گرفته شده است. به این ترتیب که ابتدا تعدادی ویژگی از تصویر استخراج شده و این ویژگی‌ها به عنوان ورودی به یک الگوریتم یادگیری ماشین داده می‌شوند که روی مجموعه‌ای از داده‌ها آموزش داده شده و به حل مسئله‌ی دو کلاسه‌ی چهره یا غیر چهره می‌پردازد. برای افزایش سرعت از ویژگی های شبه هار[1] استفاده می شود که در ادامه توضیح

[1] Haar-like

داده خواهد شد. برای یادگیری نیز از الگوریتم آدابوست[1] استفاده شده است که در ادامه، مورد بررسی قرار خواهد گرفت.



شکل 8 نمونه ای از ویژگی های مستطیلی. ویژگی های دو مستطیلی در A و B ، سه مستطیلی در C و چهار مستطیلی در D نشان داده شده اند.

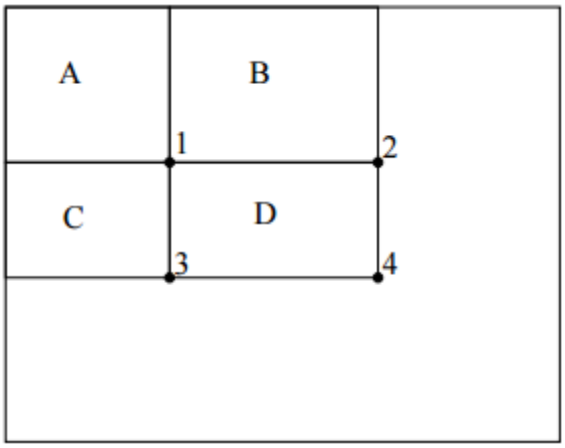
در صورتی که از مقدار پیکسل‌های یک تصویر به عنوان ویژگی استفاده شود، به طور کلی توصیف مناسبی از تصویر صورت نمی‌گیرد. به همین دلیل باید با بهره بردن از روش‌هایی به استخراج ویژگی از تصویر پرداخت. در الگوریتمی که مقاله‌ی [8] ارائه می‌کند از ویژگی های شبه هار استفاده می‌شود. دلیل ترجیح این ویژگی‌ها به موارد دیگر سادگی و سرعت بالای محاسبه‌ی این ویژگی‌ها می‌باشد. سه نوع ویژگی شبه هار استفاده می‌شود. ویژگی دو مستطیلی که مقدار آن برابر با اختلاف مجموع پیکسل‌های دو مستطیل هم اندازه‌ی کنار هم می‌باشد. ویژگی سه مستطیلی حاصل اختلاف مجموع پیکسل‌های دو مستطیل خارجی از مجموع پیکسل‌های مستطیل داخلی است. در نهایت ویژگی چهار مستطیلی اختلاف بین مجموع پیکسل‌های مستطیل‌های قطری از قطر دیگر را محاسبه می‌کند (به شکل ۱۱ مراجعه شود). بدیهی است که تعداد بسیار زیادی از این ویژگی‌ها از تصویر استخراج می‌شوند.

[1] Adaboost

برای محاسبه‌ی ویژگی‌های شبه هار از یک تصویر میانی که از روی تصویر اصلی ساخته می‌شود استفاده می‌شود که به آن تصویر مجتمع[1] گفته می‌شود. تصویر مجتمع در نقطه‌ی (x, y) از رابطه‌ی زیر محاسبه می‌شود.

$$ii(x,y) = \sum_{x' \le x,\ y' \le y} i(x', y')$$

در این فرمول $ii(x,y)$ بیانگر تصویر مجتمع و $i(x,y)$ تصویر اصلی می باشد.



شکل 9 مقدار نقطه‌ی 4 برابر مجموع پیکسل های موجود در تمام ناحیه های A, B, C, D میباشد. مقدار نقطهی 2 برابر A, B و نقطهی 3 نیز برابر A, C میباشد. نقطهی 1 فقط شامل A می باشد. بنابراین برای محاسبهی مجموع پیکسل های ناحیهی کافیست مقدار 4-2-3+1 محاسبه شود.

با استفاده از دو رابطه‌ی زیر می توان تصویر مجتمع را در یک بار گذر از تصویر با استفاده از مقادیر محاسبه شده قبلی در هر مرحله به دست آورد.

$$s(x,y) = s(x,y-1) + i(x,y)$$
$$ii(x,y) = ii(x-1,y) + s(x,y)$$

که در آن $s(x,y)$ مجموع سطری تصویر می باشد. پس از محاسبه‌ی تصویر مجتمع می توان با ۴ بار دسترسی به تصویر جمع پیکسل‌های یک مستطیل مورد نظر را به دست آورد (شکل ۱۲).

پس از استخراج ویژگی‌ها نوبت به مرحله‌ی یادگیری می‌رسد. در این روش از الگوریتم آدابوست برای دسته بندی تصاویر به چهره و غیر چهره استفاده می‌شود. الگوریتم آدابوست از یک الگوریتم یادگیری ضعیف

[1] Integral Image

استفاده کرده و کارایی آن را بهبود می‌بخشد. در ادامه دسته بند[1] ضعیف $h_j(x)$ که در این روش استفاده شده آورده شده است.

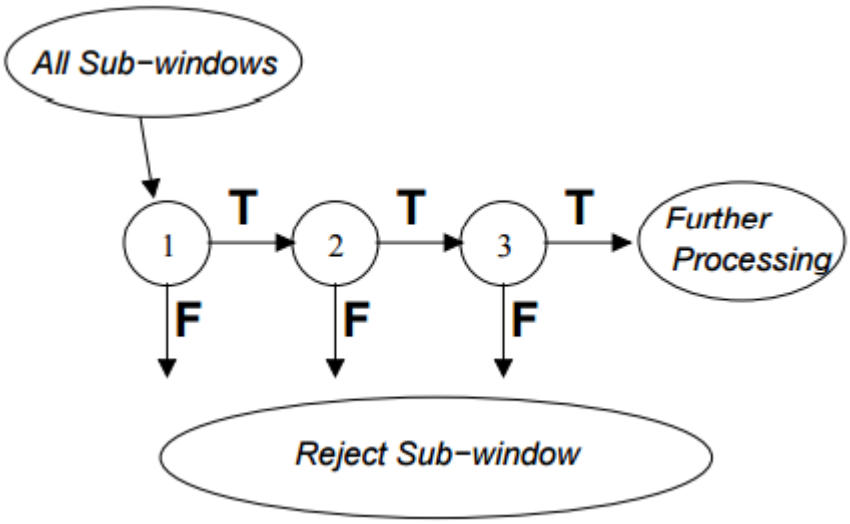
$$h_j(x) = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases}$$

در این عبارت x یک پنجره‌ی ۲۴ در ۲۴ از تصویر می‌باشد. f_j یک ویژگی، θ_j یک مقدار آستانه و p_j مشخص کننده‌ی جهت نامساوی می باشد. حال الگوریتم آدابوست با استفاده از این دسته بند ضعیف آموزش داده خواهد شد. در ادامه مراحل الگوریتم آدابوست بیان شده است.

- عکس های $(x_1, y_1), \dots, (x_n, y_n)$ به عنوان ورودی دریافت می‌شوند. y_i می تواند ۰ یا ۱ باشد که بودن یا نبودن چهره را مشخص می کند.
- وزن‌ها مقدار اولیه داده می‌شوند. $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ برای $y_i = 0, 1$ که m تعداد ۰ ها و l تعداد ۱ ها می‌باشد.
- برای $t = 1, \dots, T$:
  1. وزن ها نرمال می‌شوند. $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
  2. به ازای هر ویژگی j دسته بند h_j که به یک ویژگی محدود است، آموزش داده می‌شود. خطا با توجه به w_t از رابطه‌ی $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$ محاسبه می‌شود.
  3. دسته بند h_t با کم ترین مقدار ϵ_t انتخاب می‌شود.
  4. وزن ها به روزرسانی می‌شوند. $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ که $e_i = 0$ در صورتی که داده ی x_i درست پیش بینی شده باشد و $e_i = 1$ در غیر این صورت. $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
- مدل دسته بندی کننده‌ی نهایی عبارت است از:

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

[1] Classifier



شکل 10 نحوه ی کارکرد دسته بند آبشاری[1]

از هر تصویر تعداد زیادی پنجره ی کوچک تر انتخاب می‌شود که باید چهره یا عدم چهره مشخص شود. تعداد بسیار زیادی از این پنجره ها چهره نخواهند بود. برای این که بتوان به سرعت پنجره های غیر چهره ی واضح را کنار زد از آبشاری از دسته بند ها استفاده می‌شود. به این ترتیب که در دسته بند اول فقط از یک ویژگی برای دسته بندی استفاده می‌شود که تعدادی از تصاویر را غیر چهره تشخیص می‌دهد و کنار می گذارد. سپس تصاویر باقیمانده با دسته بندی با ویژگی‌های بیش تر مورد بررسی قرار می‌گیرند و این فرآیند به همین ترتیب ادامه پیدا می‌کند (شکل ۱۳). تعداد ویژگی‌های انتخاب شده در پنج دسته بند اول به ترتیب ۱، ۱۰، ۲۵، ۲۵ و ۵۰ می‌باشد. ۳۸ دسته بند آموزش داده می‌شود که مجموع ویژگی‌ها در تمام دسته بندها برابر ۶۰۶۱ می‌باشد.

پس از آموزش دسته بند آبشاری و در مرحله‌ی استفاده از آن تصویر توسط پنجره‌هایی در مکان‌ها و اندازه های مختلف پیمایش می‌شود. تمام این پنجره‌ها به مدل دسته بند آبشاری داده می‌شوند تا وجود چهره را در هرکدام از آن‌ها تشخیص بدهد. در انتها از بین پنجره‌هایی که چهره تشخیص داده شده و با هم تداخل دارند یکی انتخاب می‌شود. الگوریتم ویولا-جونز می‌تواند با سرعت بسیار خوبی وجود چهره در تصویر را تشخیص دهد ولی در تشخیص چهره‌هایی که دارای چرخش در زوایای مختلف می‌باشند، دچار مشکل می شود.

[1] Cascade Classifier

در نتیجه ابتدا در تصویر مورد نظر به دنبال استخراج صورت هستیم. در صورت یافتن صورت بر اساس مراحل زیر پیش می‌رویم. اما در صورت عدم وجود صورت در فریم، با استفاده از محدوده‌ی رنگی به دست آمده از مجموعه داده عمل می‌کنیم.

پس از استخراج صورت توسط دسته‌بندی کننده، مختصات کادر دور صورت به عنوان خروجی داده می‌شود. حال می‌توانیم از این ناحیه، مولفه‌های فام و خلوص مرتبط با پوست را استخراج کنیم.

در واقع، به ازای هرکدام از پیکسل‌های موجود در مستطیل تبدیل فضای رنگی به فضای موردنظر انجام می‌شود. حال ما به دنبال بازه‌ای برای پوست صورت هستیم. به دلیل امکان وجود نویز در ناحیه صورت، از میانه نتایج حاصل استفاده شده است. هم‌چنین برای به دست آوردن مقادیر حد بالا و پایین، با استفاده از آزمایشات انجام گرفته، از مقادیر زیر استفاده شده است.

```
Hue_lower = np.median(Hue_face) - 20
Hue_upper = np.median(Hue_face) + 20
Sat_lower = np.median(Saturation_face) - 7
Sat_upper = np.median(Saturation_face) + 7
```

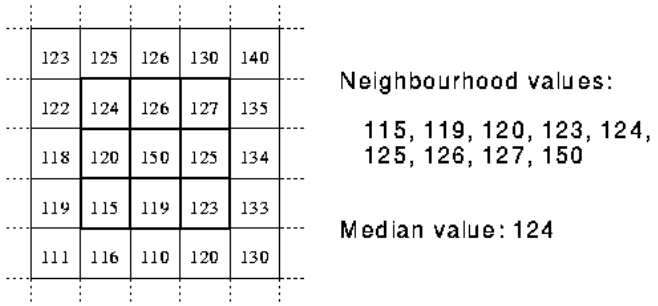
شکل 11 تعیین بازه با استفاده از مقدار میانه مستطیل صورت

پس از استخراج بازه‌ها، همانند روش قبل نقاطی از تصویر که در بازه صدق می‌کنند، به رنگ سفید (عدد ۲۵۵) و نقاطی که در بازه صدق نمی‌کنند، به رنگ سیاه (عدد ۰) تغییر مقدار می‌دهند و تصویر باینری به دست می‌آید.

3.2.3 کاهش نویز

تصویر به دست آمده از مرحله قبل می‌تواند شامل نویز باشد. به این صورت که در میان تعداد زیادی پیکسل با مقدار ۲۵۵ تعداد محدودی و انگشت شماری پیکسل با مقدار ۰ وجود داشته باشد(و یا برعکس). در نتیجه در راستای بهبود کیفیت، از کاهش نویز استفاده می‌شود. روش استفاده شده برای کاهش نویز در این بخش، فیلتر میانه می‌باشد. این فیلتر توسط دستور موجود در کتابخانه OpenCV اعمال شده است.

عدد ورودی فیلتر، اندازه کرنل یا پنجره اعمال فیلتر است. کرنل، مربعی به اندازه عدد ورودی است. مرکز کرنل یا همان مربع بر روی تک تک پیکسل‌های تصویر قرار می‌گیرد. سپس از بین اعدادی که داخل مربع قرار می‌گیرند، میانه انتخاب و جایگزین عنصر وسط می‌شود.



شکل 12 نمونه‌ای از چگونگی عملکرد فیلتر میانه با اندازه کرنل ۳

در صورت وجود نویز بین نقاط تصویر، با عمل میانه‌گیری اثر نویز تا حد خوبی حذف می‌شود. در لبه‌های دست، تعداد نقاط سفیدبر نقاط سیاه غلبه می‌کند. در نتیجه در هنگام اعمال فیلتر میانه، لبه‌ها از بین نرفته و حفظ می‌شوند.

اندازه فیلتر به کار رفته در این پروژه ۱۵ است. که با انجام مقایسه بین نتایج به دست آمده انتخاب شده است.

3.2.4 انتخاب دست به عنوان بزرگترین بخش

با توجه به کارهای انجام شده در بخش قبل، حال ما یک تصویر باینری در دست داریم. در این تصویر، نواحی تشخیص داده شده به عنوان پوست به رنگ سفید و نواحی غیر پوستی به رنگ سیاه هستند. اما هدف ما انتخاب ناحیه پوستی دست از بین نواحی پوست موجود است. از این رو قصد داریم تا از بین ناحیه‌های سفید، دست را انتخاب کنیم. نکته‌ای که باید به آن توجه کنیم، امکان وجود صورت در تصویر است. اما ما مستطیل متعلق به صورت را در مرحله‌های قبل پیدا کرده‌بودیم. در نتیجه ابتدا ناحیه پوستی متعلق به صورت با صفر می‌کنیم.

حال باید در بین نواحی باقی‌مانده به دنبال دست باشیم. برای این کار از تابع connectedComponentsWithStats موجود در کتابخانه OpenCV 3 استفاده می‌کنیم. به این صورت که تصویر موردنظر را به عنوان ورودی به تابع می‌دهیم. این تابع با در نظر داشتن قطعه‌های چسبیده به هم، به هر قسمت یک برچسب اختصاص می‌دهد. برچسب صفر همواره متعلق به پس‌زمینه است. در نتیجه در صورتی که n برچسب داشته باشیم، برچسب نهایی n-1 است. این تابع برای مشخص کردن اتصالات از

نگاه‌کردن به همسایگی‌های هر خانه کمک می‌گیرد. برای این کار، می‌تواند به ۴ همسایه در ۴ جهت نگاه کند و یا در ۸ جهت همسایه‌ها را بررسی کند. به صورت پیش‌فرض این تابع به ۸ همسایگی نگاه می‌کند. این تابع دارای ۴ خروجی است. retval نشان‌دهنده تعداد برچسب‌های موجود پس از تشخیص نواحی متصل به هم است. labels ماتریسی به اندازه تصویر ورودی است. هر عضو آن نشان‌دهنده برچسب متعلق به پیکسل متناظر در تصویر اصلی است. خروجی بعدی stats نشان‌دهنده اطلاعاتی درباره برچسب‌های خروجی است. برای مثال یکی از اطلاعات موجود برچسب پس‌زمینه آن برچسب است. centroid نیز نشان‌دهنده مرکز هر کدام از نواحی متعلق به برچسب یکسان است.

در صورت علاقه می‌توان برچسب‌های حاصل را نمایش داده و درکی از بخش‌های تصویر به دست آورد. اما از آنجایی که تعداد برچسب‌های تصویر، هر عددی می‌تواند باشد، آن را به محدوده ۰ تا ۲۵۵ انتقال می‌دهیم.

در راستای انتخاب بزرگترین بخش در تصویر، قصد داریم برچسب با بیشترین تعداد تکرار را مشخص کنیم. برای این کار، با استفاده از ماتریس برچسب‌های خروجی، تعداد تکرار هرکدام از برچسب‌ها را به دست می‌آوریم و برچسب با بیشترین تعداد تکرار را انتخاب می‌کنیم. البته باید این برچسب حتما چک شود. در صورتی که ۰ باشد یعنی متعلق به پس‌زمینه است. در نتیجه باید برچسب بعدی با بیشترین تعداد تکرار را انتخاب کنیم.

نکته‌ای که وجود دارد عملکرد این بخش در صورت عدم وجود دست است. برخی از نواحی به دلیل شباهت رنگی به ناحیه پوست، به اشتباه به رنگ سفید درآمده‌اند. حال در صورتی که دست در تصویر وجود نداشته باشد، یکی از آن نواحی به عنوان دست انتخاب می‌شود. برای رفع این مشکل، کمینه اندازه‌ای برای دست در نظر گرفته شده است. با این‌کار در صورتی که تعداد تکرار برچسب با بیشترین تکرار، از کمینه موردنظر کمتر باشد، نادیده در نظر گرفته می‌شود.

حال با در دست داشتن برچسب با بیشترین تعداد تکرار که متعلق به ناحیه دست است، هر ناحیه‌ای که دارای این برچسب باشد به رنگ سفید و بقیه نواحی به رنگ سیاه تغییر رنگ داده می‌شوند.

خروجی این مرحله نیز تصویر باینری است. اما در این تصویر، تنها مناطق متعلق به دست دارای رنگ سفید هستند.

3.2.5 پرکردن حفره‌های تصویر حاصل

در تصویر باینری به دست آمده، برخی از نواحی متعلق به دست می‌تواند دارای رنگ سیاه باشد و تشکیل نقاط ریز سیاه و یا حفره‌هایی بدهد. در واقع به دلایلی مثل تغییر نورپردازی، رنگ برخی نقاط تغییر کرده و از محدوده‌ی استخراجی برای رنگ پوست خارج شده است. از این رو تصویر یک دست سفید برای دست حاصل نشده‌است. حال قصد داریم تا این نواحی را نیز پر کنیم. برای این کار از تابع floodFill موجود در کتاب‌خانه OpenCV استفاده می‌کنیم. این تابع جزء داده شده را با رنگ داده شده پر می‌کند. به عنوان ورودی، تصویری را که باید عملیات بر روی آن انجام شود، دریافت می‌کند. علاوه بر تصویر، ماسک[1] نیز یکی از ورودی‌ها است. عملیات بر روی این ماسک انجام می‌شود. باید دقت داشت که ماسک، هم ورودی و هم خروجی است. در نتیجه باید حتما تعریف شود. هم‌چنین باید در تعریف ماسک باید در نظر داشت که با توجه به عملیات داخلی تابع، اندازه ماسک در هر جهت باید ۲ پیکسل بیشتر از تصویر ورودی باشد. تابع floodFill با مواجهه با عناصر غیر صفر در ماسک، بر روی خانه‌های متناظر کاری نمی‌کند. در نتیجه اگر می‌خواهید در لبه‌ها تغییری ایجاد نشود، به جای می‌توان از یک لبه‌یاب[2] استفاده کرد.

حال ما قصد داریم تا دست را با رنگ سفید پرکنیم. برای این کار از حالت برعکس استفاده می‌کنیم. یعنی دست را با سیاه و پس‌زمینه را باسفید پرکرده و سپس برعکس می‌کنیم. دلیل استفاده از این کار دسترسی راحت به پس‌زمینه است. به عنوان نقطه شروع از نقطه (0,0) استفاده می‌کنیم. پس از پر کردن تصویر، حال زمان برعکس کردن رنگ رسیده است. از این رو از عملیات bitwise_not استفاده می‌کنیم. پس از این عملیات با or کردن نتیجه حاصل با تصویری که از قبل داشتیم، حفره‌ها پر می‌شود. نکته‌ای که در هنگام اجرا با آن مواجه شدیم، ظهور اعداد منفی در فرایند Not کردن بود، از این رو به جای اعداد منفی، عدد متناظر مثبت را جایگزین کردیم.

[1] Mask

[2] Edge detector

3.2.6 برش ناحیه مربوط به دست

حال ما تصویری باینری داریم که تنها ناحیه سفید آن، نشان‌دهنده دست است. هدف در این بخش برش تصویر و به دست آوردن مربع پوشش‌دهنده دست است. برای این کار به دنبال نقاط سفید یا همان دارای مقدار ۲۵۵ در تصویر می‌گردیم. کمترین و بیشترین مقدار متعلق به x و y را می‌یابیم. سپس با استفاده از این مقادیر مربعی به دور دست می‌کشیم. دقت داشته باشید از آنجایی که بحث مربع است، از بین تفاضل بیشینه و کمینه x و بیشینه و کمینه y بیشتر مقدار به عنوان ضلع مربع انتخاب می‌شود. هم‌چنین مربع را به نحوی می‌کشیم که دست در وسط تصویر قرار بگیرد.

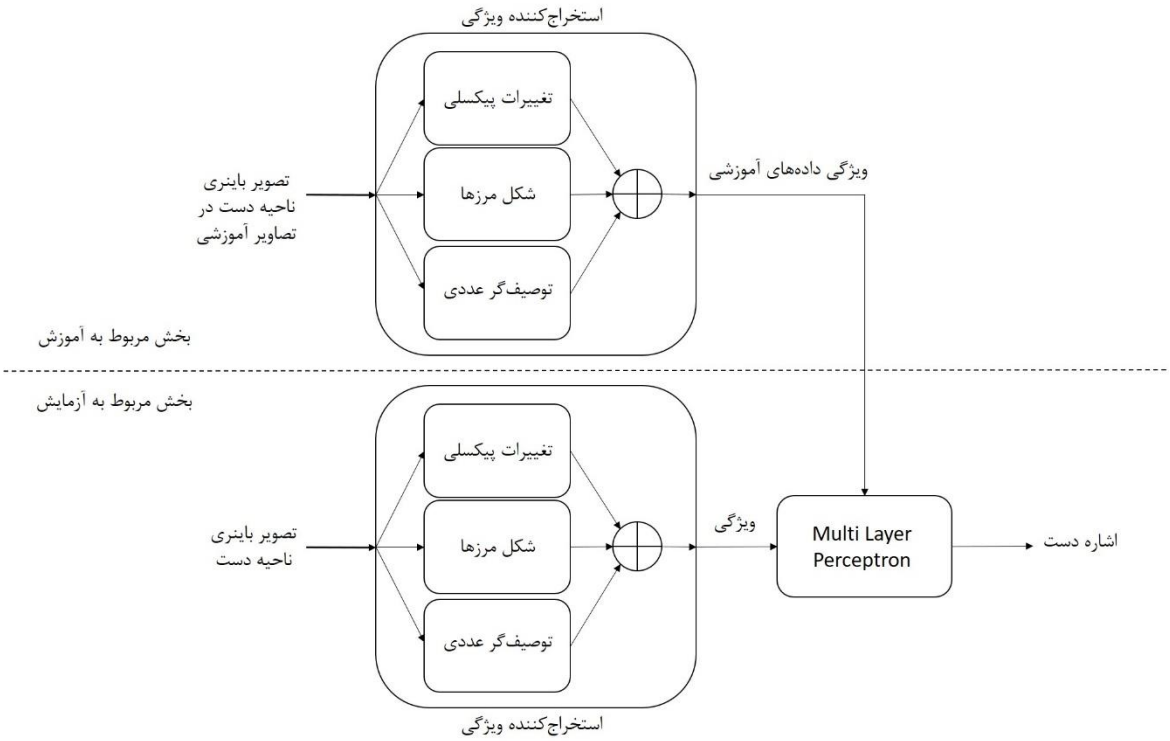
پس از مشخص کردن ابعاد و مختصات، آن ناحیه را برش داده و در متغیر دیگری ذخیره می‌کنیم.

3.3 تشخیص اشاره با استفاده از شبکه عصبی

در بخش قبل ناحیه مربوط به دست شناسایی و با رنگ سفید مشخص و بقیه نواحی با رنگ سیاه مشخص شدند. سپس با برش مربعی ناحیه دست از تصویر استخراج گردید. در نتیجه کادر مربعی به دور ناحیه دست در تصویر باینری به عنوان ورودی در این مرحله وجود دارد. حال قصد داریم تا با استفاده از شبکه‌های عصبی اشاره متناظر با ورودی را به دست آوریم. برای این‌کار از دو نوع شبکه عصبی، ساده و عمیق استفاده می‌کنیم. در بخش‌های جداگانه بلوک دیاگرام مربوط به هر نوع شبکه توضیح داده می‌شود.

3.3.1 استفاده از شبکه عصبی ساده

بلوک دیاگرام روش به کار رفته در این بخش برای تشخیص اشاره در شکل زیر نمایش داده شده است.



شکل 13 بلوک دیاگرام بخش تشخیص اشاره دست با شبکه عصبی ساده

شبکه عصبی در نظر گرفته شده، پرسپترون چندلایه[1] است. این شبکه با سه لایه برای این کاربرد در نظر گرفته شده‌است. لایه اول، مربوط به ورودی‌های برنامه است، که بردار ویژگی‌ها به آن داده می‌شود. در نتیجه تعداد نورون‌های لایه ورودی برابر با اندازه بردار ورودی است. لایه آخر نیز مربوط به خروجی‌های برنامه است، که دارای تعداد یک نورون است. خروجی این نورون شماره اشاره است. لایه میانی، لایه مخفی است. تعداد نورون‌های این لایه توسط صحیح و خطا[2] مشخص می‌شود. نرخ یادگیری برابر با ۰٫۰۱ در نظر گرفته شده‌است. هم‌چنین تغییرات نرخ یادگیری به صورت adaptive در نظر گرفته شده است. در این شرایط با کاهش خطا، نرخ یادگیری نیز کاهش می‌یابد. حداکثر تعداد تکرار نیز برابر با ۱۰۰۰۰۰ در نظر گرفته شده‌است. برای بهینه‌سازی وزن‌ها از گرادیان نزولی تصادفی[3] استفاده شده‌است. تابع فعالیت در نظر گرفته شده، تابع فعالیت tansig است که به صورت زیر تعریف می‌شود.

[1] Multi-Layer Perceptron (MLP)

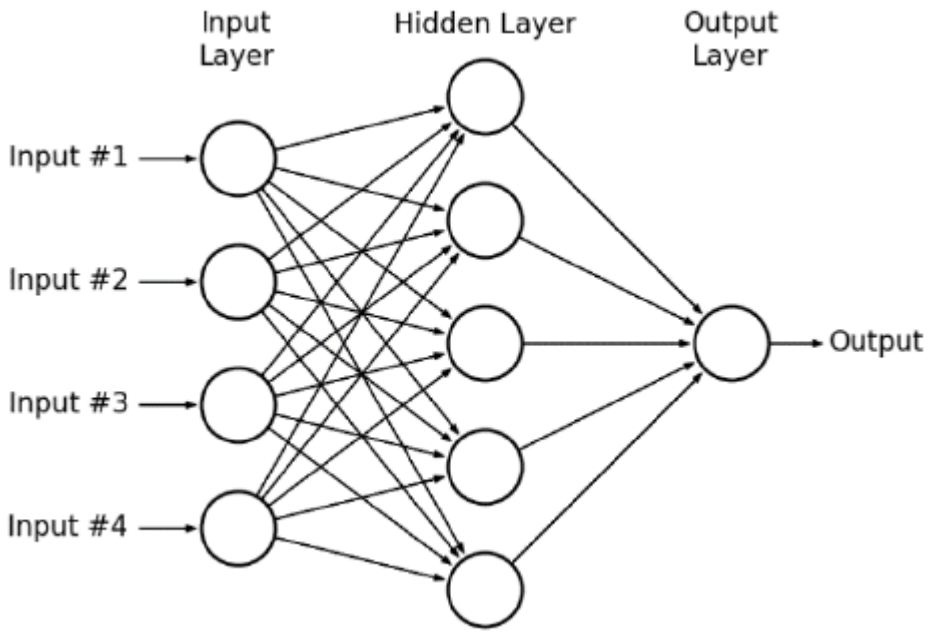
[2] Trial and Error

[3] Stochastic gradient descent (SGD)

$$tansig(n) = \frac{2}{1 + e^{-2n}} - 1$$

(3)

در شکل زیر حالت کلی شبکه عصبی چندلایه به کار رفته نمایش داده شده است.



شکل 14 شبکه عصبی چند لایه

بردار ورودی در شبکه عصبی چندلایه، بردار ویژگی مربوط به تصویر دست ورودی است. در نتیجه ابتدا از تصویر ورودی، بردار ویژگی استخراج می‌شود. سپس این بردار به عنوان ورودی به شبکه عصبی ارسال شده و خروجی یا همان شماره اشاره متناظر تولید می‌شود. شبکه عصبی باید از قبل توسط ویژگی‌های استخراج شده از داده‌های آموزشی، آموزش داده شده باشد. سپس با استخراج بردار ویژگی از داده آزمایشی و ارسال آن به شبکه عصبی، جواب که همان اشاره دست است، دریافت می‌شود.

قبل شروع بررسی فرایند آموزش و آزمایش شبکه عصبی، به معرفی بردار ویژگی موردنظر و نحوه استخراج آن می‌پردازیم.

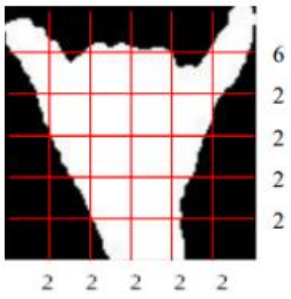
3.3.1.1 بردار ویژگی

سه بردار ویژگی برای استخراج در نظر گرفته شده است، که با چسباندن این سه بردار به هم بردار ویژگی نهایی حاصل می‌شود.

1) تغییرات پیکسل‌ها در تصویر

برای اندازه‌گیری تغییرات پیکسل‌ها در تصویر، برش‌های متقاطعی[1] را بر روی تصویر رسم می‌کنیم. این برش‌های متقاطع در هر دو جهت افقی و عمودی رسم می‌شوند. سپس روی خطوط این برش‌ها، تعداد تغییرات از ۰ به ۲۵۵ یا ۲۵۵ به ۰ شمرده می‌شود. در واقع در حال شمارش تعداد تغییرات از پس‌زمینه به پیش‌زمینه[2] و برعکس هستیم. در نتیجه زمانی که ۱۰ برش متقاطع داریم (۵ برش در هر کدام از جهات)، ۱۰ عدد به عنوان تغییرات به دست می‌آید. برای انتخاب تعداد برش‌ها باید توجه داشت که با افزایش تعداد برش‌ها، اطلاعات بیشتری به دست می‌آوریم. در نتیجه بردار ویژگی به دست آمده، طولانی‌تر و جزئیات حمل شده نیز بیشتر می‌شود. اما از سمتی نیز محاسبات پیچیده‌تر شده و نیاز به فضای ذخیره‌سازی بیشتری داریم. در نتیجه برای انتخاب تعداد برش‌های متقاطع، باید با در نظر داشتن اطلاعاتی که می‌خواهیم و هزینه‌ای که قصد داریم بدهیم، تصمیم بگیریم.

همان‌طور که گفتیم و در شکل زیر نیز می‌بینیم، ۱۰ عدد توسط ۱۰ برش متقاطع به دست آمده و تشکیل برداری به طول ۱۰ می‌دهند.



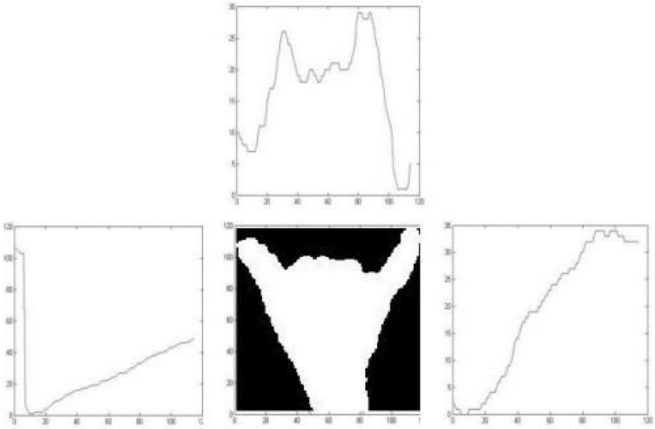
شکل 15 تصویر اشاره دست با ۱۰ برش متقاطع و اعداد به دست آمده

2) شکل مرزها

[1] Cross Section

[2] foreground

این ویژگی به بررسی فاصله لبه‌های دست از مرز بیرونی تصویر در جهت خواسته شده، می‌پردازد. در سه جهت بالا، راست و چپ این مقدار محاسبه می‌شود. پس از محاسبه این مقدار برای هرکدام از جهات، برداری به بزرگی ضلع تصویر به دست می‌آید. اندازه این بردار بزرگ است و قصد داریم تا آن را کوچک‌تر کنیم. در این راستا، با انتخاب عدد n بردار به دست آمده را به n بخش تقسیم می‌کنیم و در هربخش عملیات میانگین‌گیری انجام می‌دهیم. در نتیجه برای هرکدام از جهت‌ها برداری به اندازه n به دست می‌آید.



شکل 16 شکل مرزها - تصویر سمت چپ شکل مرز نسبت به لبه سمت چپ تصویر، تصویر بالا شکل مرز نسبت به لبه بالای تصویر و تصویر سمت راست شکل مرز دست نسبت به لبه سمت راست تصویر است.

با کنار هم قرار دادن سه بردار به دست آمده در سه جهت بردار نهایی در این بخش به دست می‌آید. در هرکدام از جهات برداری به اندازه ۱۰ تولید می‌شود. در نهایت برداری به اندازه ۳۰ در این بخش تولید می‌شود.

3) توصیف‌گر عددی

این توصیف‌گر از ترکیب دو بخش به دست می‌آید.

- نسبت بین پهنا (w) و ارتفاع (h) دست

$$Edges\ ratio = \frac{w}{h} \quad (4)$$

- نسبت بین محدوده‌ی اختصاصی به دست و کل تصویر

$$Area\ ratio = \frac{\sum pixel_{foreground}}{w * h} \quad (5)$$

با ترکیب دو مقدار بالا بردار این ویژگی به اندازه ۲ به دست می‌آید.

3.3.1.2 آموزش شبکه توسط استخراج بردار ویژگی موردنظر از تصاویر آموزشی

برای آموزش شبکه، نیاز به تعدادی داده آموزش داریم. این داده‌ها شامل تصاویری از دست با اشاره‌های موردنظر است. برای تبدیل داده‌ها به فرمت موردنظر از فایل gatherTrain.py و یا متود gatherTrainingData در فایل networkFeature.py استفاده کرد. این تابع ابتدا عکس‌های رنگی را که به عنوان ورودی داده شده‌است، می‌خواند. بعد از خواندن عکس‌ها، تابع extractFeatures() برای استخراج ویژگی‌ها فراخوانی می‌شود. خروجی این تابع ماتریسی از ویژگی‌ها و خروجی است. ماتریس ویژگی‌ها ماتریسی به تعداد سطرهای عکس‌های موجود و تعداد ستون‌های بردار ویژگی موجود است. خروجی‌ها به تعداد عکس‌های ورودی هستند. در هر سطر ماتریس ویژگی‌ها، بردار ویژگی یک عکس وجود دارد. در خروجی نیز برچسب متناظر با عکس یا همان اشاره موردنظر وجود دارد.

حال وارد جزییات این تابع می‌شویم. در این تابع تصاویر از آدرس داده شده خوانده می‌شوند. اسم عکس‌ها متناسب با اشاره در نظر گرفته شده‌اند. ابتدا از روی اسم عکس، برچسب یا خروجی مشخص می‌شود.

در ادامه نیز بردار ویژگی برای هر تصویر به شیوه گفته‌شده در بخش قبل، استخراج می‌شود و در سطر متناظر در ماتریس ویژگی‌ها ریخته می‌شود. سپس این دو به عنوان خروجی مشخص می‌شوند.

حال زمان آموزش شبکه فرا می‌رسد. شبکه با ساختار توصیف شده، تعریف می‌شود. پس از تعریف شبکه، زمان ارسال داده‌های یادگیری (ویژگی‌های متناظر داده‌های یادگیری) و برچسب‌های متناظر با داده‌های یادگیری، فرا می‌رسد. پس از آموزش شبکه با داده‌های ارایه‌شده، مدل به دست آمده را ذخیره می‌کنیم.

برای بررسی کارایی مدل از داده‌های آزمایش فراهم شده، استفاده می‌کنیم. به این صورت که این داده‌ها توسط تابعی مشابه با بخش قبل، به اسم gatherTest.py یا gatherTestingData در networkFeatures.py آماده‌سازی می‌شود. پس از آماده‌سازی ماتریس آزمایش و برچسب‌های متناظر با آن، داده‌های به عنوان داده‌های آزمایش به مدل داده می‌شود و دقت اندازه‌گیری می‌شود.

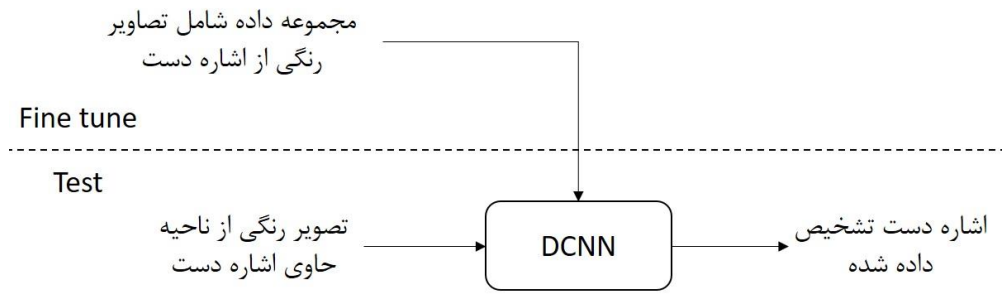
3.3.1.3 استخراج بردار ویژگی از تصویر دست موجود و پیش‌بینی اشاره توسط شبکه‌عصبی

حال زمان مشخص کردن برچسب متعلق به اشاره‌های تشخیص داده‌شده در مرحله استخراج دست از فریم است. همان‌طور که در بخش 3.2 دیدیم، نتیجه حاصل از کل مراحل، تصویر باینری شامل دست به رنگ سفید و پس‌زمینه سیاه بود. هم‌چنین نحوه استخراج بردارهای ویژگی مربوط به داده‌های آموزش را نیز دیدیم. ابتدا باید مدل مربوط به شبکه‌ی عصبی که در مرحله قبل ذخیره شده، بارگذاری شود. سپس بردار ویژگی به دست آمده به مدل شبکه عصبی داده می‌شود و برچسب موردنظر به عنوان خروجی دریافت می‌شود.

3.3.2 استفاده از شبکه‌های عصبی عمیق

یکی از راه‌حل‌های پیشنهادی برای دسته‌بندی تصویر دست حاصل، استفاده از شبکه‌های عصبی عمیق است. این شبکه‌ها در بسیاری از کاربردها، عملکرد بهتری ارایه می‌دهند. اما از طرفی برای رسیدن به عملکر بهتر، نیاز به داده‌های آموزشی بیشتری است. که این عامل در بسیاری از موارد محدود کننده است. در نتیجه می‌توان از شبکه‌های عصبی عمیق از پیش آموزش‌داده‌شده، استفاده کرد. به این صورت که تنها به تنظیم مجدد وزن‌های لایه‌های خاصی از این شبکه‌ها پرداخت.

در مقاله [9] بر روی شبکه‌ی عصبی عمیق با ۲۲ لایه که از قبل آموزش دیده‌است، توسط یک میلیون داده دست، تنظیم مجدد انجام شده است. ما نیز از قصد داریم تا با تنظیم مجدد مدل این مقاله با استفاده از مجموعه داده دست خود عمل دسته‌بندی را انجام دهیم. در زیر بلوک دیاگرام مربوط به این بخش را مشاهده می‌کنیم.



شکل 17 بلوک دیاگرام مربوط به تشخیص اشاره دست با استفاده شبکه‌های عصبی عمیق

3.3.2.1 تنظیم شبکه با استفاده از داده‌های آموزش

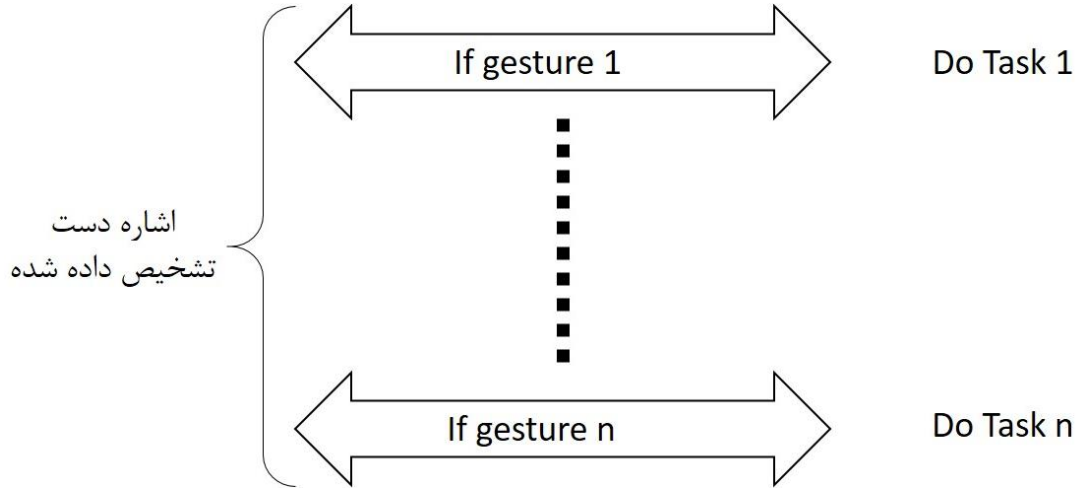
همان‌طور که در بخش قبل توضیح داده‌شد، یک شبکه عمیق آموزش داده‌شده توسط یک میلیون داده دست، در اختیار داریم. حال ابتدا باید توسط داده‌های خود به تنظیم مجدد بپردازیم. برای تنظیم مجدد لایه آخر شبکه عصبی بر اساس نیازهای پروژه تغییر داده می‌شود. در واقع لایه آخر بر اساس تعداداشاره‌های موجود برای دسته‌بندی با نام‌های جدید مجددا تغییر می‌یابد. پس از اعمال این تغییرات شبکه با داده‌های جدید ورودی تنظیم مجدد می‌شود. در شبکه‌های عصبی عمیق داده‌های ورودی به صورت رنگی در نظر گرفته می‌شود. از این رو بخشی از پیش‌پردازش‌های انجام شده برای تهیه ورودی مناسب برای شبکه‌های عصبی ساده، در این بخش نیاز نیست.

3.3.2.2 ارایه تصویر دست استخراج شده در مراحل استخراج دست و دریافت اشاره موردنظر

در هر فریم پس از تشخیص ناحیه مربوط به دست، برش رنگی از آن انجام داده و به عنوان ورودی به شبکه عصبی عمیق می‌دهیم. خروجی این شبکه اشاره دست تشخیص داده شده است.

3.4 انجام کار متناظر با اشاره تشخیص داده شده

پس از انجام مراحل قبل، حال اشاره مشخص شده‌است. بر اساس شماره اشاره به دست آمده، عمل متناظر انجام می‌شود.



شکل 18 انجام کار متناظر با اشاره

3.5 ابزارها

3.5.1 کتابخانه OpenCV[1]

کتابخانه OpenCV یک کتابخانه متن باز برای انجام کارهای بینایی ماشین و یادگیری ماشین است. این کتابخانه به منظور فراهم ساختن زیرساخت مناسب برای انجام کارهای مربوط به بینایی ماشین به صورت بهینه و سریع، آماده شده است. این کتابخانه شامل بیش از 2500 الگوریتم بهینه و زمان واقعی، شامل روش‌های پایه و روش‌های حال حاضر علم دانش است. الگوریتم‌های موجود در این کتابخانه دارای قابلیت‌های همچون تشخیص صورت، تشخیص انسان، دسته‌بندی کارصورت گرفته توسط انسان در ویدیو، پیگیری حرکات دوربین، پیگیری حرکات اشیا و ... است. این کتابخانه دارای واسط‌هایی به زبان‌های C، C++، Python،Matlab و Java است. هم‌چنین از سیستم‌عامل‌های Windows، Linux، Android و Mac OS پشتیبانی می‌کند.

برای این پروژه از نسخه 3.2 تحت سیستم عامل لینوکس و با زبان برنامه‌نویسی پایتون استفاده شده است.

[1] Open Source Computer Vision Library

4

فصل چهارم

جمع‌بندی و نتیجه‌گیری

جمع‌بندی و نتیجه‌گیری

در فصل قبل به بررسی روش پیاده‌سازی شده پرداختیم. حال در این فصل ابتدا به بررسی نتایج به دست آمده پرداخته و در نهایت جمع‌بندی و نتیجه‌گیری خواهیم داشت.

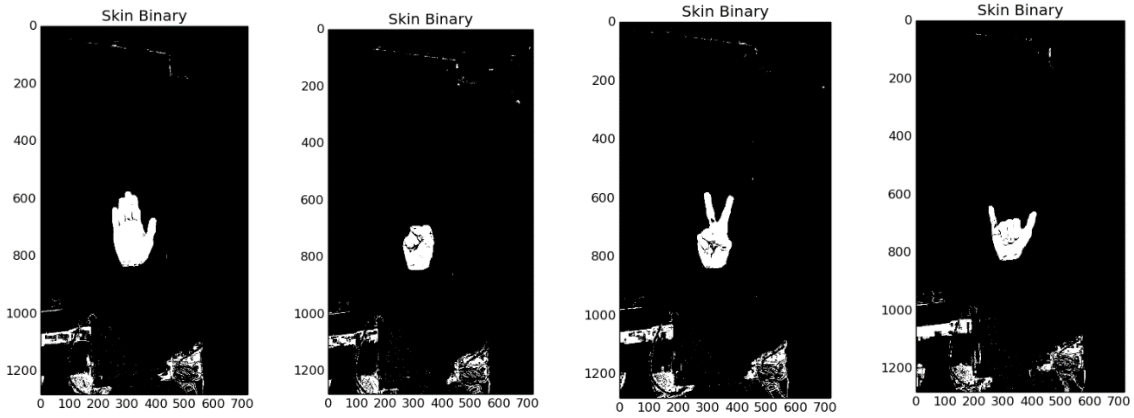
4.1 نتایج به دست آمده

4.1.1 نتیاج ارزیابی بر روی ویدیوهای تهیه شده

برای ارزیابی سیستم، نتایج به دست‌آمده بر روی ویدیوهای انتخاب ضبط شده مورد بررسی قرار گرفتند.

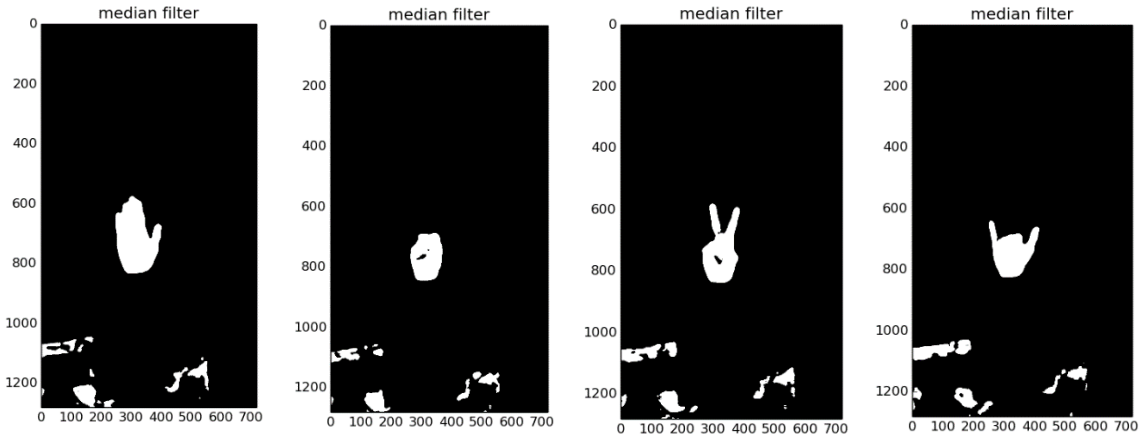
ویدوی اول شامل فرد بدون حضور صورت است.

همان‌طور که در فصل قبل مورد بررسی قرار گرفت، پس از تبدیل فضای رنگی به فضای رنگی HSV ، با استفاده از محدوده به دست آمده توسط مجموعه داده قطعه‌بندی صورت، تصویر باینری به دست می‌آید. در زیر تصاویر باینری به دست آمده برای چهار اشاره دست موجود در ویدیو، نمایش داده شده است.



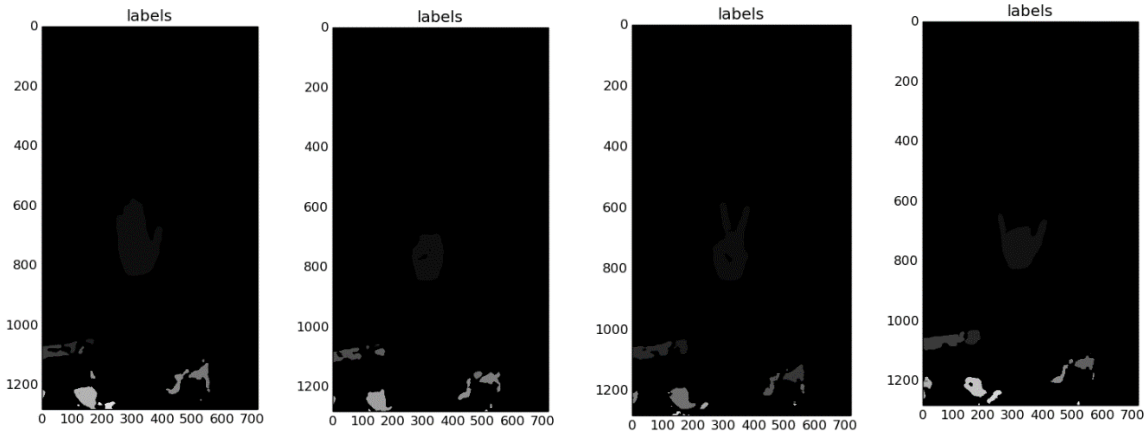
شکل 19 تصاویر باینری به دست آمده برای اشاره‌های دست موجود در ویدیوی بدون حضور صورت

پس از تهیه تصویر باینری، زمان کاهش نویز با استفاده از فیلتر میانه است. همان‌طور که گفتیم این فیلتر به خوبی نویزها را حذف کرده و در اکثر مواقع لبه‌ها را نیز تا حد خوبی حفظ می‌کند. نتیجه حاصل از اعمال فیلتر میانه بر روی تصاویر باینری نمایش داده شده در بخش قبل، در زیر نمایش داده شده است.



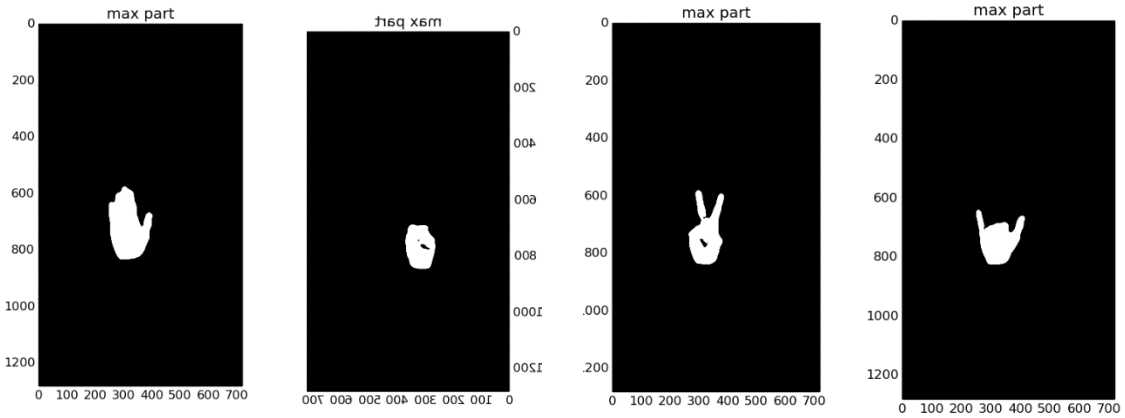
شکل 20 اعمال فیلتر میانه بر روی تصاویر باینری به دست آمده در مرحله قبل

حال پس از اعمال فیلتر میانه، قصد داریم تا بزرگترین بخش موجود در تصویر را توسط روش‌های گفته شده، بیابیم. برچسب‌های انتسابی به هر بخش را در زیر می‌بینیم.



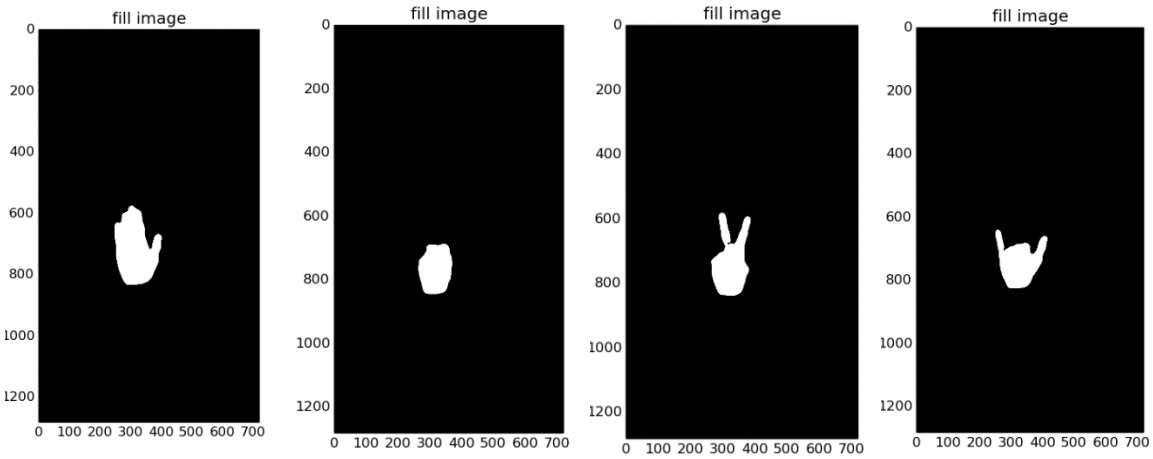
شکل 21برچسب‌های انتسابی به بخش‌های مختلف تصویر

نتیجه را پس از صفر کردن نواحی دارای برچسب‌های با تعداد کمتر را، در زیر می‌بینیم.



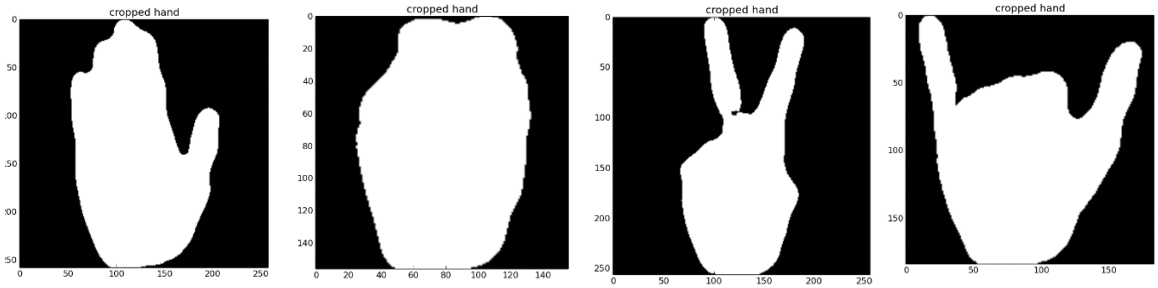
شکل 22 نمایش بزرگترین جزء تصویر

پس از یافتن بزرگترین بخش، باید به پر کردن حفره‌های موجود پرداخت.



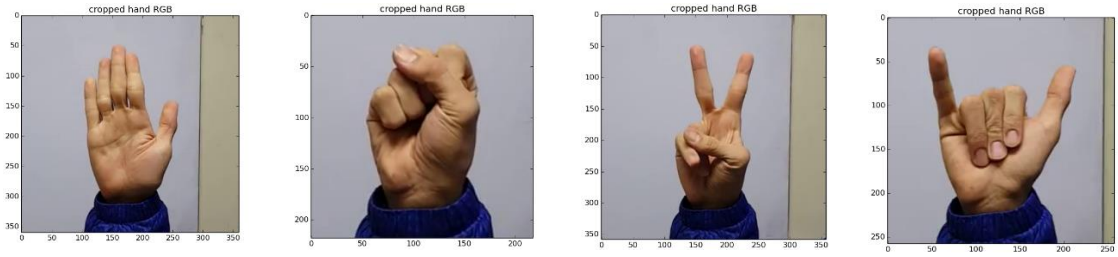
شکل 23 نتیجه حاصل از پر کردن حفره ها

در نهایت باید ناحیه مربوط به دست برش داده شود.



شکل 24 نتیجه حاصل از برش ناحیه مربوط به دست

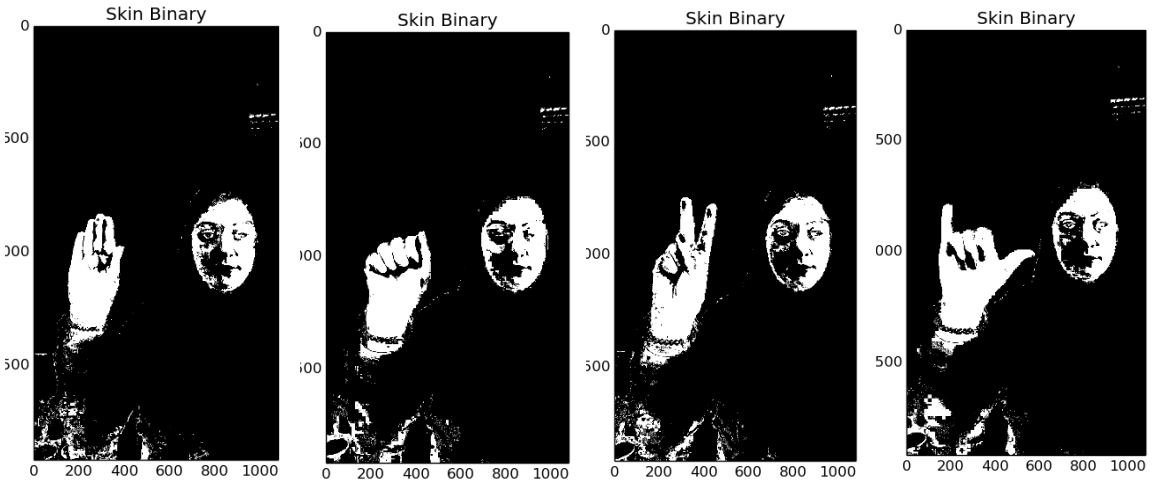
حال تصویر رنگی دست را نیز به برش می‌دهیم. این تصویر برای شبکه عصبی عمیق موردنیاز است.



شکل 25 تصویر رنگی برش داده شده برای دست

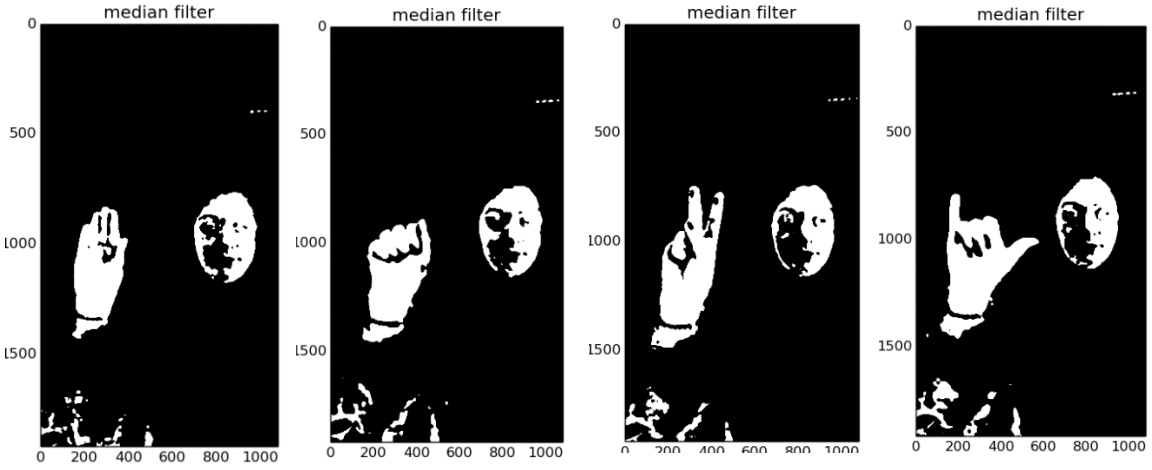
حال به بررسی نتایج حاصل از ویدیوی دوم که در حضور صورت انجام شده است می‌پردازیم.

مجددا پس از تبدیل فضای رنگی به فضای رنگی HSV ، با استفاده از محدوده به دست آمده توسط مجموعه داده قطعه‌بندی صورت، تصویر باینری به دست می‌آید. در زیر تصاویر باینری به دست آمده برای چهار اشاره دست موجود در ویدیو، نمایش داده شده است.



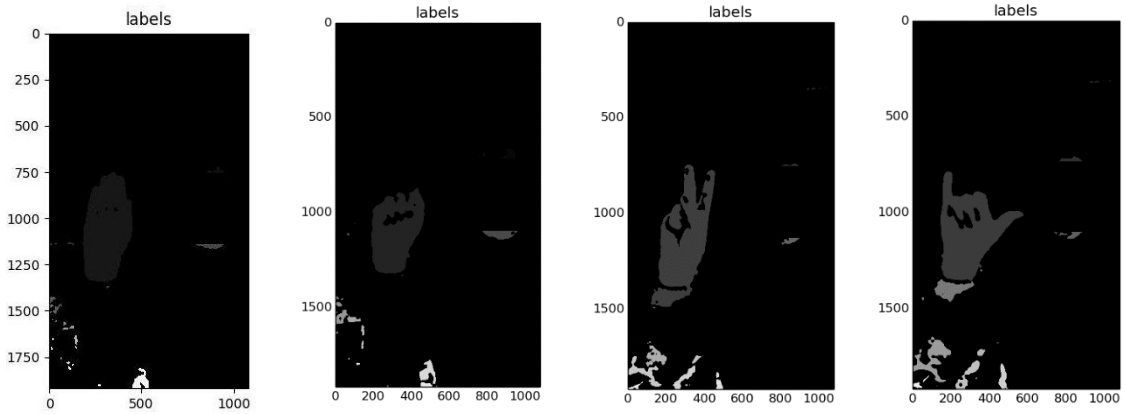
شکل 26 تصاویر باینری به دست آمده برای اشاره‌های دست موجود در ویدیوی در حضور صورت

پس از تهیه تصویر باینری، زمان کاهش نویز با استفاده از فیلتر میانه است. همان‌طور که گفتیم این فیلتر به خوبی نویزها را حذف کرده و در اکثر مواقع لبه‌ها را نیز تا حد خوبی حفظ می‌کند. نتیجه حاصل از اعمال فیلتر میانه بر روی تصاویر باینری نمایش داده شده در بخش قبل، در زیر نمایش داده شده است.



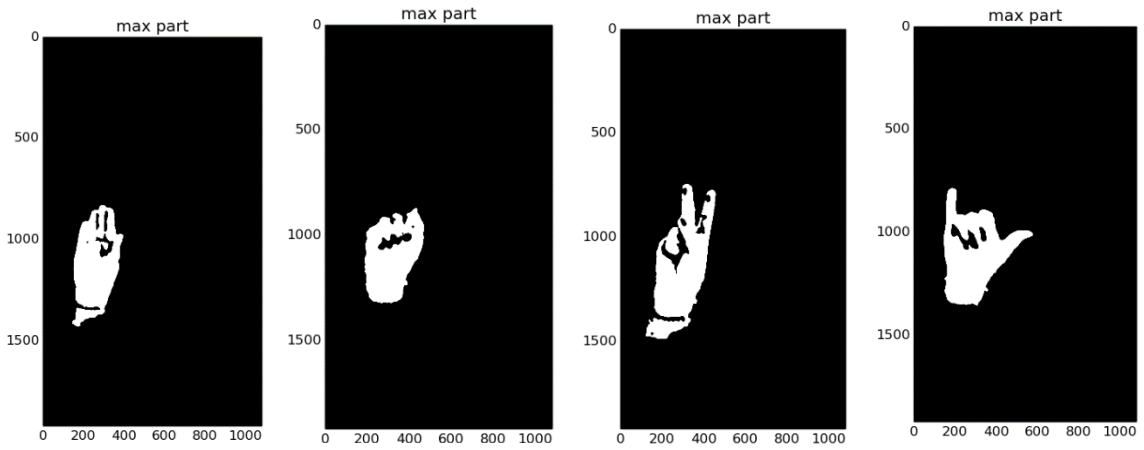
شکل 27 اعمال فیلتر میانه بر روی تصاویر باینری به دست آمده در مرحله قبل

حال پس از اعمال فیلتر میانه، قصد داریم تا بزرگترین بخش موجود در تصویر را توسط روش‌های گفته شده، بیابیم. البته باید توجه کرد صورت را در این مرحله صفر می‌کنیم. در صورت دقت می‌توان مسطیل سیاه متعلق به صورت را در تصویر مشاهده کرد.



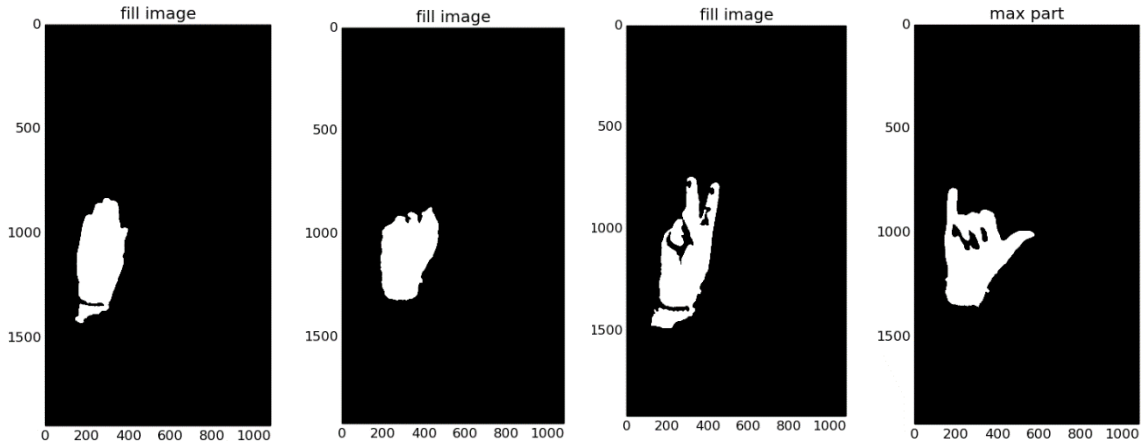
شکل 28 برچسب‌های انتسابی به تصویر

نتیجه را پس از صفر کردن بقیه نواحی، در زیر می‌بینیم.



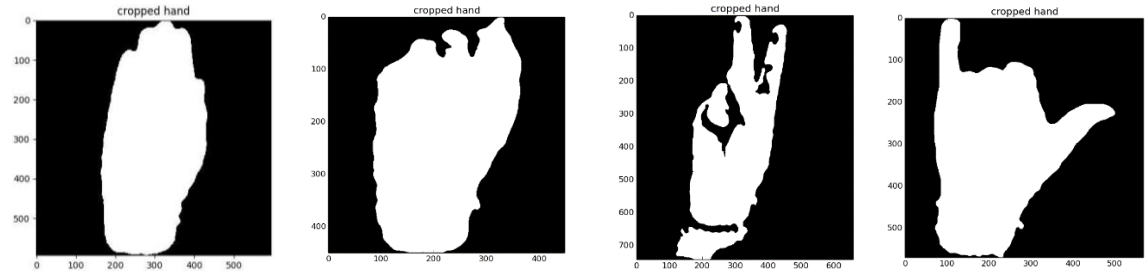
شکل 29 نمایش بزرگترین جزء تصویر

پس از یافتن بزرگترین بخش، باید به پر کردن حفره‌های موجود پرداخت.



شکل 30 نتیجه حاصل از پر کردن حفره ها

در نهایت باید ناحیه مربوط به دست برش داده شود.



شکل 31 نتیجه حاصل از برش ناحیه مربوط به دست

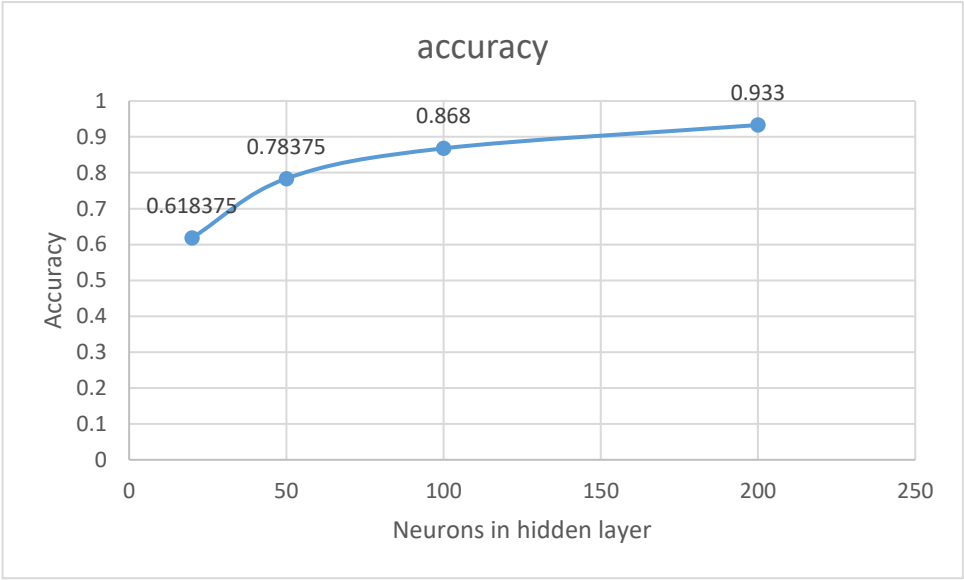
حال تصویر رنگی دست را نیز به برش می‌دهیم. این تصویر برای شبکه عصبی عمیق موردنیاز است.

شکل 32 تصویر رنگی برش داده شده برای دست

هر ستون از تصاویر نشان‌داده شده در بالا، سیر نمایش خروجی در یک فریم از ویدیو می‌باشد.

4.1.2 بررسی بخش تشخیص اشاره

همان‌طور که در بخش روش پیشنهادی اشاره کردیم، تعداد نورون‌های لایه مخفی توسط روش صحیص و خطا به دست می‌آید. حال در زیر به بررسی روند تغییر زمان و دقت با افزایش تعداد تورون‌های لایه مخفی می‌پردازیم.



شکل 33 بررسی روند تغییر دقت با افزایش و کاهش نورون‌های لایه مخفی

همان‌طور که می‌بینیم با افزایش تعداد نورون‌های لایه مخفی، دقت حاصل نیز افزایش می‌یابد. البته این تغییرت دقت در ابتدا چشم‌گیر و زیاد و در ادامه با شیب کمتری در حال افزایش است. از این رو با در نظر داشتن هزینه محاسبانی پرداختی و دقت موردنظر باید تعداد نورون‌های لایه مخفی را مشخص کرد. تعداد به کار رفته در پروژه ما ۲۰۰ عدد نورون در لایه مخفی می‌باشد.

در زیر به بررسی دو نوع شبکه عصبی به کار رفته می‌پردازیم.

جدول 1 بررسی دقت دو روش به کار رفته در پروژه برای شناسایی اشاره دست.

روش	دقت
شبکه عصبی پرسپترون چند لایه	0.9330
شبکه عصبی عمیق	1

البته به طور معمول در اکثر موارد دقت حاصل از شبکه عصبی عمیق بسیار خوب می‌باشد. اما از آنجایی که در هردو روش تعداد داده‌های اموزش کم می‌باشد، شبکه چند لایه‌ای پرسپترون توانسته است دقت بهتری از شبکه عمیق ارایه دهد.

اما نکته‌ی موجود استخراج دستی ویژگی‌های سطح پایین در شبکه‌های ساده و استخراج خودکار ویژگی‌ها در شبکه‌های عمیق است.

4.2 جمع‌بندی و کارهای آینده

همان‌طور که دیدیم، هدف این پروژه طراحی و پیاده‌سازی نرم‌افزاری برای برقراری ارتباط انسان و کامپیوتر توسط اشاره دست و بدون تماس می‌باشد. همان‌طور که می‌دانیم امروزه کامپیوترها از اجزای جدا نشدنی زندگی روزمره ما هستند. در این بین تلفن همراه جزو پر استفاده‌ترین وسایل ارتباطی امروزه است و هر روزه ساعت زیادی صرف کار با تلفن همراه می‌شود. در این پروژه اشاره دست به منظور راه ارتباطی با

کامپیوتر و به ویژه تلفن همراه در نظر گرفته شده است. با استفاده از ابزارهای پردازش تصویر، کتابخانه‌ها و توابع موجود سعی بر آن است تا ویدئویی به صورت برخط از دوربین و یا از قبل ذخیره شده دریافت شود و به فریم‌های تشکیل دهنده خود تقسیم شود و این فریم‌ها به عنوان تصاویر پایه مورد پردازش و تغییر قرار گیرند. پس از اعمال پردازش‌هایی ناحیه مربوط به دست از تصویر کلی استخراج می‌شود. حال از سمت دیگر شبکه‌ی عصبی با ویژگی‌های مجموعه داده دست آموزش داده شده است و با دادن ناحیه دست استخراج شده به عنوان، نمونه آزمایشی، نتیجه یا همان اشاره مورد نظر توسط شبکه عصبی به دست می‌آید. شبکه‌های عصبی به کار رفته، شبکه پرسپترون سه لایه و شبکه عمیق بر پایه گوگل‌نت[1] است. در نهایت عملی متناظر با اشاره انتسابی انجام می‌شود.

ساختار کلی پیاده‌سازی نرم‌افزار را می‌توان به صورت زیر در نظر گرفت:

- دریافت ویدئوی ورودی و تبدیل آن به فریم
- استخراج دست از فریم

الف) استخراج مولفه‌های مربوط به فام[2] و خلوص[3] رنگ از فریم

ب) مشخص کردن بازه‌ی متعلق به پوست و استخراج محدوده پوست با استفاده از بازه به دست آمده

  - استخراج بازه متعلق به پوست با استفاده از داده‌های موجود در مجموعه داده‌ی پوست
  - استخراج بازه متعلق به پوست با استفاده از صورت

ج) کاهش نویز

د) انتخاب دست به عنوان بزرگترین بخش

ه) پرکردن حفره‌های تصویر حاصل

و) برش ناحیه مربوط به دست

---

[1] Google Net

[2] Hue

[3] Saturation

- تشخیص اشاره توسط شبکه عصبی

الف) استفاده از شبکه‌های عصبی عمیق

- تنظیم[1] شبکه با استفاده از داده‌های آموزشی
- ارایه تصویر دست استخراج شده در مراحل قبل‌تر و دریافت اشاره متناظر

ب) استفاده از شبکه‌های عصبی ساده‌تر

- آموزش شبکه توسط استخراج بردار ویژگی موردنظر از تصاویر آموزشی
- استخراج بردار ویژگی از تصویر دست موجود و پیش‌بینی اشاره توسط شبکه‌عصبی

- انجام کار متناظر با اشاره تشخیص داده شده

4.2.1 کارهای آینده

به عنوان کارهایی که می‌توان در ادامه این پروژه انجام داد، می‌توان به موارد زیر اشاره کرد:

- افزودن کلاس متعلق به اشاره‌های دست ناشناس

با افزودن این بخش شبکه با دریافت اشاره ناموجود، آن را به یکی از دسته‌های اشاره موجود اختصاص نداده، بلکه برچسب ناشناخته برای آن اشاره در نظر می‌گیرد.

- استفاده از مدل‌های احتمالاتی برای تشخیص پوست

در این روش‌ها همان‌طور که در فصل دوم توضیح داده‌شد، بر پایه روش‌های احتمالاتی عملیات انتساب احتمال انجام می‌پذیرد.

- در نظر گرفتن دنباله‌ای از اشارات برای انتساب به یک حرکت

در واقع گویا در حال بررسی حرکت دست هستیم و بر اساس حرکت صورت گرفته، تعامل انجام می‌شود. می‌توان در هرکدام از فریم‌های اشاره‌های موجود را استخراج کرده و سپس با استفاده از روش‌های موجود برای مدل کردن امتداد زمانی این اشارات، حرکت را در نظر گرفت.

[1] Fine-tune

5 منابع و مراجع

[1] H. Haitham and K. S.Abdul, "Human Computer Interaction for Vision Based Hand Gesture Recognition : A Survey," in *International Conference on Advanced Computer Science Applications and Technologies*, Kuala Lumpur, 2012.

[2] A. Królak, "Use of Haar-like Features in Vision-Based Human-Computer Interaction Systems," in *New Trends in Audio & Video and Signal Processing: Algorithms, Architectures, Arrangements, and Applications (NTAV/SPA)*, Lodz, 2012.

[3] V. Bhame, R. Sreemathy and H. Dhumal, "Vision based hand gesture recognition using eccentric approach for human computer interaction," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, New Delhi, India, 2014.

[4] J. S. Sonkusare, N. B. Chopade, R. Sor and S. Tade, "A Review on Hand Gesture Recognition System," in *International Conference on Computing Communication Control and Automati*, 2015.

[5] V. S., K. L.R. and J. S. J., "Vision Based Gesturally Controllable Human Computer Interaction System," in *International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Chennai, 2015.

[6] C. Dhule and T. Nagrare, "Computer Vision Based Human-Computer Interaction Using Color Detection Techniques," in *Fourth International Conference on Communication Systems and Network Technologies*, 2014.

[7] A. Agrawal, R. Raj and S. Porwal, "Vision-based Multimodal Human-Computer Interaction using Hand and Head Gestures," in *IEEE Conference on Information and Communication Technologies*, 2013.

[8] V. P. and J. M., "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.

[9] O. Koller, H. Ney and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Amirkabir University of Technology
(Tehran Polytechnic)

Computer Engineering Department

Bsc Thesis

Title

Title of Thesis

By

Mina Ghadimi Atigh

Supervisor

Dr. Rahmati

Aban & 1395