دانشکده مهندسی کامپیوتر و فن‌آوری اطلاعات
دانشگاه صنعتی امیرکبیر

گزارش پروژه دسته‌بندی تصاویر ایستای حالات دست

استاد درس:

دکتر صفابخش

نام دانشجو:

احمد اسدی

۹۴۱۳۱۰۹۱

تیرماه ۱۳۹۵

فهرست مطالب

۱ مقدمه

تشخیص حالت دست از روی تصاویر، یکی از پرکارترین و فعال‌ترین حوزه‌های پژوهشی در زمینه ارتباط انسان و رایانه[۱] است. تعامل انسان و رایانه با استفاده از حرکات و حالات دست، علاوه بر این‌که حس بهتری به کاربر در هنگام کار با سیستم‌های رایانه‌ای می‌بخشد، راه‌حل مناسبی برای حل بسیاری از مشکلات از جمله ارتباط افراد ناتوان با رایانه است. از این‌رو توجه پژوهشگران زیادی را به خود جلب کرده است و روش‌های مختلف و گوناگونی برای حل این مشکل، ارائه شده است.

در این پژوهش، ابتدا نگاه کوتاهی بر پژوهش‌های مرتبط انجام شده در این زمینه می‌اندازیم و تعدادی از این روش‌ها را بیان کرده و مورد بررسی اجمالی قرار می‌دهیم. سپس روش ارائه شده در این پروژه را مطرح کرده و در مورد چالش‌های مختلف موجود در پیاده‌سازی این روش بحث خواهیم کرد و در انتها، نتایج آزمایشات و عملکرد الگوریتم را مورد بررسی قرار می‌دهیم.

۲ تشخیص حالت دست

روش‌های تشخیص حالت دست مبتنی بر بینایی ماشین، در حال حاضر، از جمله پویاترین بخش‌های پژوهشی مطروحه در جامعه پژوهشی بینایی ماشین است. به طور کلی می‌توان یک دسته‌بندی جامع از این روش‌ها را به شکل زیر انجام داد:

۱. روش‌های مبتنی بر اسکلت دست این دسته از روش‌ها با در نظر گرفتن اسکلت دست و انگشتان و نحوه قرار گیری بخش‌های مختلف این ساختار در کنار هم در تصویر ورودی، سعی بر دسته‌بندی تصویر ورودی بر اساس الگوهای از پیش تعیین شده، دارند.

۲. روش‌های مبتنی بر مساحت دست در این روش‌ها، تشخیص اسکلت دست و انگشتان و نواحی مختلف آن، اهمیت ندارد. این دسته از پژوهش‌ها، عموما با تکیه بر قطعه‌بندی تصویر[۲]، استخراج ویژگی‌های مختلف از تصویر حاصل و استفاده از الگوریتم‌های دسته‌بندی مختلف بر اساس این ویژگی‌ها، سعی در تشخیص حالت دست دارند.

در ادامه این بخش، با نمونه‌هایی از هریک از این روش‌ها آشنا خواهیم شد. از طرف دیگر، روش‌های ارائه شده برای حل این مساله را می‌توان از نگاهی دیگر بر اساس نوع ورودی و نوع عملکرد الگوریتم، به شکل زیر دسته‌بندی نمود:

۱. تشخیص حالت دست به طور ایستا در این دسته از روش‌ها، ورودی الگوریتم فقط یک عکس از یک حالت دست است و الگوریتم باید بتواند این عکس را در یکی از دسته‌های ازپیش‌تعیین‌شده خود قرار دهد. این روش‌ها در کاربردهای زیادی از جمله تشخیص حروف زبان ناشنوایان کاربردهای گسترده دارد.

۲. تشخیص حالت دست به طور پویا ورودی این دسته از الگوریتم‌ها، برخلاف دسته قبلی، یک تصویر ثابت نیست؛ بلکه یک دنباله‌ای از تصاویر مورد استفاده قرار می‌گیرد. در این روش‌ها با ورودی دادن یک دنباله از تصاویر که یک حرکت دست را نمایش می‌دهد، به دنبال یافتن دسته‌بندی‌کننده مناسب برای دسته‌بندی حرکت هستیم. یک نمونه از کاربردهای مربوط به این دسته از روش‌ها در بازی‌های رایانه‌ای است که در آن بازیکنان با حرکات دست خود، یک دستور خاص را به بازی ارسال می‌کنند.

در ادامه نمونه‌هایی از این دسته از الگوریتم‌ها را مورد بررسی قرار می‌دهیم.

---

[۱] Human Computer Interaction (HCI)
[۲] Image Segmentation

۱.۲ روش‌های مبتنی بر اسکلت دست

در این روش‌ها، همان‌طور که قبلا گفته شد، با استفاده از مدل اسکلت دست و انگشتان، عمل دسته‌بندی انجام می‌شود. چالش اساسی در این روش‌ها، یافتن نقاط شاخص دست و انگشتان و ارتباط بین این نقاط است که منجر به یافتن مدل اسکلت دست در تصویر ورودی می‌شود. پس از یافتن این مدل، کافیست با استفاده از الگوریتم‌های تصمیم‌گیری ساده، دسته شبیه‌ترین مدل موجود در مجموعه مدل‌های از پیش تعیین شده، به تصویر جدید اختصاص داده شود.

در روش‌های قدیمی‌تر، استفاده از نظریه تطبیق گراف[۱] نقش اساسی در نتایج پژوهش‌ها داشت. اخیرا استفاده از مدل‌های گرافی احتمالاتی[۲]، در بسیاری از پژوهش‌های مرتبط با این دسته، به چشم می‌خورد. در ادامه، نمونه‌هایی از این روش‌ها را مورد بررسی قرار می‌دهیم.

۱.۱.۲ استفاده از تطبیق گراف در تشخیص حالت دست

پژوهش [۶] با استفاده از نظریه تطبیق گراف، سعی در دسته‌بندی تصاویر شامل حالات دست در زمینه‌های ساده و پیچیده دارد. در این پژوهش، ابتدا یک گرف از مدل دست ساخته می‌شود. گره‌های این گراف را، نقاط شاخص دست مانند نوک انگشتان و محل بندهای انگشتان و نقاطی از کف دست و مچ دست تشکیل می‌دهند و بسته به موقعیت قرارگیری این گره‌ها در دست، یال‌های بین گره‌ها ایجاد می‌شوند. این مدل برای حالات از پیش‌تعیین شده که قصد تشخیص آن‌ها را داریم به صورت جداگانه تشکیل می‌شود. سپس با ورود تصویر جدید و تشخیص نقاط شاخص دست، گراف مدل اسکلت دست برای تصویر جدید ایجاد می‌شود. با ایجاد این گراف، طول یال‌های بین گره‌ها و زاویه آن‌ها قابل محاسبه است. این مدل کمک می‌کند تا موقعیت قرارگیری انگشتان و زوایای آن‌ها نسبت به یکدیگر را تشخیص دهیم. با استخراج این اطلاعات و مقایسه آن‌ها با مدل‌های از پیش تعیین شده حالات قابل تشخیص توسط الگوریتم، به راحتی می‌توان دسته مربوط به تصویر ورودی را تشخیص داد.

این روش‌ها با وجود دقت خوبی که دارند، با چالش‌های اساسی و مهمی روبرو هستند. یکی از مشکلات اصلی آن‌ها، زمان مورد نیاز برای دسته‌بندی تصاویر است که بسیار زیاد است. این مشکل از آنجا ناشی می‌شود که مرحله تشخیص گره‌های گراف و تشکیل آن بسیار وقت‌گیر و هزینه‌بر می‌شود زیرا باید از تمام نقاط کاندید صفحه تصویر، مجموعه نقاطی را که بتوانند گراف مورد نظر را بسازند به طوری‌که قیود موجود حفظ شود، با استفاده از روش‌های مبتنی بر پیمایش گراف، استخراج نمود.

۲.۱.۲ استفاده از مدل‌های گرافی احتمالی در تشخیص حالت دست

اخیرا استفاده از مدل‌های گرافی احتمالاتی در بسیاری از پژوهش‌های مربوط به تشخیص حالت دست مبتنی بر اسکلت، به طور چشم‌گیری افزایش یافته است. در عموم این پژوهش‌ها، مانند پژوهش‌های مبتنی بر تطبیق گراف، ابتدا مدل اسکلت و انگشتان دست ساخته می‌شود. تفاوت این روش‌ها با روش‌های قبلی در نحوه ایجاد گراف است. در این دسته از پژوهش‌ها، با استفاده از مدل‌های گرافی احتمالی مانند مدل میدان تصادفی مارکف[۳]، برای تشخیص گره‌های گراف و اتصالات بین آن‌ها استفاده می‌شود.

۲.۲ روش‌های مبتنی بر مساحت دست

استفاده از مساحت دست در پژوهش‌های بسیار زیادی برای تشخیص حالت دست، به‌کار گرفته شده است. در عموم این پژوهش‌ها، با استفاده از روش‌های قطعه‌بندی تصویر در مرحله پیش‌پردازش، ناحیه‌ای که شامل دست است، استخراج شده و با یک آستانه‌ای‌سازی و رفع نویز، آماده استفاده می‌شود. در مرحله بعدی، با تکیه بر ویژگی‌های مختلف، نگاشتی از فضای تصویر به فضای ویژگی‌ها برای هر تصویر انجام می‌شود. در انتها با استفاده از ابزارها و روش‌های مختلف دسته‌بندی، بردارهای ویژگی استخراج شده با استفاده از بردارهای ویژگی داده‌های آموزشی ارائه شده قبلی، دسته‌بندی می‌شود. در ادامه به تعدادی از

---

۱ Graph Matching Theory (GMT)
۲ Probabilistic Graphical Models
۳ Markov Random Field (MRF)

پژوهش‌هایی که در این زمینه فعالیت‌کرده‌اند، اشاره خواهیم کرد.

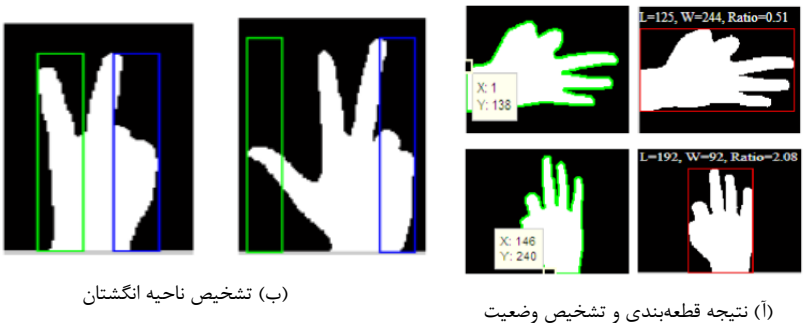
در پژوهش [۵]، الگوریتمی برای تشخیص ارقام در زبان ناشنوایان کشور انگلیس برای استفاده به عنوان ورودی یک دستگاه خودپرداز بانکی، ارائه شده است. شکل ۸، سمت راست، تصاویر مربوط به ارقام این زبان مشخص شده است. در این پژوهش، از یک آستانه‌ای سازی با توجه به رنگ پوست برای مرحله قطعه‌بندی تصاویر استفاده شده است. سپس با اعمال فیلترهای حذف نویز، نویز موجود در تصویر حذف شده است. با توجه به الگوی ارقام در زبان ناشنوایان کشور انگلیس، ویژگی‌های استخراج شده در این مقاله بسیار ساده هستند و شامل تعداد انتقال‌های سیاه به سفید در امتداد یکی از سطرها و ستون‌های تصویر مطابق با شکل۸، سمت چپ، است. سپس با استفاده از چند قانون ثابت روی ویژگی‌های استخراج شده، عمل تشخیص انجام می‌شود.

(ب) الگوی ارقام زبان ناشنوایان انگلیس

(آ) الگوی استخراج ویژگی

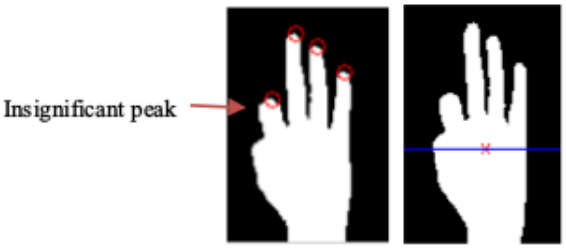
شکل ۱: الگوی ارقام در زبان ناشنوایان کشور انگلیس و الگوی استخراج ویژگی در پژوهش [۵]

در پژوهش [۴]، با استفاده از قطعه‌بندی تصویر از طریق آستانه‌ای‌سازی، ناحیه‌ای که دست در آن قرار دارد، استخراج می‌شود. سپس با محاسبه طول و عرض این ناحیه و تقسیم آن‌ها بر هم، موقعیت و زاویه دست، محاسبه می‌شود. پس از این مرحله، ویژگی‌هایی از قبیل مرکز ثقل دست و ناحیه انگشتان، از تصویر استخراج شده و در یک بردار ۵ بیتی بازنمایی می‌شود. سپس با استفاده از این بردارهای ویژگی استخراج شده، عمل دسته‌بندی انجام می‌شود. شکل ۳، تصاویر مربوط به این مقاله را نمایش می‌دهد.



(ب) تشخیص ناحیه انگشتان

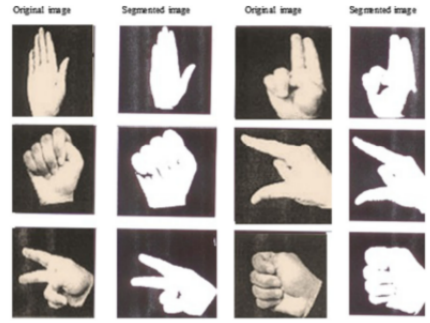
(آ) نتیجه قطعه‌بندی و تشخیص وضعیت



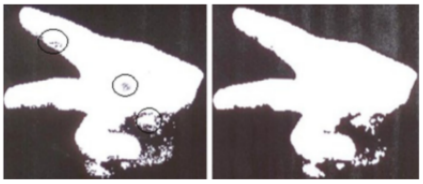
(ج) استخراج ویژگی‌های نوک انگشتان و مرکز ثقل دست

شکل ۲: مراحل مختلف پردازش در پژوهش [۴]

در پژوهش [۱]، مروری بر فعالیت‌های انجام شده مبتنی بر بینایی رایانه برای تشخیص حالت دست در سال ۲۰۱۶ انجام شده است. هم‌چنین در روش ارائه شده در این پژوهش، پس از قطعه‌بندی تصویر با استفاده از آستانه‌ای‌سازی تصویر و حذف نویز با اعمال پشت سرهم فیلتر میانه[۱]، ناحیه تصویر برای استخراج لبه‌ها به مرحله بعد ارسال می‌شود. استخراج لبه با استفاده از فیلتر سوبل[۲]، انجام می‌شود. سپس از یک شبکه عصبی پرسپترونی چندلایه[۳]، برای دسته‌بندی استفاده شده است. شبکه عصبی مورد استفاده در این پژوهش، از ۱۰۶۰ گره در لایه ورودی، ۱۰۰ گره در لایه مخفی و ۶ گره در لایه خروجی با نرخ یادگیری ۰.۹ تشکیل شده‌است. شکل؟؟ مراحل مختلف اجرای این الگوریتم را نمایش می‌دهد.



(آ) تصاویر ورودی و نتیجه قطعه‌بندی. ستون‌های دوم و چهارم نتایج قطعه‌بندی و ستون‌های اول و سوم تصاویر ورودی هستند.



(ب) حذف نویز

(ج) استخراج لبه‌ها با استفاده از فیلتر سوبل

شکل ۳: مراحل مختلف پردازش در پژوهش [۱]

روش‌های مشابه دیگری نیز ارائه شده است که به دلیل پرهیز از بلندگویی، از بیان آن‌ها صرفه‌نظر می‌کنیم. یکی از فعالیت‌های شاخصی که در این زمینه انجام شده است، پژوهش [۲] است که با استفاده از اعمال فیلترهای گابور[۴]، عمل تشخیص حالت دست را انجام داده است. این پژوهش را به طور مفصل در ادامه بررسی خواهیم نمود.

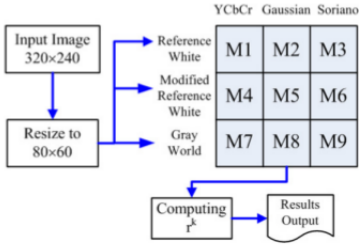
۱.۲.۲ استفاده از فیلترهای گابور

نتایج پژوهش‌ها نشان می‌دهد، ساختار چشم گربه، سلسله‌مراتبی است و سلول‌های موجود در پایین‌ترین سطح، فقط قادر به استخراج ویژگی‌های بسیار ساده از تصویر هستند. این ویژگی‌های استخراج شده، در سطوح بالاتر، با یک‌دیگر ترکیب می‌شوند و تصویر کلی را می‌سازند. نحوه کار سلول‌های سطح پایین

[۱] Median Filter
[۲] Sobel Edge Detection Filter
[۳] Multilayer Perceptron Network (MLP)
[۴] Gabor Filters

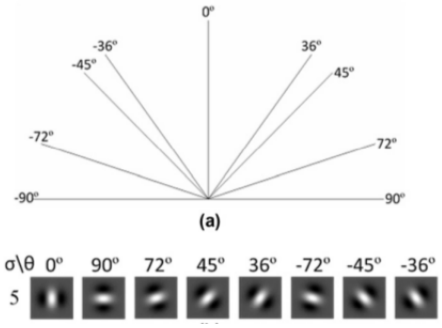
چشم، مشابه اعمال فیلترهای گاوسی در پردازش تصویر است. فیلترهای گابور، فیلترهای گاوسی هستند که در توابع سینوسی کانوالو[۱]، شده‌اند. پژوهش‌ها نشان می‌دهند، اعمال این فیلترها در زوایای مختلف، عملکرد بسیار مناسبی در قطعه‌بندی تصاویر دارد.

در پژوهش [۲]، از این فیلترها برای تشخیص حالت دست استفاده شده است. در این پژوهش، در مرحله پیش‌پردازش، ابتدا تصویر ورودی، به فضاهای رنگی مختلف نگاشت می‌شود. برای هر فضای رنگی، یک معیار ارزیابی محاسبه می‌شود. در نهایت، تصویر ورودی به فضای رنگی با بیشترین مقدار معیار ارزیابی نگاشت می‌شود و از آن برای مراحل بعدی استفاده می‌شود. این مرحله به دلیل جلوگیری از تاثیر نورهای اضافی موجود در محیط و نویز حاصل از شدت روشنایی انجام می‌شود. شکل ۴ نحوه انجام این محاسبه را نمایش می‌دهد.



شکل ۴: انتخاب مناسب‌ترین فضای رنگی برای انجام محاسبات [۲]

پس از انتخاب بهترین فضای رنگی، فیلترهای گابور را در زوایای مختلف به تصویر اعمال می‌کنیم. شکل۵، فیلترهای گابور را در زوایای مختلف نمایش می‌دهد.



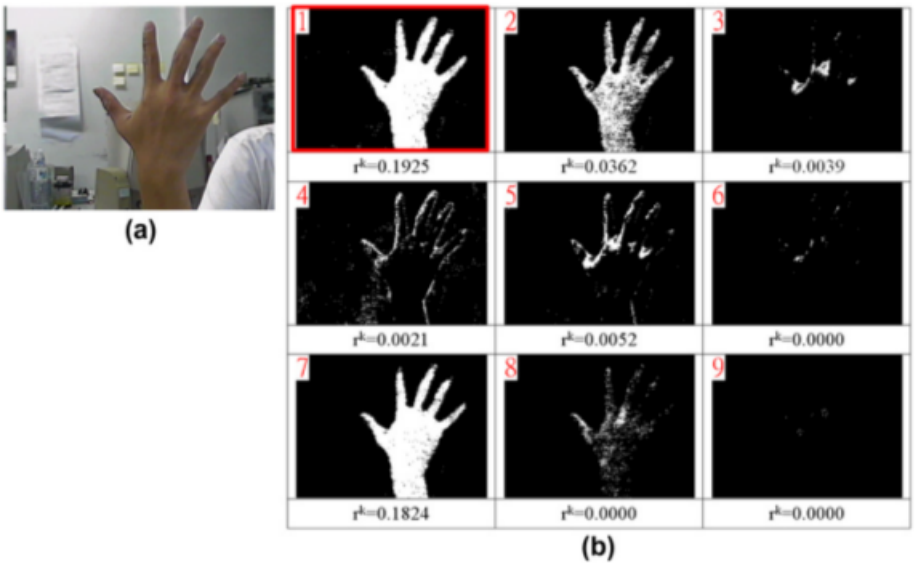
شکل ۵: فیلترهای گابور در زوایای مختلف [۲]

با اعمال هریک از فیلترها به تصویر در فضای رنگی جدید، برجستگی‌های تصویر در راستای زاویه فیلتر، استخراج می‌شود. شکل ؟؟ نتیجه اعمال این فیلترها را به یکی از تصاویر ورودی نمایش می‌دهد. همان‌طور که مشخص است، در تمام تصاویر بخش‌هایی از دست استخراج شده است و بقیه حواشی تصویر مانند آستین فرد و یا بافت زمینه به طور کامل از تصویر حذف شده‌اند.

پس از استخراج ناحیه دست توسط فیلترهای گابور و قطعه‌بندی تصویر، برای استخراج ویژگی در این پژوهش، با اسکن کردن تصویر از بالا به پایین و در راستای سطرها، هیستوگرام گرادیان[۲] تصویر به عنوان ویژگی‌های کاندید محاسبه می‌شود. شکل۷ هیستوگرام گرادیان یک تصویر را نمایش می‌دهد. سپس با استفاده از الگوریتم PCA، کاهش بعد روی این هیستوگرام انجام شده و ویژگی‌های مفید برای دسته‌بندی استخراج می‌شوند. ویژگی‌های استخراج شده، بعدا با استفاده از دسته‌بندی‌کننده SVM دسته‌بندی می‌شوند.

---

[۱] Convolve
[۲] Histogram of Gradient (HoG)



شکل ۶: نتیجه اعمال فیلترهای گاوسی در زوایای مختلف به تصویر [۲]

شکل ۷: محاسبه هیستوگرام تصویر پاسخ فیلترهای گابور به تصویر ورودی [۲]

۳ پیاده‌سازی

در این پروژه، با استفاده از فیلترهای گابور، قطعه‌بندی روی تصویر ورودی انجام می‌شود. سپس با استفاده از ویژگی‌های سیفت[۱]، تصویر را به بردار ویژگی تبدیل کرده و با استفاده از دسته‌بندی کننده SVM، عمل دسته‌بندی را انجام می‌دهیم. در ادامه به بررسی بخش‌های مختلف کد خواهیم پرداخت.

۱.۳ پیش‌پردازش داده‌های ورودی

ابتدا رزولوشن تصویر ورودی را با استفاده از هرم دقت[۲]، کاهش می‌دهیم. این کار به افزایش سرعت اجرای الگوریتم کمک بسیار زیادی می‌کند. برنامه ??، بخشی از کد را که این کار را انجام می‌دهد نمایش می‌دهد. این کار را تا زمانی انجام می‌دهیم که ابعاد تصویر ورودی از ابعاد از پیش‌تعیین‌شده‌ای کمتر شود تا مطمئن باشیم الگوریتم، کارایی خود را حفظ خواهد کرد.

کاهش رزولوشن تصویر ۱: برنامه

```
1  Mat input = input_src  ;
2  while(input.rows > max_size || input.cols > max_size)
3  {
4    pyrDown(input_src,input,Size(input_src.cols/2,input_src.rows/2));
```

[۱] SIFT Features
[۲] Resolution Pyramid

```
    input_src = input ;
  }
```

در مرحله پیش‌پردازش، ما برخلاف پژوهش [۲]، به طور پویا، فضای رنگی مورد استفاده را انتخاب نمی‌کنیم. بلکه در این بخش، ما به طور مشخص، تصاویر ورودی را به فضای رنگی YCrCb منتقل می‌کنیم. ویژگی این فضا این است که به طور کامل، روشنایی نقاط را از رنگ آن‌ها جدا می‌کند. با این کار، تصویر ورودی را که در فضای رنگی RGB تولید شده است و دارای سه کانال رنگی است به فضای رنگی YCrCb که دارای سه کانال رنگی است منتقل می‌کنیم. از این تصویر در مراحل بعد استفاده می‌کنیم. برنامه ۱.۳ بخشی از کد را که انتقال فضای رنگی را انجام می‌دهد، نمایش می‌دهد.

انتقال فضای رنگی ۲: برنامه

```
cout << "converting color space"<<endl;
cvtColor(input, input, CV_BGR2YCrCb);
input *= float(1)/255;
```

۲.۳ اعمال فیلترهای گابور

در این بخش، فیلترهای گابور را با زوایای مختلف به تصاویر هر یک از کانال‌های تصویر به صورت جداگانه اعمال می‌کنیم. این کار باعث می‌شود متوجه شویم کدام‌یک از کانال‌های موجود، بهترین تفکیک را انجام می‌دهد. برنامه ۲.۳، فیلترهای گابور را در زوایای مختلف به تصویر اعمال می‌کند. این فیلتر ها در زوایای صفر تا ϕ با فاصله ۱۵ درجه، اعمال می‌شوند.

اعمال فیلترهای گابور در زوایای مختلف به تصویر پیش‌پردازش شده ۳: برنامه

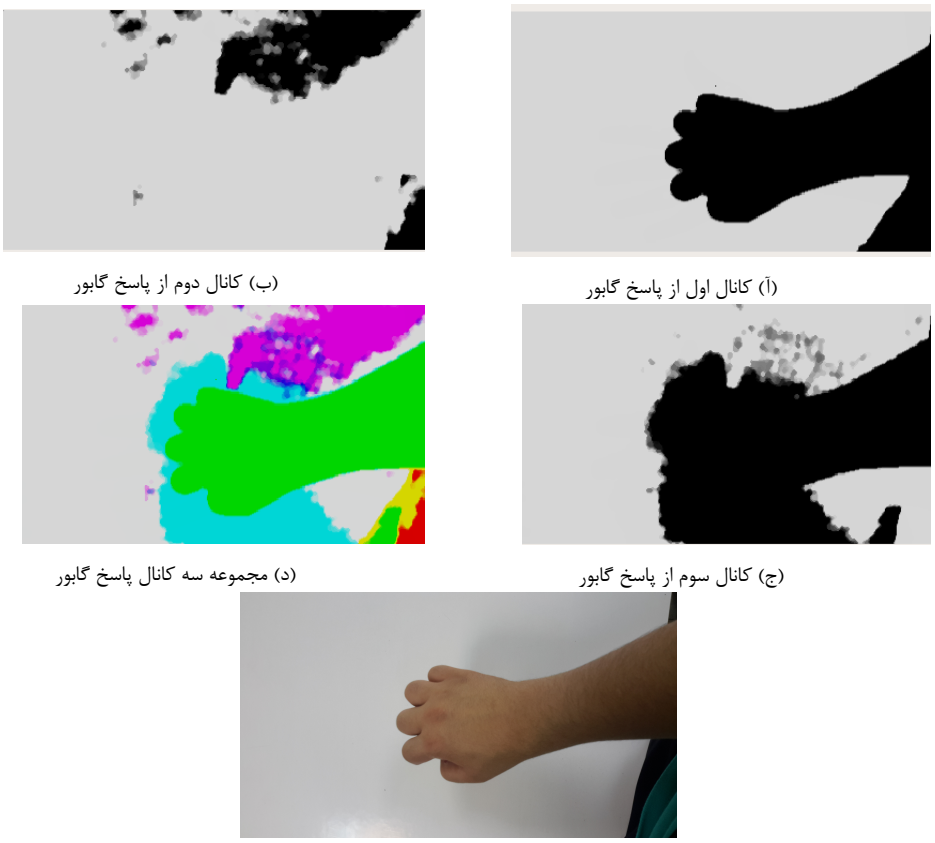
```
cout << "filter convolution" <<endl;
Mat weighted_sum_image (input.rows, input.cols, 21);
int kernel_size = 3;
double sig = 1, th = 0, lm = 1.0, gm = 1, ps = 1;
for (;th <= 180 ; th += 15)
{
  if(DEBUG)
    cout <<"Theta:" << th << endl;
  Mat kernel = getGaborKernel(Size(kernel_size, kernel_size), sig, th, lm, gm, ps);
  Mat filtered_image ;
  filter2D(input, filtered_image,  CV_32F, kernel);
  addWeighted(weighted_sum_image, 0.5 , filtered_image , 0.5 , 0, weighted_sum_image ,
      weighted_sum_image.type()) ;
}

for(int i = 0 ; i < 200 ; i++)
  medianBlur(weighted_sum_image, weighted_sum_image , 3);
```

همان‌طور که مشاهده می‌شود، **برخلاف تمام پژوهش‌ها**، ما پاسخ فیلترهای مختلف را به صورت جداگانه بررسی نمی‌کنیم. بلکه از آنجایی‌که تمام این فیلترها، نواحی مختلف دست در راستاهای مختلف را برجسته می‌کنند، با تجمیع همه آن‌ها در یک تصویر، تصویر کامل دست را در فضا تشکیل می‌دهیم. شکل ؟؟ تصاویر حاصل از این مرحله را نمایش می‌دهد.

همان‌طور که در شکل ؟؟ مشخص است، اولین کانال از این پاسخ، بهترین پاسخ ممکن را برای قطعه‌بندی تصویر می‌دهد. در ادامه کار روی خروجی این کانال تمام پردازش‌ها را انجام خواهیم داد.

۳.۳ استخراج ویژگی‌های سیفت

در این مرحله به دنبال یافتن بهترین توصیف‌گرها برای تصاویر هستیم. در ادامه این بخش، مراحل مختلف این کار را مورد بررسی قرار خواهیم داد.



(آ) کانال اول از پاسخ گابور

(ب) کانال دوم از پاسخ گابور

(ج) کانال سوم از پاسخ گابور

(د) مجموعه سه کانال پاسخ گابور

(ه) تصویر ورودی

شکل ۸: نتایج اعمال فیلترهای گاوسی در زوایای مختلف و تجمیع پاسخ‌ها

۱.۳.۳ استخراج نقاط کلیدی

برای استخراج نقاط کلیدی از برنامه ۱.۳.۳ استفاده می‌نماییم. در این مرحله ابتدا تصویر ورودی را با یک آستانه‌ای‌سازی ثابت و ساده، آستانه‌ای کرده و سپس به سطح خاکستری تبدیل می‌کنیم. سپس با استفاده از توابع کتابخانه اپن‌سی‌وی[۱]، نقاط کلیدی را استخراج می‌نماییم.

استخراج نقاط کلیدی ۴: برنامه

```
image = threshold_and_convert(image);

SiftFeatureDetector detector;
  vector<KeyPoint> keypoints;
  detector.detect(image, keypoints);
```

۲.۳.۳ محاسبه توصیف‌گر سیفت

پس از استخراج نقاط کلیدی توسط الگوریتم، می‌توانیم توصیف‌گر تصویر را بسازیم. برنامه ۲.۳.۳، این توصیف‌گر را برای هر تصویر با استفاده از نقاط کلیدی استخراج شده، محاسبه می‌کند و در قالب یک ماتریس بازنمایی می‌نماید.

محاسبه توصیف‌گر سیفت ۵: برنامه

```
SiftDescriptorExtractor extractor ;

```

[۱] OpenCV

```
Mat descriptor;

image.convertTo(image, CV_8U);

extractor.compute(image, keypoints , descriptor);
```

۴.۳ تولید بردار ویژگی با استفاده از ویژگی‌های سیفت استخراج شده

توصیف‌گر سیفت، یک ماتریس با ۱۲۸ ستون و به تعداد نقاط کلیدی استخراج شده سطر است. این ماتریس، ۱۲۸ ویژگی از هر کدام از نقاط کلیدی را مشخص می‌کند. برای تبدیل این توصیف‌گر به یک بردار ویژگی، با توجه به این‌که از هر تصویر تعداد متفاوتی نقطه کلیدی استخراج می‌شود، باید از روش‌های مبتنی بر کیسه کلمات[۱] استفاده کنیم. به این منظور، ماتریس توصیف‌گر سیفت را به صورت ستونی جمع می‌کنیم و عملیات نرمال‌سازی را روی هر ستون انجام می‌دهیم. بردار حاصل، بردار ویژگی تولید شده از توصیف‌گر سیفت است. برنامه ۴.۳ کد مربوط به تولید بردار ویژگی را نمایش می‌دهد.

تولید بردار ویژگی ۶: برنامه

```
  double feature_value[descriptor.cols] ;
  double gamma = 0.05 ;

  for(int j = 0 ; j < descriptor.cols ; j++)
  {
    feature_value[j] = 0 ;
    for(int h = 0 ; h < descriptor.rows ; h++)
    {
      double to_be_added_value = (descriptor.at<double>(h,j)/pow(10,15) ) ;
      feature_value[j] += to_be_added_value;

      if(feature_value[j] > 1)
        feature_value[j] = 1 ;
    }
  }


  for(int j = 0 ; j < descriptor.cols ; j++)
  {
    if(feature_value[j] < 0 )
      feature_value[j] = 0 ;

    feature_value[j] = 128 * pow(feature_value[j],gamma) ;
    if(DEBUG)
      cout << "feature_value "<< j<<": " << feature_value[j] << endl ;
  }

  Mat result (1,descriptor.cols, CV_32FC1, feature_value) ;
```

۵.۳ آموزش دسته‌بندی کننده و استفاده از آن

آموزش و استفاده از دسته‌بندی کننده برای پیش‌بینی بچسب یک تصویر که به بردار ویژگی نگاشت شده است، با فراخوانی دو تابع، به راحتی انجام می‌شوند.

آموزش دسته‌بندی کننده ۷: برنامه

```
training_svm.train(training_data_mat, training_labels_mat, Mat(), Mat(), params);
```

پیش‌بینی با استفاده از دسته‌بندی‌کننده SVM ۸: برنامه

```
float predicted_label = svm.predict(feature_vector);
```

[۱] bag of words

۴ آزمایشات

مجموعه‌داده مورد آزمایش در این پروژه، مجموعه‌داده مورد استفاده در مقاله [۳] است که با استفاده از روش‌های فازی، اقدام به دسته‌بندی تصویر کرده است. در این مجموعه‌داده، ۱۰ کلاس مختلف از حالات دست وجود دارد که در آن از هر دسته حالات، ۲۴ نمونه تصویر وجود دارد. نمونه‌های موجود در هرکدام از کلاس‌های تصویری موجود در این مجموعه‌داده، با تغییر دادن موقعیت و اندازه تصاویر و چرخش‌های جزئی، بوجود آمده‌اند. زمینه تمام تصاویر، ساده و یکنواخت است. شکل۹، یک نمونه تصویر از هر کدام از کلاس‌های موجود را نمایش می‌دهد.

(آ) نمونه تصویر از کلاس اول (ب) نمونه تصویر از کلاس دوم (ج) نمونه تصویر از کلاس سوم (د) نمونه تصویر از کلاس چهارم

(ه) نمونه تصویر از کلاس پنجم (و) نمونه تصویر از کلاس ششم (ز) نمونه تصویر از کلاس هفتم (ح) نمونه تصویر از کلاس هشتم

(ط) نمونه تصویر از کلاس نهم (ی) نمونه تصویر از کلاس دهم

شکل ۹: نمونه تصاویر موجود در مجموعه‌داده

جدول ۱، نتایج اعمال روش پیاده‌سازی‌شده را بر روی مجموعه‌داده مورد استفاده گزارش می‌کند.

جدول ۱: نتایج نهایی اعمال روش ارائه شده به مجموعه‌داده

	دقت پیش‌بینی	زمان اجرا برای هر تصویر
مجموعه آموزشی	۸۳.۷۵	۰.۰۵۹
مجموعه تست	۳۳.۳۴	۰.۰۵۶

همین‌طور جدول ۲، نتیجه پیش‌بینی الگوریتم به ازای هر کدام از کلاس‌ها را به طور جداگانه گزارش می‌دهد. همان‌طور که مشخص است، تصاویر کلاس‌های ۵، ۷، ۸، ۹ و ۱۰ بسیار به هم شبیه هستند. این مورد باعث می‌شود، در زمان آموزش، تعداد ویژگی‌هایی که به کلاس ۱۰ شبیه هستند، زیاد شود. دسته‌بندی‌کننده در مواجهه با چنین مشکلی، بیشتر به سمت کلاس ۱۰ متمایل می‌شود و روی داده‌های این کلاس، به نوعی بیش‌برازش اتفاق می‌افتد. مؤید این نکته این است که در جدول ۲، روی مجموعه تست، دقت این کلاس برابر با ۱۰۰ شده و بقیه کلاس‌ها به شدت پایین هستند. برای حل این مساله باید با اضافه کردن ویژگی یا ویژگی‌هایی که بتواند این کلاس‌ها را از هم متمایز کند، دقت دسته‌بندی را افزایش دهیم. استفاده از الگوریتم‌هایی مانند PCA هم که کاهش بعد انجام داده و ویژگی‌های تکراری را از بین می‌برند، می‌تواند بسیار مفید باشد.

جدول ۲: نتایج نهایی اعمال روش ارائه شده به مجموعه‌داده به تفکیک کلاس‌ها

شماره کلاس	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
مجموعه آموزشی	۵۰	۷۵	۸۷.۵	۸۳.۳۴	۹۵.۸۴	۸۷.۵	۸۷.۵	۹۱.۶۷	۷۹.۱۶	۱۰۰
مجموعه تست	۲۲.۲۳	۲۲.۲۳	۳۳.۳۴	۱۱.۱۲	۳۳.۳۴	۱۱.۱۲	۲۲.۲۳	۳۳.۳۴	۴۴.۴۵	۱۰۰

۵ جمع‌بندی

تشخیص حالت دست، به طور کلی به یکی از دو روش زیر انجام می‌شود:

۱. تشخیص مبتنی بر اسکلت دست در این دسته از روش‌ها، با استفاده از نظریه تطبیق گراف یا با استفاده از مدل‌های گرافی احتمالی، سعی در بازسازی مدل اسکلت دست در هر تصویر می‌شود. با داشتن مدل اسکلت دست، می‌توان عمل دسته‌بندی را به خوبی انجام داد.

۲. تشخیص مبتنی بر مساحت دست در این دسته از روش‌ها، با استفاده از قطعه‌بندی تصویر و آستانه‌سازی آن و استخراج ویژگی از تصاویر حاصل برای استفاده در دسته‌بندی کننده، می‌تواند منجر به دسته‌بندی تصاویر بر حسب حالات مختلف دست شود.

استفاده از فیلترهای گابور برای برجسته‌کردن نقاط مهم تصویر از جمله روش‌هایی است که در تشخیص مساحت دست بسیار مفید است. روند انجام عملیات در روش ارائه شده در این مقاله به طور خلاصه به شرح زیر است:

۱. انتقال فضای رنگی دست از RGB به YCrCb

۲. کاهش رزولوشن تصویر با استفاده از هرم دقت

۳. اعمال فیلترهای گابور به کانال‌های مختلف تصویر و انتخاب بهترین کانال

۴. استخراج نقاط کلیدی سیفت

۵. استخراج توصیف‌گر سیفت برای تصویر

۶. تولید بردار ویژگی بر اساس روش کیسه کلمات از توصیف‌گر سیفت

۷. آموزش و استفاده از دسته‌بندی کننده SVM

روش ارائه شده در این پروژه، با اضافه کردن ویژگی‌های متمایز کننده کلاس‌های ۷ تا ۱۰ از یک‌دیگر، قابلیت ارتقا عملکرد زیادی دارد.

مراجع

[1] Badi, H. Recent methods in vision-based hand gesture recognition. *International Journal of Data Science and Analytics* (2016), 1–11.

[2] Huang, D.-Y., Hu, W.-C., and Chang, S.-H. Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Expert Systems with Applications 38*, 5 (2011), 6031–6042.

[3] Kumar, P. P., Vadakkepat, P., and Loh, A. P. Hand posture and face recognition using a fuzzy-rough approach. *International Journal of Humanoid Robotics 7*, 03 (2010), 331–356.

[4] Panwar, M. Hand gesture recognition based on shape parameters. in *2012 International Conference on Computing, Communication and Applications* (2012), IEEE, pp. 1–6.

[5] Rupanagudi, S. R., Ranjani, B., Bhat, V. G., Surabhi, K., Reshma, P., Shruthi, G., Sarayu, K., Sangeetha, R., Rao, B. R., and Vasanti, S. A high speed algorithm for identifying hand gestures for an atm input system for the blind. in *2015 IEEE Bombay Section Symposium (IBSS)* (2015), IEEE, pp. 1–6.

[6] Triesch, J., and Von Der Malsburg, C. A system for person-independent hand posture recognition against complex backgrounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 12 (2001), 1449–1453.