

به نام خدا

الیار علیزاده

عرفان علی محمدی

تشخیص حروف فارسی
(OCR)

معلمان راهنما:

آقایان صبّاحی و رضی زاده

- ۲.....مقدمه
- ۳.....مراحل و نحوه‌ی کار برنامه
- ۴.....مرحله‌ی پیش پردازش
- ۵ و ۶.....مرحله‌ی استخراج ویژگی‌ها
- ۷.....شبکه‌ی عصبی
- ۸.....پاسخ‌گویی نهایی
- ۹.....چشم‌انداز آینده‌ی برنامه

OCR چیست؟

مخفف **Optical Character Recognition** می‌باشد و در واقع نرم افزار متن خوانی است که برای تشخیص و بازیابی الفبا و نوشته‌های دست نویس یا تایپ شده طراحی شده است. پس با این نرم افزار می‌توان متن‌های دست‌نویس یا متونی را که قبلاً با ماشین تحریر تایپ شده یا نوشته‌های چاپ شده را به صورت خودکار به متن کامپیوتری تبدیل نمود. این عمل یعنی تبدیل تصویر کلمات به کد کلمات قابل ذخیره در داخل کامپیوتر. بدیهی است که هر متنی که در کامپیوتر ذخیره شود، می‌توان روی آن پردازش‌های دلخواهی انجام داد. برای مثال، می‌توان آن را ویرایش کرد.

کاربردهای OCR

یکی از کاربردهای OCR را می‌توان تبدیل حروف و نوشته‌های خطی، به خط بریل (مخصوص نابینایان) دانست؛ به این ترتیب همگی نوشته‌ها توسط نابینایان قابل خواندن خواهند بود. البته استفاده از OCR تنها برای تبدیل تصویر متون تایپ شده به متن تایپی نیست. بلکه هر جا شما عکسی داشته باشید که در قسمتی از آن حرف یا شماره‌ای وجود داشته باشد، نرم افزار OCR می‌تواند آن را تشخیص می‌دهد. به عنوان مثال، می‌توان از آن در دوربین‌های هوشمند سرعت سنج جاده‌ها استفاده کرد. دوربین سرعت سنج، به صورت مستمر سرعت خودروها را اندازه‌گیری کرده و بعد از شناسایی جسم متحرک (خودرو)، از پلاک خودرو عکس گرفته و به وسیله‌ی این نرم‌افزار پلاک را شناسایی می‌کند. البته این نرم افزار کاربردهای دیگری نیز دارد که از آن‌ها می‌توان به وارد کردن ثبت نام سازمان‌ها و ادارات (که حروف را جدا از هم می‌نویسند) در کامپیوتر بدون دخالت انسان و همچنین تصحیح جداول و سودو کو اشاره نمود.

انواع OCR

OCR به دو دسته‌ی کلی **online** و **offline** تقسیم می‌شود.

در نوع **online** مسیر نوشته شدن حرف بسیار اهمیت دارد و از آن برای تشخیص نوشته‌ی دست‌نویس در رایانه‌ها و تفن‌های همراه لمسی استفاده می‌گردد.

در نوع **offline** مسیر نوشته شدن حرف برای نرم‌افزار مشخص نمی‌شود و از آن برای تشخیص متون اسکن شده استفاده می‌شود. نرم‌افزار نوشته‌شده توسط ما نیز، از این نوع است.

تصویر ورودی

پیش پردازش

- حذف نویز
- بریدن اضافات تصویر
- کوچک سازی
- سیاه و سفید کردن
- نازک سازی
- جدا کردن نقطه ها

استخراج ویژگی

- تعداد نقطه ها
- تعداد تقاطع ها
- تعداد خطوط عمودی
- تعداد خطوط افقی
- تعداد مناطق بسته
- هیستوگرام
- تابش نور

شبکه‌ی عصبی

- شبکه‌ی عصبی ویژگی ها
- شبکه‌ی عصبی پیکسل ها

تشخیص حرف

خروجی

حرف تشخیص
داده شده

گرفتن تصویر ورودی



- این عکس، ورودی برنامه و در ابعاد 225×180 است.
- در این تصویر یک حرف فارسی با رنگ تیره (مثلاً سیاه) کشیده شده است.
- این تصویر می‌تواند همراه نویز باشد.

کاهش نویز تصویر



- در این مرحله، با استفاده از Gaussian blur از نویز تصویر می‌کاهیم و اختلاف رنگ‌ها را کاهش می‌دهیم.

افزایش تفاوت میان رنگ‌های روشن و تیره



- در این مرحله، با تغییر در میزان روشنایی پیکسل‌ها بین رنگ‌های روشن و تیره اختلافی آشکار ایجاد می‌نماییم.

کوچک‌سازی و باینری کردن تصویر



- به دلیل زیاد بودن حجم اطلاعات، سعی می‌کنیم تا از تصویر پیشین، تصویری ساده و کم حجم ایجاد نماییم.
- تصویر جدید در ابعاد 24×30 است.
- هر پیکسل تصویر جدید فقط می‌تواند سیاه یا سفید باشد.

نازک‌سازی



- برای ساده شدن بیش‌تر تصویر، اقدام به نازک کردن قسمت سیاه‌رنگ می‌کنیم.
- برای این کار از الگوریتم Zhang-Suen استفاده شده است.
- برای انجام نازک‌سازی لازم است که تعدادی از پیکسل‌های سیاه، حذف شوند.
- پس از نازک‌سازی، لازم است که حرف به گوشه‌ی تصویر منتقل شود.

جداسازی نقطه‌ها



- نقطه‌های حرف که کوچک‌ترین منطقه‌ی به هم پیوسته‌ی سیاه را تشکیل می‌دهند، اطلاعات مهمی را می‌توانند به ما منتقل کنند. پس بهتر است که نقطه‌ها را از حرف جدا کنیم و آن‌ها را به صورت جداگانه ذخیره کنیم.

ایجاد تصویر 15×15 

- این تصویر برای شبکه‌ی عصبی ایجاد می‌شود.
- نقطه‌های حرف را در این تصویر جای نمی‌دهیم.
- این تصویر در بسیاری از حروف مشابه است؛ مانند ف و ق یا ب و ن.

در مرحله‌ی استخراج ویژگی‌ها، سعی می‌کنیم از حرف ویژگی‌هایی استخراج کنیم تا بتوانیم حروف را از روی ویژگی‌های آن‌ها تشخیص دهیم.

ویژگی‌های استخراج شده از حروف در برنامه:

منطقه‌های بسته



- در مورد این ویژگی باید گفت که دلیل اهمیت آن این است که در بعضی از حرف‌ها (مانند ص، ض، ط، ف، ق، ه و ...) وجود دارد ولی در تعدادی نیز وجود ندارد.
- برای یافتن تعداد مناطق محصور شده، با استفاده از الگوریتم بازگشتی اگر پیکسلی پیدا بشود که فقط با تعداد محدودی خانه در ارتباط است و آن نقاط توسط نقاط سیاه محصور شده‌اند، می‌توان فهمید که آن منطقه، یک منطقه‌ی بسته است.

تقاطع‌ها



- تعریف از تقاطع جایی است که در آن حداقل ۲ خط به یکدیگر برسند یا از آن نقطه عبور کنند.
- اگر برنامه پیکسلی را بیابد که از ۸ همسایه‌ی آن حداقل ۳ قسمت از پیکسل‌ها سیاه هستند، می‌توان گفت که از آن پیکسل حداقل ۲ خط گذشته است؛ پس در آن محل یک تقاطع وجود دارد.

خطوط عمودی



- در این مورد سعی شده تا خط‌های بزرگ عمودی در حرف پیدا شود. برای این کار به دنبال سری پیکسل‌های تقریباً پشت سر هم می‌رویم.
- امکان دارد خط دارای کمی خمیدگی باشد ولی چون خمیدگی کم است، آن را به طور کلی یک خط به شمار می‌آوریم (نه ۲ خط).

خطوط افقی



- در این مورد سعی شده تا خط‌های بزرگ افقی در حرف پیدا شود. برای این کار به دنبال سری پیکسل‌های تقریباً پشت سر هم می‌رویم.
- امکان دارد خط دارای کمی خمیدگی باشد ولی چون خمیدگی کم است، آن را به طور کلی یک خط به شمار می‌آوریم (نه ۲ خط).

نقطه‌ها و مکانشان



- در ابتدا اگر نقطه‌ها چند قسمت بودند هر کدام را به طور جداگانه ذخیره می‌کنیم.
- سپس دو پیکسل دور از هم را در هر قسمت می‌یابیم و سعی می‌کنیم با وصل کردن این دو پیکسل به یکدیگر، باعث تشکیل منطقه‌ای بزرگ و احاطه شده شویم. اگر چنین منطقه‌ای بسته‌ای به وجود آمد، حرف دارای ۳ نقطه است؛ در غیر این صورت، حرف دارای ۱ یا ۲ نقطه است.
- تشخیص ۱ نقطه از ۲ نقطه از روی طول پیکسل‌های به هم پیوسته امکان‌پذیر است.
- مکان نقاط را نسبت به بالاترین نقطه‌ی حرف مقایسه می‌شود که آیا از بالاترین پیکسل حرف بالاتر است (مانند ت، خ، ش و...) یا بین بالاترین و پایین‌ترین پیکسل حرف است (مانند ج، چ و...) یا پایین‌تر از پایین‌ترین نقطه حرف قرار دارد (مانند ب، پ و...).

شبکه‌ی عصبی چیست؟

شبکه‌ی عصبی روشی است که در برنامه‌هایی که روش مشخصی برای حل مسئله وجود ندارد و انسان در آن بهتر از کامپیوتر است استفاده می‌شود؛ مثل OCR.

شبکه‌ی عصبی الگوریتم و روشی است که به چندین ویژگی و عدد، بر حسب تکرار، ضرایبی را نسبت می‌دهد.

برای مثال وقتی که شبکه‌ی عصبی دریابد اکثر «ض»های کشیده‌شده ۱ منطقه‌ی بسته و تقاطع دارند، پس ضریب آن‌ها را زیاد می‌کند که به واسطه‌ی آن اگر حروف رسم شده ۱ منطقه بسته داشته باشند احتمال «ض» بودن آن‌ها افزایش پیدا می‌کند.

کاربرد شبکه‌ی عصبی در OCR

در این برنامه از ۲ شبکه عصبی استفاده شده است (که یکی در داخل دیگری است)؛ هر دو نیز دارای ۳ لایه می‌باشند.

شبکه‌ی اول برای تشخیص شکل ظاهری حرف استفاده می‌گردد و با استفاده از شبکه‌ی عصبی ۳ لایه‌ی خود نتیجه را به شبکه‌ی عصبی دوم می‌دهد. نتیجه و خروجی شبکه‌ی عصبی اول به همراه ویژگی‌های دیگر به شبکه‌ی عصبی دوم داده می‌شود و آن نتیجه‌ی نهایی را بیان می‌کند.

لازم به ذکر است که در ابتدا شبکه‌های عصبی برای تنظیم ضرایب با ۲۵ نمونه آموزش می‌بینند.



پاسخ شبکه‌های عصبی عددی بین ۱ تا ۱۶ و نشان‌دهنده‌ی گروهی از حروف است که تفاوت آن‌ها در تعداد یا مکان حروف است. برای مثال: اگر حرف تشخیص داده‌شده از گروه ۳ باشد و دارای ۱ نقطه بالای حرف باشد، آن حرف «خ» است.

گروه‌های حروف مشابه

- | | |
|---------------|------|
| ۱ | (۱) |
| ب، پ، ت، ث، ن | (۲) |
| ج، چ، ح، خ | (۳) |
| د، ذ | (۴) |
| ر، ز، ژ | (۵) |
| س، ش | (۶) |
| ص، ض | (۷) |
| ط، ظ | (۸) |
| ع، غ | (۹) |
| ف، ق | (۱۰) |
| ک، گ | (۱۱) |
| ل | (۱۲) |
| م | (۱۳) |
| و | (۱۴) |
| ه | (۱۵) |
| ی | (۱۶) |

برنامه سعی می‌کند از میان حروف فارسی، حرفی را که بیش‌ترین همخوانی با ویژگی‌ها دارد را انتخاب کند و آن را به عنوان خروجی اعلام کند. اگر در آخر هیچ گروهی از حروف با اطلاعات شبکه عصبی سازگار نبود، در خروجی نوشته می‌شود: «مشخص نیست».

این برنامه برای حروف جدا و برای ۱ حرف با حدود ۸۰ درصد پاسخ می‌دهد ولی به امید خدا سعی در این است که درصد خطا را کاهش دهیم و آن را برای حروف به هم چسپیده نیز بسازیم.

از معلمانمان که با راهنمایی‌هایشان ما را در این کاریاری کردند، بسیار تشکر می‌کنیم.