روشی نو در تشخیص حروف در متون چاپی عربی و فارسی با استفاده از پویش خط زمینه

محبوبه شمسی[۱]، عبدالرضا رسولی کناری[۲]، سوده شادروان[۳]
۱- مربی، گروه کامپیوتر، دانشگاه آزاد اسلامی، واحد بردسیر، mh.shamsi@yahoo.com
۲- مربی، گروه کامپیوتر، دانشگاه آزاد اسلامی، واحد بردسیر، rs.reza@gmail.com
۳- مربی، گروه کامپیوتر، دانشگاه آزاد اسلامی، واحد بردسیر، shadrava123@yahoo.com

چکیده

مطمئناً یکی از مراحل مهم در سیستم‌های تشخیص متن جداسازی یا قطعه‌بندی حروف است چرا که جداسازی نادرست حروف منجر به خطاهای بسیاری در تشخیص حروف خواهد گشت. بنابر ویژگی‌های خاص زبان فارسی و عربی که ذکر خواهد شد این عمل در مورد این دو زبان به مراتب مشکل‌تر از زبان‌های لاتین از جمله زبان انگلیسی است و هنوز الگوریتم مناسبی برای این کار معرفی نشده است. الگوریتم‌های موجودکه پایه روش این مقاله قرار گرفته است تا حدود ۸۵٪ موفق می‌باشد. در این مقاله یک روش جدید بهبود یافته با بررسی ویژگی‌های بصری زبان فارسی و عربی معرفی شده است. الگوریتم جدید توانسته است با بعضی از قلم‌های چاپی موجود کارآیی حدود ۹۸/۵٪ و حتی در موارد خاص تا ۱۰۰٪ داشته باشد. مابقی خطاها را هم می‌توان با یک روش مناسب تشخیص کاراکتر و یک پایگاه داده لغات کامل فارسی حل نمود.

واژه‌های کلیدی

سیستم تشخیص متن فارسی و عربی، قطعه بندی حروف، ملایم سازی، روش پویش خط زمینه

۱– مقدمه

سیستم تشخیص اتوماتیک حروف جهت تبدیل متون قابل فهم برای انسان به کاراکترها با فرم دیجیتالی است و هدف غائی آن دستیابی به کلیه قابلیت‌های قرائت انسان است [۱]. تکنولوژی OCR از انواع شناسائی الگو محسوب می‌شود و دارای کاربردهای مهمی همچون کمک به قرائت روشندلان، اتوماسیون اداری و فرآیند تولید کتابخانه با منابع دیجیتالی می‌باشد.

باوجود این که حروف فارسی در نوشتار زبان‌های بسیاری مانند عربی, فارسی, اردو و ... بکار می‌رود، تحقیق کمتری روی شناسایی الفبای زبان فارسی انجام گشته است و اغلب تحقیقات بر روی الفبای لاتین، چینی انجام شده است [۲].

ویژگی‌های الفبای زبان فارسی [۳] به قرار زیر است:

– زبان فارسی دارای الفبای متصل است که از راست به چپ نوشته می‌شود. لذا شناسایی آن هم از راست به چپ است.

– زبان فارسی ۳۲ حرف دارد که هرکدام ۴-۲ نماد مختلف، بسته به موضع دارد. این ویژگی باعث می‌شود که زبان فارسی در واقع حدود ۱۰۰ علامت برای الفبا داشته باشد (شکل (۱)).

– حروف ممکن است یک نقطه، دو نقطه، سه نقطه یا یک زیگزاگ مرتبط با خود داشته باشد و می‌تواند بالا، پایین یا حتی داخل حرف باشند.

– کلمه‌ها ممکن است یک یا چند کلمه فرعی داشته باشند. این به‌دلیل آن است که بعضی حروف الفبا از سمت چپ قابل اتصال به حرف بعد نمی‌باشند.

– بعضی از حروف ممکن است در دامنه افقی خود هم‌پوشانی داشته باشند.

تحقیق در حیطه شناسایی الفبای زبان عربی تا حدی تازه است و صرفاً از سال ۱۹۸۰ میلادی به بعد انجام گرفت [۴] در این مقاله ابتدا به بررسی اجمالی روش‌های قبل برای تقسیط و قطعه‌بندی کاراکترها می‌پردازیم. سپس یک الگوریتم قطعه‌بندی جدید برپایه الگوریتم‌های قبلی پیشنهاد می‌کنیم.

حرف	اشکال مختلف			
الف	آ	ا	ـا	
ب	بـ	ـبـ	ـب	ب
پ	پـ	ـپـ	ـپ	پ
ت	تـ	ـتـ	ـت	ت
ث	ثـ	ـثـ	ـث	ث
جیم	جـ	ـجـ	ـج	ج
چ	چـ	ـچـ	ـچ	چ
ح	حـ	ـحـ	ـح	ح
خ	خـ	ـخـ	ـخ	خ
دال	د	ـد		
ذال	ذ	ـذ		
ر	ر	ـر		
ز	ز	ـز		
ژ	ژ	ـژ		
سین	سـ	ـسـ	ـس	س
شین	شـ	ـشـ	ـش	ش

حرف	اشکال مختلف			
صاد	صـ	ـصـ	ـص	ص
ضاد	ضـ	ـضـ	ـض	ض
طا	ط	ـط		
ظا	ظ	ـظ		
عین	عـ	ـعـ	ـع	ع
غین	غـ	ـغـ	ـغ	غ
ف	فـ	ـفـ	ـف	ف
قاف	قـ	ـقـ	ـق	ق
کاف	کـ	ـکـ	ـک	ک
گاف	گـ	ـگـ	ـگ	گ
لام	لـ	ـلـ	ـل	ل
میم	مـ	ـمـ	ـم	م
نون	نـ	ـنـ	ـن	ن
واو	و	ـو		
ه	هـ	ـهـ	ـه	ه
یا	یـ	ـیـ	ـی	ی

شکل ۱- حروف فارسی در انواع اشکال ممکن

۲– پیش‌زمینه

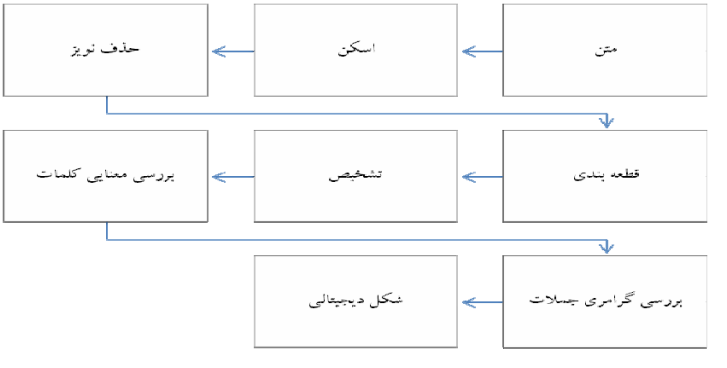
مراحل تشخیص کاراکترها در یک سیستم OCR شامل اسکن کردن مناسب تصویر و پردازش و آماده‌سازی آن برای تشخیص، سپس قطعه‌بندی آن به خطوط و کلمات و کاراکترها و بعد از آن تشخیص قطعات از هم تفکیک شده می‌باشد [۵]. در مرحله آخر می‌توان با بررسی پایگاه داده لغات به با معنی بودن کلمات بدست آمده و یا حتی بالاتر از آن به تصحیح از روی گرامر جملات زبان پرداخت [۶]. شکل (۲) خطای حاصله از هر کدام از مراحل بالا تأثیر مستقیم روی نتیجه نهائی خواهد داشت. لذا کار روی هرکدام از این مراحل خود مستلزم تحقیقات بسیاری می‌باشد. در این مقاله عمده تحقیق روی مرحله جداسازی و بهبود قطعه‌بندی کاراکترها انجام گرفته است.

۲–۱– کسب تصویر و ملایم‌سازی آن

اولین مرحله در شناسایی متن بدست آوردن تصویر دیجیتالی متن با استفاده از سیستم Scanning مناسب می‌باشد. وضوح ۴۰۰-۲۰۰ برای اغلب فونت‌های چاپی فارسی کافی تشخیص داده شده است [۷]. مرحله بعدی تبدیل تصویر به‌صورت باینری یا [۰، ۱] می‌باشد. این امر با مقایسه ارزش gray هر نقطه با یک آستانه معین تحقیق می‌یابد. جهت یافتن مقدار آستانه ابتدا ارزش gray همه نقاط در متن محاسبه می‌گردد و عددی که دارای بیشترین تکرار است به‌عنوان ارزش غالب در متن مشخص می‌گردد. این عدد عموماً معرف رنگ زمینه است. آنگاه ارزش آستانه با محاسبه معدل ارزش gray غالب و ماکزیمم gray بدست می‌آید. فرمول مربوط به محاسبه مورد نظر در زیر آمده است [۸].

Grayscale: = (0.299 * R) + (0.587 * G) + (0.114 * B)
if Grayscale > Threshold is 0 else is 1

چنانچه در مورد هر سیستم اکتساب داده‌ها صادق است، نویز در ورودی روی خواهد داد. برای حذف نویزها از تصویر متن نیاز به ملایم‌سازی (Smoothing) داریم. این کار می‌تواند از هر یک از روش‌های استاندارد و فراوانی که در زمینه پردازش تصویر موجود است انجام گیرد.



شکل ۲- مراحل OCR – تبدیل یک عکس نوشته به فرمت دیجیتالی

۲-۲- قطعه‌بندی کاراکترها (Segmentation)

قطعه‌بندی مرحله‌ای مهم است که در آن اجزاء تصویر متنی که باید به مرحله شناسایی تحویل داده شوند از یکدیگر جدا می‌شوند. هیچ روش عمومی برای حروف کلمات فارسی هنوز ابداع نشده است زیرا تفسیر هر حرف مورد نظر متقابلاً به حروف اطراف خود بستگی دارد. در بعضی روش‌های OCR با الزام کاربر به نوشتن در مرکز خانه تا حدودی قطعه‌بندی را میسر می‌سازند که به‌دلیل ویژگی متصل بودن حروف به یکدیگر، این کار برای متون فارسی عملی نیست [۹] و حروف ممکن است هم‌پوشانی کنند.

قطعه‌بندی در روش‌های قبل و همچنین مراحل اولیه روش جدید ارائه شده در مقاله، در سه سطح انجام می‌گیرد (شکل (۳)):

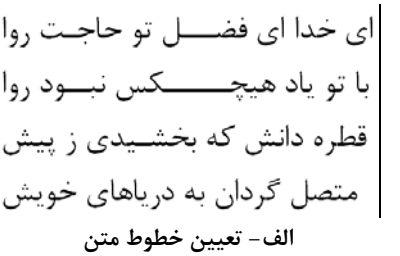
تشخیص خطوط

تشخیص کلمات یا زیر کلمات

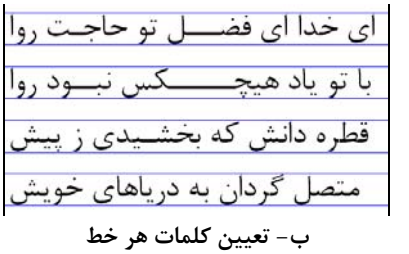
تشخیص کاراکترها

معمولاً خطوط با شکاف‌های افقی جدا می‌شوند و لذا چگالی افقی نقاط (تعداد ۱های موجود در هر ردیف) برای ردیابی خطوط متن انجام می‌گیرد. با فرض اینکه متن به خوبی تنظیم شده است این کار عملی می‌باشد. هرگاه تعداد سطرهای خالی از یک مقدار آستانه تجاوز کند یک خط تشخیص داده می‌شود.

بعد از جدا کردن خطوط متن، کلمات و کلمات فرعی درون خط با بازرسی شکاف‌های عمودی تعیین می‌گردد. ارزش متوسط آستانه‌ای محاسبه شده از تمام شکاف‌های عمودی برای تعیین این که آیا فاصله عمودی بین قسمت‌ها فاصله درون کلمه‌ای است یا بین دو کلمه مجزا است بکار می‌رود.



الف- تعیین خطوط متن



ب- تعیین کلمات هر خط

ج- تعیین کاراکترهای هرخط

شکل ۳- مراحل قطعه‌بندی یک عکس نوشته

برای جداسازی علائم (حروف زبان فارسی) روش‌های قبلی برپایه تعیین تکرار مرزهای حروف قرار دارد. در فارسی اغلب مرزها معمولا نزدیک به خط مبنا هستند.

در این قسمت به مراحل عمده قطعه‌بندی حروف در یکی از بهترین روش‌های موجود می‌پردازیم [۱۰]:

خط مبنا را برای کلمه (یا کلمه فرعی) معین تعیین کنید. یافتن خط مبنا مبتنی‌بر تحلیل چگالی افقی نقاط (horizontal density histogram) و تعیین بیشترین مقدار آن است.

یافتن برآمدگی حروف برروی کلمه را با جمع کردن پیکسل‌های حروف به‌صورت عمودی انجام دهید. یا به‌عبارتی هیستوگرام عمودی کلمه را بیابید.

کلمه را از راست به چپ پویش کنید و هر قسمت از کلمه را که دارای میزان برآمدگی کمتر از متوسط برآمدگی می‌باشد به‌عنوان مرزهای بالقوه حروف علامتگذاری کنید. این نقاط متناظر با خط زمینه می‌باشند.

مرزهای بالقوه را برای تعیین مرزهای بالفعل (واقعی) حروف بررسی کنید. مرزهای واقعی جایی هستند که قطعه‌بندی باید در آنجا انجام گیرد. هر مرز بالقوه باید درصورتی به مرز واقعی تبدیل شود که یکی از شرایط زیر را تأمین کند:

عرض هر حرف نباید بسیار کوچک باشد (بسیار کوچکتر ازمیانگین عرض حروف)

یک تغییر محسوس در هیستوگرام عمودی و در همسایگی مرزهای بالقوه صورت گرفته باشد.

اولین شرط لازم است زیرا بعضی حروف مانند "س" هم مرزهای بالقوه بدست می‌دهند و باید مرزهای دندانه‌ای آنها نادیده

گرفته و درحقیقت دندانه‌ها با هم ترکیب شوند. شرط دوم برای این وضع گشته است که حروف در اشکال آغازین و میانی، با ارزش بالایی از برآمدگی عمودی، دارای مرز عرض کوچکی می‌باشند. یعنی حرفی مانند الف دارای عرض کمی است که با شرط اول حذف خواهد شد. تفاوت این حروف با دندانه‌ها در طول زیاد آنها است.

بعد از تعیین مرز واقعی حروف منتهی‌الیه سمت چپ به‌عنوان نقطه‌ای انتخاب می‌گردد که قطعه‌بندی در آنجا صورت خواهد گرفت (شکل (۴)).

شکل ۴- تشخیص مرزهای بالقوه و بررسی شرائط برای تبدیل به مرزهای واقعی در مراحل جداسازی در یکی از بهترین روش‌های قبلی موجود

این روش، یک روش مؤثر برای قطعه‌بندی شناخته شده است که با تاکید بر جنبه‌های بصری و واقعی حروف فارسی و عربی به جداسازی آنها می‌پردازد و حدودا تا ۷۵٪ الی ۸۰٪ موفق بوده است.

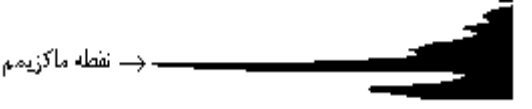
ما در ادامه مقاله سعی بر آن داریم که مشکلات جداسازی‌های فعلی را بررسی و با یک پردازش تصویر مناسب‌تر به بهبود این روش‌ها تا حدود ۹۸/۵٪ و تا ۱۰۰٪ در نوع فونت خاص برسانیم. در مورد بقیه فونت‌ها نیز با یک بررسی آماری جامع می‌توان به‌دقت مورد نظر دست یافت.

۳- الگوریتم ارائه شده

مراحل اولیه روش جدید همانند مراحل قبلی با جداسازی خطوط و کلمات آغاز می‌گردد. در این قسمت همچنان از همان روش‌های قبلی استفاده می‌گردد و نوآوری در قسمت مهم تشخیص کاراکترها انجام گرفته است.

اولین مرحله، تشخیص خط مبنا با بررسی هیستوگرام افقی نقاط تصویر می‌باشد [۱۱]. به این ترتیب که تعداد یک‌های هر ردیف را می‌شماریم و ردیفی که دارای بیشترین نقاط باشد به‌عنوان خط مبنا در نظر گرفته می‌شود (شکل (۵)).

$$H_i = \sum_{j=1}^{m} I(i,j) \;\; \forall i = 1..n$$



شکل ۵- بررسی هیستوگرام افقی نقاط جمله برای یافتن محدوده خط مبنا

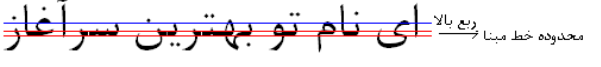
در روش‌های قبلی یک ردیف به‌عنوان خط مبنا شناخته می‌شد که با توجه به این واقعیت که خط زمینه در فارسی دارای عرض می‌باشد ما به بهبود تعریف خط مبنا از یک ردیف به یک محدوده پرداخته‌ایم. به این ترتیب که خط مبنا از بالا و پایین بسط داده می‌شود تا محدوده خط مبنا در خط بدست آید. این بسط بستگی به اندازه قلم دارد که در پیاده‌سازی تحقیقی روش به‌عنوان یک متغیر درنظرگرفته شده است. البته باید توجه داشت که در یک نرم‌افزار واقعی این مقدار باید از طریق یک بررسی آماری جامع و به‌صورت خودکار برای اندازه قلم مربوطه بدست آید (شکل (۶)).

$$BaseLine = \left(\max^{n}_{i=1} H_i\right) \pm \delta_b \qquad (۱)$$

شکل ۶- بسط خط مبنا برای یافتن محدوده عرض قلم

برای بررسی‌های بعدی همچنین مقدار هیستوگرام از خط مبنا به‌سمت بالا تا رسیدن به مقدار ۱/۴ مقدار مبنا پویش و به‌عنوان ربع بالا علامتگذاری می‌شود. این خط عموما ردیف بالای دندانه‌ها و حروف را بدست می‌دهد. مقدار ۱/۴ به‌صورت تجربه آماری بدست آمده است (شکل (۷)).

$$Q_{up} = i \; so \; H_i = \tfrac{1}{4}\left(\max(H_i)\right) \qquad (۲)$$



شکل ۷- پویش جمله برای یافتن ربع بالائی

از خط ربع بالا در بررسی‌های بعدی برای یافتن حروف بلند مانند الف و لام و ... و همچنین یافتن نقاط بالای جمله استفاده می‌شود.

در روش‌های قدیم بررسی عمودی هیستوگرام برای یافتن هرگونه برآمدگی از خط مبنا برای تشخیص حروف تنها معطوف به مقدار یک‌های هر ستون میگشت که این مسئله در مورد نقاط مشکل‌ساز بوده است. (شکل (۸)) به این معنا که ارتفاع عمودی نقطه به‌علاوه عرض عمودی خط مبنا از مقدار آستانه تجاوز کرده و به‌عنوان یک حرف مستقل درنظرگرفته می‌شد.

شکل ۸- مشکل تشخیص نقاط به عنوان حروف

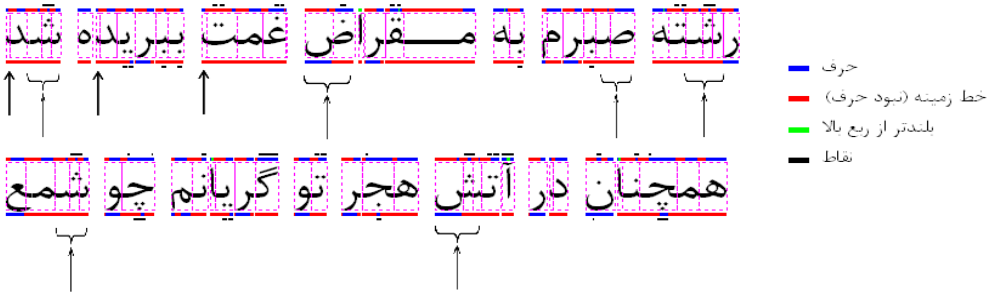
در این روش جدید با توجه به این واقعیت که کلیه حروف فارسی به خط زمینه پیوسته می‌باشند، برای یافتن برآمدگی‌ها تجاوز مقدار هیستوگرام عمودی از آستانه را با شرط پیوستگی به خط مبنا همراه کردیم تا از این مشکل جلوگیری کرده باشیم. به این ترتیب ابتدا یک‌های بالای خط مبنا بررسی و درصورت پیوستگی به خط مبنا به‌عنوان یک برآمدگی حرفی علامتگذاری می‌شوند و درغیراین‌صورت یک‌ها به‌عنوان نقاط جمله علامتگذاری می‌شوند. چون هر نقطه سیاه در یک کلمه فارسی به شرط پیوستگی با خط مبنا حرف و درغیراین‌صورت نقطه خواهد بود.

علامتگذاری نقاط جمله به‌ترتیبی که در بالا ذکر شد در خیلی از مراحلی که شرح آن خواهد آمد مانند تمایز ن از بقیه حروف گرد انتهائی یا یافتن دندانه‌های بدون نقطه سـ ، شـ ، صـ و ... که یکی از مراحل مهم بهبود روش جدید می‌باشد، کاربرد دارد.

مانند همین عمل را به‌طور مستقل و جداگانه برای قسمت زیر خط مبنا انجام داده و حروف و نقاط زیر خط نیز علامتگذاری می‌شوند.

به‌این‌ترتیب هر یک از ستون‌های تصویر به یکی از مقادیر حرف یا خط مبنا یا خالی علامتگذاری می‌شوند. همچنین هرستون یا حاوی نقاط بالای جمله یا حاوی نقاط پایین جمله و یا بدون نقطه می‌باشد. این مقداردهی پایه انجام کلیه مراحل بعدی به‌شمار می‌رود (شکل (۹)).

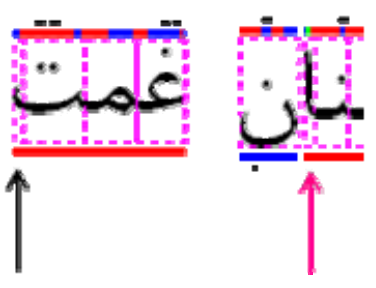
$$V_{ji} = \sum_{i=1}^{n} I(i,j)$$
$$if \;\; V_i > Q_{up} \;\; then \;\; Mark \; as \; So \; High$$
$$if \;\; V_i > BaseLine \pm \delta \;\; then \begin{cases} if \;\; Continues \;\; Mark \; as \; Char \\ if \;\; Continues \;\; Mark \; as \;\; Dot \end{cases} \qquad (۳)$$
$$else \;\; Mark \;\; as \;\; BaseLine$$



شکل ۹- علامتگذاری ستون‌های جمله به مقادیر حرف، خط مبنا، بلندتر از ربع بالا، تعیین نقاط و قسمت‌بندی اولیه حروف

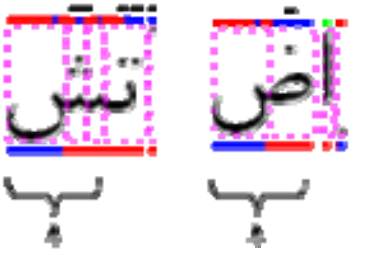
در مرحله بعدی ستون‌ها از راست به چپ پویش و منتهی‌الیه سمت چپ را که در مرحله قبل به‌عنوان خط مبنا علامتگذاری شده است با مرزهای بالقوه جدا می‌کنیم. همچنین رسیدن به یک ستون خالی نیز باعث ایجاد یک مرز واقعی برای حروف می‌شود (شکل (۹)). در مرحله بعد مرزهای نادرست تشخیص داده می‌شود و مرزهای درست باقیمانده به‌عنوان مرزهای واقعی علا متگذاری می‌شوند.

یکی از تشخیص‌های نادرست و شایع که باید اصلاح شود برآمدگی انتهایی برخی حروف مانند "ت ، ب ، ... د ، ..." است که به‌عنوان یک حرف مستقل علامتگذاری می‌شوند (شکل (۱۰)). برای رفع این مشکل کلیه حروف علامتگذاری شده که عرض آنها از مقدار میانگین عرض حروف کمتر است را می‌یابیم. از سمت چپ حرف تا راست آن را برای یافتن هیستوگرام عمودی یک‌های هر ستون البته از خط ربع بالا تا بالای خط را پویش و مقدار هر ستون را می‌یابیم. چنانچه در یکی از ستون‌ها مقدار هیستوگرام از نصف میانه بالایی بیشتر باشد یک مرز واقعی تشخیص داده می‌شود. دلیل این امر این است که عرض حرف الف در انتها و لام در وسط جمله نیز به اندازه خطاهای مطرح شده در بالاست وتنها تفاوت آنها در طول بلند این دوحرف و تجاوز این‌ها از خط میانه می‌باشد. بقیه این حروف با حرف قبلی ترکیب می‌شوند و مرزهای بالقوه آنها پاک می‌شود (باز نگاه کنید به شکل (۱۰)).



شکل ۱۰- تشخیص اشتباه دندانه‌های انتهای حروف

مشکل مهم بعدی حروف دایره‌ای در انتهای کلمات است مانند س، ش، ص و ... . در این مورد ابتدای حروف و دایره انتهای آنها به‌عنوان دو حرف مجزا تشخیص داده می‌شوند که باید ترکیب گردند (شکل (۱۱)).



شکل ۱۱- دایره‌های انتهائی حروفی مانند س، ص و ... که باید اصلاح شوند.

سه حرف ن و ل و ی از این قاعده مستثنی هستند. مشکل اصلی تشخیص دایره‌های انتهایی و تمایز آنها از این سه حرف است. دراین روش ما حروف انتهایی کلمات را بررسی می‌کنیم و چنانچه شرایط زیر را دارا باشند به‌عنوان حروف دایره‌ای شناخته می‌شوند:

در ابتدا و انتها در محدوده خط مبنا تصویر داشته باشیم.

اواسط حروف در بالای خط مبنا به‌عنوان خالی علامتگذاری شده باشند.

اواسط حروف در زیر خط مبنا به‌عنوان حرف به‌طور کامل علامتگذاری شده باشد.

در اواسط حروف (به‌جز ابتدا و انتها) محدوده خط مبنا سفید باشد.

در محدوده حرف جمله حاوی نقطه نباشد.

حروف با طول بالاتر از خط میانه ظاهر نشوند.

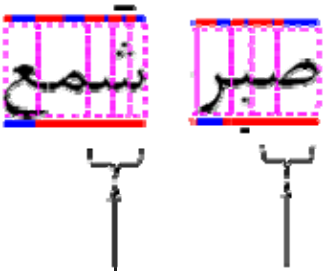
شرط آخر برای اطمینان از تمایز حرف ل از دایره‌ها قرار داده شده است. طرز تشخیص این بلندی‌ها مانند مرحله قبلی برای تشخیص حرف الف می‌باشد. (با استفاده از خط ربع بالا) شرط پنجم نیز برای کنار گذاشتن حرف ن از این مجموعه می‌باشد. همچنین شرط چهارم نیز برای تمایز حرف ی قرار داده شده است.

برای ترکیب این دایره‌ها با حروف مجازی قبل از آنها و از بین بردن مرزهای نادرست دو انتخاب پیش رو داریم:

عرض حرف مجازی قبلی از میانگین عرض حروف بیشتر است که در این صورت حروفی مانند ص ظاهر شده‌اند که کافی است ترکیب با یک حرف قبل صورت گیرد.

عرض حرف مجازی قبلی از میانگین عرض حروف کمتر است که در این صورت دندانه داریم و حروفی مانند س ظاهر شده‌اند که کافی است ترکیب با دو حرف قبل که نمایانگر دو دندانه این حروف است صورت گیرد.

آخرین مرحله بررسی و یافتن و ترکیب دندانه‌های حروف سـ ، شـ ، صـ و ضـ در اواسط جمله است (شکل (۱۲)). یکی از مهمترین بررسی‌هایی که در این روش صورت گرفته و به نوعی بدعت به‌شمار می‌رود این مطلب است که برای پرهیز از ترکیب‌های اشتباه این بررسی باید حتماً از انتهای جمله به ابتدا صورت گیرد. این مسئله هم با مداقه در نوع جانشینی حروف زبان فارسی بدست آمده است.



شکل ۱۲- دندانه‌های حروفی مانند سـ صـ و ... که باید اصلاح شوند.

برای رفع این مشکل، ابتدا باید به یک دندانه بدون نقطه برسیم و از آنجایی که هیچ حرف دندانه‌داری در زبان فارسی بدون نقطه یافت نمی‌شود، پس یک حدس اشتباه رخ داده که باید تصحیح شود. طرز تشخیص این دندانه‌ها نیز از روی عرض این حروف می‌باشد. چنانچه عرض حرف از عرض میانگین کمتر باشد یک دندانه خواهیم داشت. با بررسی علامتگذاری انجام شده در مورد نقاط جملات متوجه خواهیم شد که این دندانه بدون نقطه است یا نه؟

بعد از یافتن این دندانه‌ها دو حالت در پیش رو داریم که به بررسی این دو حالت می‌پردازیم:

عرض حرف مجازی قبلی از میانگین عرض حروف بیشتر است که در این صورت حروفی مانند صـ ظاهر شده‌اند که کافی است ترکیب با یک حرف قبل صورت گیرد.

عرض حرف مجازی قبلی از میانگین عرض حروف کمتر است و حرف قبلی دارای نقطه زیر خط مبنا نمی‌باشد که در این صورت حروفی مانند سـ یا شـ ظاهر شده‌اند که کافی است ترکیب با دو حرف قبل که نمایانگر دو دندانه این حروف است صورت گیرد.

بعد از اعمال مراحل فوق حروف تا حدود ٪۹۸/۵ تا ٪۹۹ قطعه‌بندی شده اند.

در این مرحله متن آماده تشخیص می‌باشد که یک روش تشخیص خوب مانند یادگیری الگو یا شبکه‌های عصبی به‌همراه یک پایگاه داده لغات جامع می‌تواند راه‌گشا باشد.

۴– پیاده‌سازی و نتایج آزمایشات

الگوریتم فوق با استفاده از زبان برنامه‌نویسی دلفی به‌صورت کامل پیاده‌سازی و بر روی کامپیوتر پنتیوم ۴ با اندازه رم ۲ گیگابایت تست شده است. برای انجام آزمایشات از ۵۰ صفحه تایپ شده فارسی با فونت‌ها واندازه‌های مختلف استفاده شده است.

جدول (۱) میزان صحت قطعه‌بندی الگوریتم را برای فونت‌های مختلف در اندازه‌های مختلف نشان می‌دهد. بررسی‌های انجام گرفته برای دقت بیشتر هرکدام به تعداد ۵ بار تکرار شده است.

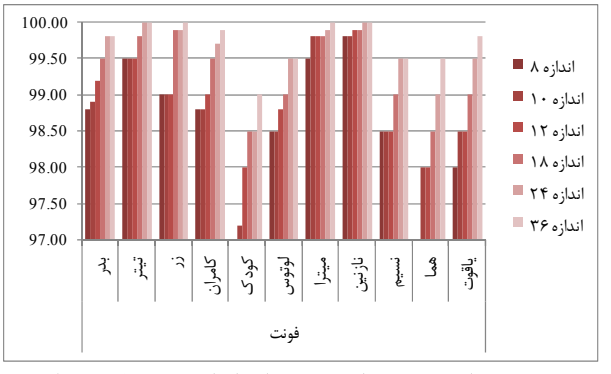
جدول ۱- میزان صحت الگوریتم برای فونت‌های مختلف

فونت	اندازه						میانگین
	۸	۱۰	۱۲	۱۸	۲۴	۳۶	
بدر	۹۸/۸۰	۹۸/۹۰	۹۹/۲۰	۹۹/۵۰	۹۹/۸۰	۹۹/۸۰	۹۹/۳۳
تیتر	۹۹/۵۰	۹۹/۵۰	۹۹/۵۰	۹۹/۸۰	۱۰۰	۱۰۰	۹۹/۷۲
زر	۹۹/۰۰	۹۹/۰۰	۹۹/۰۰	۹۹/۹۰	۹۹/۹۰	۱۰۰	۹۹/۴۷
کامران	۹۸/۸۰	۹۸/۸۰	۹۹/۰۰	۹۹/۵۰	۹۹/۷۰	۹۹/۹۰	۹۹/۲۸
کودک	۹۷/۰۰	۹۷/۲۰	۹۸/۰۰	۹۸/۵۰	۹۸/۵۰	۹۹/۰۰	۹۹/۰۳
لوتوس	۹۸/۵۰	۹۸/۵۰	۹۸/۸۰	۹۹/۰۰	۹۹/۵۰	۹۹/۵۰	۹۸/۹۷
میترا	۹۹/۵۰	۹۹/۸۰	۹۹/۸۰	۹۹/۸۰	۹۹/۹۰	۱۰۰	۹۹/۸۰
نازنین	۹۹/۸۰	۹۹/۸۰	۹۹/۹۰	۹۹/۹۰	۱۰۰	۱۰۰	۹۹/۹۰
نسیم	۹۸/۵۰	۹۸/۵۰	۹۸/۵۰	۹۹/۰۰	۹۹/۵۰	۹۹/۵۰	۹۸/۹۲
هما	۹۷/۰۰	۹۸/۰۰	۹۸/۰۰	۹۸/۵۰	۹۹/۰۰	۹۹/۵۰	۹۸/۳۳
یاقوت	۹۸/۰۰	۹۸/۵۰	۹۸/۵۰	۹۹/۰۰	۹۹/۵۰	۹۹/۸۰	۹۸/۸۸

نتایج بررسی‌ها به‌خوبی عملکرد بالای الگوریتم را نمایان می‌نماید. به‌خصوص الگوریتم در مورد فونت‌های چاپی کتابی مانند زر، نازنین و یا حتی میترا عملکرد بسیار بالایی را از خود نشان می‌دهد که این امر برای فونت‌های مزبور نزدیک به ۱۰۰٪ می‌باشد. کمترین کارآیی الگوریتم مربوط به فونت‌هایی است که بنا به شمایل گرافیکی آنها دارای انحراف یا انحناهای بیش از حد می‌باشند. وجود همپوشانی‌های بسیار زیاد و انحنای خط زمینه تا حدودی بر عملکرد الگوریتم تأثیر داشته است. نتایج میانگین صحت اجرای الگوریتم برای فونت‌های مختلف در شکل (۱۳) آمده است.

همچنین با بررسی نتایج بدست آمده می‌توان دریافت که نسبت کارآیی الگوریتم با بالا رفتن اندازه فونت‌ها رشد بسیار چشمگیری از خود نشان می‌دهد. این امر ناشی از افزایش اندازه دندانه‌ها و حروف بلند است. البته برای افزایش میزان قطعه‌بندی در فونت‌های با اندازه کوچک‌تر، می‌توان از تغییر دستی میزان آستانه‌های مورد استفاده در الگوریتم بهره جست. از سوی دیگر استفاده از روش‌های بزرگنمایی مناسب و تکنیک‌های پردازش تصویر برای افزایش اندازه فونت‌ها مفید به فایده است. شکل (۱۴) نمایانگر میزان قطعه‌بندی با اندازه فونت‌های مختلف می‌باشد.

شکل ۱۳- درصد میانگین صحت قطعه‌بندی برای فونت‌های مختلف



شکل ۱۴- میزان صحت قطعه‌بندی برای اندازه فونت‌های مختلف

۵– نتیجه‌گیری

هدف اصلی این مقاله ارائه یک سیستم جداسازی متن فارسی چاپی است. فرآیند قطعه‌بندی در سه مرحله مجزای تقسیم متن به خطوط و تقسیم خطوط به کلمات و جداسازی کلمات به حروف مستقل انجام می‌گیرد. دو مرحله اول از روش‌های قبلی اقتباس شده است و نوآوری در زمینه جداسازی کلمات به حروف مستقل انجام شده است.

متأسفانه به‌دلیل برخی خصوصیات متن فارسی فرایند جداسازی حروف با مشکل روبرو است. به‌طوریکه هنوز روش خاص وکاملی برای آن ذکر نشده است. از آنجا که مهمترین و پایه اصلی تشخیص حروف به کاراکترهای اسکی فرایند قطعه‌بندی حروف است لذا الگوریتم پیشنهادی به بررسی خطاها و تحقیق در اصول بصری حروف زبان فارسی پرداخته و پیشنهادهای مؤثری را برای رفع

خطاها تا حدود ۹۹٪ انجام داده است.

بعد از تشخیص خطوط مبنا و ربع بالا به بررسی برآمدگی‌های پیوسته و نقاط جمله می‌پردازیم. فرایند جداسازی براساس فاصله‌های خالی و ظاهر شدن خطوط زمینه به ساخت مرزهای مجازی می‌پردازد. بعد از این مرحله نقاط مرزی بررسی تا مرزهای واقعی تشخیص داده شوند. از جمله این بررسی‌های بعد از جداسازی می‌توان به از بین بردن لبه‌های انتهایی حروفی مانند ت، ک، ... و همچنین بررسی و تشخیص دایره‌های اشتباه در آخر کلمات مانند س، ص، ... و همچنین ترکیب دندانه‌های بدون نقطه با حروف مجاور مانند ـسـ ، صـ ، ... اشاره نمود.

۶– کارهای آینده

نتایج آمارگیری آمار خوبی در حدود۹۸/۵٪ تا ۹۹٪ را با استفاده از این الگوریتم نشان می‌دهد. در پیاده‌سازی تحقیقی، برنامه نوع فونت خاص نازنین با اندازه ۳۶ را تا ۱۰۰٪ و به‌طور کامل جداسازی می‌نماید. مهمترین مسئله روش یافتن عرض خط مبنا و عرض حداقل کاراکترها برای هر فونت خاص است که در برنامه به‌عنوان متغیر قابل مقداردهی در نظر گرفته شده است. یک تحقیق جامع برای یافتن این اعداد در مورد دیگر فونت‌ها نیاز است تا به یک جداسازی ۱۰۰٪ نائل شویم. در مورد اندازه فونت هم‌نوع نمونه‌گیری دقیق‌تر و یا حتی بزرگ کردن تصویر با یک الگوریتم پردازش تصویر مناسب و دقت خوب می‌تواند کارساز باشد. البته هنوز ممکن است حروفی درست جدا نشوند. یکی از مشکلات حل نشده این روش ترکیب "لا" است. چون در بیشتر فونت‌ها حرف لان و الف در یکدیگر ترکیب می‌شوند و قابل جداسازی نیستند. این مشکل با اضافه کردن سمبل مر بوطه به مجموعه کاراکترهای زبان قابل حل شدن است. که خوشبختانه در مجموعه کاراکترهای کامپیوتر این سمبل‌ها در نظر گرفته شده است.

البته هنوز جای کار بسیاری وجود دارد. همچنین با استفاده از یک الگوریتم تشخیص مناسب به‌همراه یک گنجینه لغات کامل می‌توان خطاهای باقیمانده را نیز از بین برد.

۶– مراجع

[1] A. Aburas, Abdurazzag, M.A. Rehiel Salem; **“Off-Line Omni-Style Handwriting Arabic Character Recognition System Based on Wavelet Compression”** Arab Research Institute in Sciences & Engineering 3, No. 4, pp.123 - 135, 2007.

[2] H. AlKhateeb Jawad, R. Jinchang, S. Ipson Stanley, J. Jianmin. **“Knowledge-Based Baseline Detection and Optimal Thresholding for Words Segmentation in Efficient Pre-Processing of Handwritten Arabic Text”** In Third International Conference on Information Technology: New Generations, ed. Ren Jinchang, S. Ipson Stan and Jiang Jianmin, pp. 1158-1159, 2008.

[3] H. Al-rashaideh; **“Preprocessing Phase for Arabic Word Handwritten Recognition”** Russian Academy of Sciences 6, No. 1, pp. 11-19, 2006.

[4] A.H. Hassin, X.L Tang, J.F Liu, W. Zhao; **“Printed Arabic Character Recognition Using Hmm”**, J. Comput. Sci. Technol. 19, No. 4, pp. 538-543, 2004.

[5] V. Märgner, E.A Haikal; **“Databases and Competitions: Strategies to Improve Arabic Recognition Systems”,** In Arabic and Chinese Handwriting Recognition, pp. 82-103, 2008.

[6] P. Mario, V. Maergne. **“Hmm Based Approach for Handwritten Arabic Word Recognition Using the Ifn/Enit- Database”**, In Proceedings of the Seventh International Conference on Document Analysis and Recognition, Vol. 2: IEEE Computer Society, 2003.

[7] S.N. Nawaz., M. Sarfraz, A. Zidouri, W.G. Al-Khatib. **“An Approach to Offline Arabic Character Recognition Using Neural Networks”**, In 10th IEEE International Conference on Electronics, Circuits and Systems, 3, pp. 1328 - 1331, 2003.

[8] M.L Liana, V. Govindaraju; **“Offline Arabic Handwriting Recognition: A Survey”**, IEEE Trans. Pattern Anal. Mach. Intell. 28, No. 5, pp. 712-724, 2006.

[9] F. Lotfi, F. Nadir, B. Mouldi; **“Arabic Words Recognition by Fuzzy Classifier”** J. Applied Sci. 6, pp. 647-650, 2006.

[10] M. S. Khorsheed; **“Off-Line Arabic Character Recognition – a Review”** Pattern Analysis & Applications 5, No. 1, pp. 31 - 45, 2002.

[11] F. Faisal, V. Govindaraju, M. Perrone; **“Pre-Processing Methods for Handwritten Arabic Documents”** In Proceedings of the Eighth International Conference on Document Analysis and Recognition: IEEE Computer Society, 2005.