
بسم الله الرحمن الرحیم

**دانشکده مهندسی کامپیوتر**

**پایان‌نامه کارشناسی ارشد هوش مصنوعی و رباتیک**

# آنالیز قالب‌بندی اسناد فارسی

نگارنده: امیررضا فاتح

**استاد راهنما**

دکتر محسن رضوانی

**استاد مشاور**

دکتر علیرضا تجری

شهریور ۱۴۰۰

در این صفحه صورت جلسه دفاع را قرار دهید. لازم است پس از صحافی این صفحه مجدداً توسط دانشکده مهر گردد و استاد راهنما با امضای خود اصلاحات پایان‌نامه را تایید کند.

تقدیم اثر به ساحت مقدس امام زمان عجل الله تعالی فرجه الشریف، آیت رب ودود

و ناجی ملک وجود، آنکه جهانی در انتظار عدالت و لطف اوست

و به آنان که مهر آسمانی شان آرام بخش آلام زمینی ام است

به استوارترین تکیه گاهم، دستان پر مهر پدرم

به سبزترین نگاه زندگیم، چشمان سبز مادرم

که هرچه آموختم در مکتب عشق شما آموختم و هرچه بکوشم قطره ای از دریای بی کران مهربانیتان را سپاس نتوانم بگویم.

این هم به برادرانم:

به همسفران مهربان زندگیم مسعود، منصور و محمود عزیز

که با آنها آغاز کردم، در کنارشان آموختم. قلبم لبریز از عشق به شماست و خوشبختی تان منتهای آرزویم.

به مصداق «من لم یشکر المخلوق لم یشکر الخالق» بسی شایسته است از استادان

فرهیخته و فرزانه‌ام جناب آقای دکتر رضوانی، جناب آقای دکتر تجری و همچنین جناب آقای دکتر فلاح که همواره در طول

تحصیل متحمل زحماتم بوده‌اند و تکیه‌گاه من در مواجهه با مشکلات، و وجودشان مایه دلگرمی من می‌باشد تشکر و قدردانی کنم.

که با کرامتی چون خورشید، سرزمین دل را روشنی بخشیدند و گلشن سرای علم و

دانش را با راهنمایی های کارساز و سازنده بارور ساختند؛ تقدیر و تشکر نمایم.

«ویزکیهم ویعلمهم الکتاب والحکمه»

# تعهدنامه

اینجانب **امیررضا فاتح** دانشجوی دوره کارشناسی ارشد رشته **هوش مصنوعی و رباتیک** دانشکده کامپیوتر دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **آنالیز قالب‌بندی اسناد فارسی** تحت راهنمائی **دکتر محسن رضوانی** متعهد می‌شوم.

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

**تاریخ**

**امضای دانشجو**

# چکیده

آنالیز قالب‌بندی اسناد یکی از گام‌های کلیدی در روند تبدیل تصویر هر سند به متن قابل ویرایش است. جداسازی مناطق متنی و غیرمتنی درون تصویر یکی از تاثیرگذارترین پیش‌پردازش‌های ممکن در سیستم‌های نویسه‌خوان نوری است. از دیگر پیش‌پردازش‌های مهم در این سیستم‌ها، استخراج خطوط حاوی متن از درون مناطق متنی است. عدم تشخیص درست مناطق حاوی متن و به‌تبع آن عدم تشخیص صحیح مختصات خطوط، تمامی مراحل بعدی یک سیستم نویسه‌خوان نوری را دچار اختلال می‌کند. مشکلاتی همچون انحنای خطوط، کج بودن تصویر، وجود اعراب و نقاط زیاد در زبان فارسی و عربی، نزدیک بودن خطوط درون تصویر و وجود تصاویر چند ستونه از چالش‌های مهم در آنالیز قالب‌بندی اسناد است. در این تحقیق به تمام این چالش‌ها توجه شده است و سعی در رفع آن‌ها شده است تا روشی نوین برای آنالیز قالب‌بندی اسناد فارسی ارائه شود. روش پیشنهادی، در گام اول، مناطق متنی را از مناطق غیرمتنی جدا می‌سازد. برای این کار، از چندین روش مختلف و کارآمد مبتنی بر یادگیری عمیق بهره گرفته شده و با استفاده از سیستم رای‌گیری در میان آن‌ها، محتمل‌ترین مناطق متنی تصویر استخراج می‌شود. در گام دوم، می‌بایست خطوط حاوی متن از درون مناطق متنی استخراج شوند. در این گام، روش پیشنهادی با بکارگیری فرآیندی بر پایه اندازه قلم متن، خطوط حاوی متن استخراج می‌شود. تا کنون روش استخراج خطوط بر مبنای اندازه قلم، ارائه نشده است. روش پیشنهادی بر روی مجموعه دادگانی از تصاویر با بیش از ۲۰۰۰ صفحه از تصاویر اسکن شده آزمون شده است که در دو بخش تشخیص مناطق حاوی متن و استخراج خطوط به ترتیب به دقت‌های ۹۸.۰۴٪ و ۹۹.۴۲٪ رسیده است.

**کلمات کلیدی:** آنالیز قالب‌بندی سند، تشخیص خطوط، تقسیم‌بندی متون، تقسیم‌بندی تصاویر، اندازه قلم، رای‌گیری

# لیست مقالات مستخرج از پایان نامه

1- **Persian Printed Text Line Detection Based on Font Size, multimedia tools and applications (Revised)**

**۲- ارائه روشی مبتنی بر رای گیری برای ترکیب خروجی های شبکه های عمیق جهت آنالیز قالب بندی اسناد چاپی، مجله ماشین بینایی و پردازش تصویر (پذیرفته شده)**

**۳- تشخیص خطوط حاوی متن در تصاویر اسناد چاپی فارسی، کنفرانس پردازش سیگنال و سیستمهای هوشمند دی ۱۳۹۹**

# فهرست مطالب

# فهرست جداول

# فهرست اشکال

# فصل۱ : مقدمه

## ۱-۱ مقدمه

امروزه از هوش‌مصنوعی و شاخه‌های آن در حوزه‌های مختلفی استفاده می‌شود. یکی از مهم‌ترین شاخه‌هایی که می‌توان از آن نام برد، پردازش تصویر و بینایی ماشین است. دامنه مطالعاتی پردازش تصویر بسیار وسیع است. از جمله مهم‌ترین حوزه‌های مورد استفاده، می‌توان به استفاده از آن در علوم پزشکی، صنعت و کاربردهای نظامی اشاره نمود. یکی از مهم‌ترین حوزه‌هایی که پردازش تصویر نقش پررنگی تاکنون در آن داشته و مطالعات بسیاری در آن زمینه صورت گرفته شده است، تشخیص متن درون تصویر است. تشخیص متن درون تصویر، یکی از موضوع‌های مهم و چالش برانگیز در بخش شناسایی الگو و هوش مصنوعی است [۱, ۲]. بسیاری از سازمان‌ها در تلاش‌اند تا اسناد بخش بایگانی خود را به صورت دیجیتال ذخیره نمایند. برای این منظور نیاز است که تمامی این اسناد توسط یک یا چند نفر در رایانه بازنویسی و در پایگاه داده آن سازمان ذخیره شود. راه آسان‌تر استفاده از سیستم‌های صفحه‌خوان[1] در حوزه هوش مصنوعی است که ابتدا باید از تک تک صفحات یک سند، عکس گرفته شود و سپس با دادن تصویر یک سند به این سیستم‌ها، متن درون آن استخراج شود. همچنین بسیاری از اسناد و کتاب‌ها، در فرمت‌هایی مانند jpeg، jpg، png، tiff و غیره در دسترس هستند. یکی از مشکلات اساسی فرمت تصویری اسناد، عدم امکان جستجو و ویرایش کلمات در آن‌ها است. از این رو، استفاده از سیستم‌های تشخیص متن اجتناب‌ناپذیر است و نیاز است که متن درون تصویر استخراج و به یک متن قابل ویرایش تبدیل شود [۳].

در یک سیستم صفحه‌خوان، تصویر یک سند به عنوان ورودی گرفته می‌شود که این تصویر می‌تواند توسط اسکنر، دوربین و یا نرم‌افزارهای خاص تولید شده باشد. این سیستم‌ها از طریق یک فرآیند شناخته‌شده به‌نام "نویسه‌خوان نوری (OCR)[2]"، هر تصویر از سند را تجزیه و تحلیل کرده و تلاش می‌کند تا حروف

[1] Page Reading
[2] Optimal Character Recognition

چاپ شده روی صفحه را تشخیص دهند. در نهایت یک فایل متنی، حاوی اعداد و حروف تشخیص داده شده از تصویر سند در خروجی سیستم قرار می‌گیرد [۴].

آنالیز قالب‌بندی سند (DLA)[1]، یکی از مراحل مهم در زمینه تبدیل اسناد به متون دیجیتال قابل جستجو است [۵]. آن ناحیه‌ای از سند که به سیستم OCR داده می‌شود باید تنها حاوی متن باشد، لذا می‌بایست آن ناحیه را از تصویر اصلی جدا نمود. عدم جداسازی متن از تصویر و عدم تشخیص صحیح خطوط و پارگراف‌ها، دقت سیستم OCR را کاهش می‌دهد، لذا DLA یک مرحله بسیار مهم در سیستم OCR است [۶].

## ۱-۲ بیان مسئله

DLA یک روش برای تعریف و تشخیص ساختار سند و به عبارت دیگر یک راه برای بخش‌بندی مناطق و نواحی مهم هر سند مخصوصا با قالب‌های پیچیده است. DLA از این رو حائز اهمیت است که دقت یک سیستم OCR در مواجهه با قالب‌های پیچیده پایین می‌آید. این پیچیدگی به سبب داشتن چندین ستون، پاراگراف، شکل، جدول و گراف در تصویر ایجاد می‌شود [۷].

در تصویری که به عنوان ورودی به سیستم OCR داده می‌شود، ابتدا باید مناطق متنی و غیرمتنی مشخص شوند. سپس DLA خطوط حاوی متن در هر یک از مناطق متنی را استخراج می‌کند. در سیستم OCR، خطوط استخراج شده به صورت جداگانه به عنوان ورودی به مرحله بعدی داده می‌شوند تا در نهایت متن درون این خطوط استخراج شود. استخراج دقیق و درست خطوط، دقت سیستم OCR و بهره‌وری از آن را بالا می‌برد.

[1] Document Layout Analysis

خواندن پلاک خودرو [۸-۱۱]، خواندن شماره کنتور گاز [۱۲]، کمک به نابینایان [۱۳]، پردازش چک‌های بانکی [۱۴] و مرتب‌سازی خودکار[1] [۱۵] از جمله مهم‌ترین برنامه‌های کاربردی هستند که DLA در آنها نقش مهمی را ایفا می‌کند.

## ۱-۳ چالش‌های پژوهشی

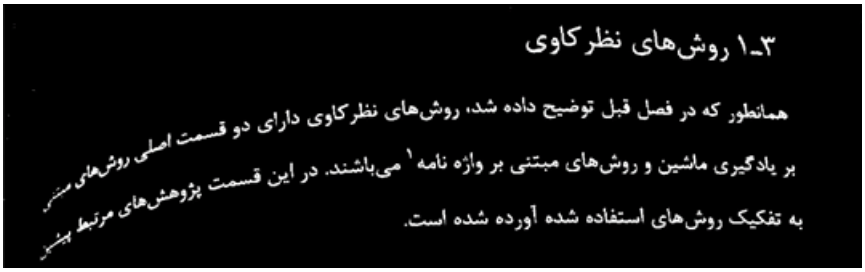
چالش‌های موجود در سیستم‌های آنالیز قالب‌بندی اسناد به دو بخش تقسیم می‌شوند: اول چالش‌های موجود در زمینه تشخیص مناطق حاوی متن و دوم چالش‌های موجود در زمینه استخراج خطوط حاوی متن از درون مناطق متنی است. مهم‌ترین چالش در زمینه تشخیص مناطق حاوی متن، نبودن یک قالب یکسان در تمامی صفحات است. بدین معنا که ممکن است در یک صفحه هیچ پیچیدگی خاصی از نظیر شکل، جدول یا گراف وجود نداشته باشد ولی در صفحه دیگر این پیچیدگی‌ها موجود باشد [۶]. بعلاوه وجود پس‌زمینه‌های پیچیده، نویزهای مختلف، کیفیت پایین و چرخش تصاویر در تشخیص و استخراج مناطق حاوی متن موثر است.

یکی از نیازهای مهم و اساسی در طراحی یک الگوریتم در زمینه تشخیص خطوط حاوی متن، دقت بالاتر همراه با محاسبات کم است [۱۶]. اما دلایلی مثل تاری تصاویر، گرفتن تصویر توسط دوربین با وضوح پایین، نور فلاش دوربین موبایل و کج گذاشتن صفحات در اسکنر مانع دستیابی الگوریتم به دقت‌های بالا می‌شود [۱۷, ۱۸].

از دیگر چالش‌هایی که در زمینه تشخیص خطوط حاوی متن با آن روبرو هستیم، می‌توان به تصاویری با چندین ستون، اشاره کرد [۱۹]. زمانی که بیش از یک ستون درون تصویر وجود داشته باشد، الگوریتم استخراج خط ممکن است که دو خط از دو ستون کنار هم را به یکدیگر بچسباند و آنها را یک خط در نظر بگیرد.

[1] Automatic Sorting

یکی دیگر از مشکلات اساسی در این حوزه، وجود خطوطی است که دارای انحنا هستند. همانطور که در شکل ۱-۱ نشان داده شده است، زمانی که از یک صفحه کتاب عکس گرفته می‌شود، انتهای خطوط حالت منحنی شکل به خود می‌گیرند. این حالت منحنی در خطوط، استخراج را سخت‌تر و دقت سیستم OCR را کاهش می‌دهد [۲۰].




شکل ۱-۱: تصویر دارای خطوط انحنا

افزایش روزافزون فایل‌های غیر قابل ویرایش مانند PDF و فرمت‌های تصویری مانند JPG، کاربرد سیستم-های OCR و به تبع آن سیستم‌های آنالیز قالب‌بندی را به صورت قابل توجهی افزایش داده است. فرآیند تشخیص در این سیستم‌ها به صورت خودکار و بدون دخالت انسان انجام می‌شود. لذا این ویژگی باعث کاربردی‌تر شدن این سیستم‌ها شده است.

پیچیدگی فرآیند تشخیص در زبان‌های گوناگون، متفاوت است. در زبان‌هایی مانند فارسی و عربی حروف به دو نوع چسبان و جدا تقسیم می‌شوند که کار تشخیص خط را نسبت به زبان‌هایی مانند انگلیسی سخت‌تر می‌کند. بعلاوه در این زبان‌ها ما با چالش‌هایی مثل وجود اعراب یا همان حرکت در بالا و پایین بعضی از کلمات روبرو هستیم که تشخیص صحیح خطوط را مشکل می‌سازد [۱۹]. به‌طور مشابه وجود نقاط زیاد در بالا و پایین کلمات و امکان تشخیص ندادن آنها به عنوان بخشی از یک خط، باعث کاهش دقت یک سیستم DLA می‌شود. در سال‌های اخیر روش‌های زیادی در حوزه DLA برای زبان‌هایی مانند انگلیسی، آلمانی، فرانسه، ایتالیایی و چینی ارائه شده است. اما برای زبان‌هایی مانند فارسی جای کار هنوز بسیار است [۲۱].

## ۱-٤ اهداف پایان‌نامه

همان‌طور که در بخش قبل گفته شد، تاکنون مطالعات پژوهشی متنوعی در زمینه تشخیص خطوط متون تایپی و دستنویس انجام شده است. هر یک از این مطالعات، از الگوریتم‌های مختلفی برای دستیابی به هدف مشخص استفاده نموده‌اند. اما هیچکدام از روش‌های ارائه‌شده کامل و بی‌نقص نیستند و هنوز چندین چالش حل نشده در سیستم‌های DLA مخصوصا برای زبان فارسی وجود دارد. هم‌پوشانی مناطق متنی و غیرمتنی، کج بودن و داشتن انحنا خطوط حاوی متن از چالش‌های حل نشده در این کار است. اما مهم‌ترین چالش موجود، عدم دقت و کارایی جامع برای زبان‌های مختلف است. هدف این پایان‌نامه ارائه‌ی راهکاری هوشمند و دقیق برای رفع چالش‌های ذکر شده، به همراه افزایش دقت تشخیص در این سیستم‌ها است.

## ۱-٥ روش تحقیق

در این تحقیق سعی بر آن است که با استفاده از یک مجموعه دادگان از متون فارسی تایپی اسکن شده، روش جدیدی در ارتباط با آنالیز قالب‌بندی سند ارائه شود. بر این اساس، این پایان‌نامه، به دو مرحله تشخیص مناطق حاوی متن و استخراج خطوط از این نواحی تقسیم می‌شود. در مرحله اول، با بکارگیری چند الگوریتم تشخیص مناطق حاوی متن که همگی مبتنی بر یادگیری عمیق هستند و استفاده از سیستم رای‌گیری بین آنها، محتمل‌ترین مناطق متنی استخراج می‌شود.

در مرحله دوم، با استفاده از یک راهکار جدید برای استخراج خطوط حاوی متن، این خطوط استخراج شده تا به عنوان ورودی به سیستم OCR داده شوند. این روش پس از باینری کردن تصویر ورودی که می‌تواند سه کاناله یا تک کاناله باشد، نقاط زائد و نویزهای فلفل نمکی را از درون تصویر حذف می‌نماید. در مرحله بعدی پهنای هر یک از اجزای متصل (CC)[1] درون تصویر را محاسبه می‌کند. سپس با استفاده از تعریف مفهومی به‌نام اندازه قلم، برای هر تصویر یک اندازه قلم استخراج می‌کند. با بکارگیری از این اندازه قلم

[1] Connected Component

استخراج شده از تصویر، شعاع جستجویی تعریف می‌شود و با آن CCهایی که به صورت افقی در نزدیکی هم هستند را به یکدیگر متصل می‌سازد تا در نهایت یک خط استخراج شود.

## ۱-۶ نوآوری پایان‌نامه

در این پایان‌نامه یک روش جدید برای آنالیز قالب‌بندی در متون فارسی ارائه شده است. خلاصه‌ای از نوآوری‌های این پایان‌نامه در ادامه آورده شده است.

۱ استفاده از تکنیکی که از چندین روش یادگیری عمیق بهره می گیرد.

۲ استفاده از روشی نوین در رای‌گیری بین روش‌های مختلف تشخیص متن.

۳ استفاده از اندازه قلم برای استخراج خطوط.

٤ تمرکز روی خطوط دارای انحنا

٥ افزایش دقت آنالیز قالب‌بندی و استخراج خطوط در مقایسه با روشهای قبلی وموارد دیگر

## ۱-۷ ساختار پایان‌نامه

این پایان نامه شامل پنج فصل است. در ادامه و در فصل دوم، کارهای پیشین مورد بحث و بررسی قرار خواهد گرفت. در فصل سوم جزئیات روش پیشنهادی ارائه شده است. فصل چهارم به بیان نتایج ارزیابی روش پیشنهادی می‌پردازد. فصل پایانی به بیان یک جمع‌بندی کلی از موضوع، به همراه فعالیت‌هایی که در آینده می‌توانند در این زمینه انجام گیرند، می‌پردازد.

# فصل۲ : پیشینه تحقیق

## ۲-۱ مقدمه

با پیشرفت علم، کارایی سیستم‌های هوش‌مصنوعی بیشتر شده است ولی به دلیل استفاده روز افزون از این سیستم‌ها، نیازمندی‌های این سیستم‌ها نیز هر روز در حال افزایش است. لذا این سیستم‌ها باید به‌طور مداوم بهبود یابند تا قادر به مرتفع‌سازی نیاز کاربران باشند. سیستم‌های DLA و به تبع آن، سیستم‌های OCR نیز از این قاعده مستثنی نیستند. به طور کلی اکثر الگوریتم‌هایی که به نوعی کار تقسیم‌بندی سند[1] را انجام داده‌اند، مرتبط با این تحقیق هستند. در این فصل ابتدا به توضیح قطعه‌بندی و انواع آن می‌پردازیم و سپس به بررسی برخی از پژوهش‌های انجام شده در این حوزه و همچنین پژوهش‌های مرتبط با هر یک از فازهای سیستم‌های DLA خواهیم پرداخت.

## ۲-۱ قطعه‌بندی

به طور کلی به فرآیندی که در آن یک تصویر به چندین بخش و یا منطقه تقسیم می‌شود، قطعه‌بندی تصویر گفته می‌شود. این مناطق که مجموعه‌ای از پیکسل‌های تصویر را به خود اختصاص می‌دهند، به عنوان اشیاء درون تصویر نیز تعریف می‌شوند [۲۲]. قطعه‌بندی تصویر معمولا برای تعیین مکان اشیاء و مرزها درون یک تصویر بکار می‌رود. به طور تخصصی‌تر، قطعه‌بندی تصویر به فرآیندی گفته می‌شود که در آن به هر یک از پیکسل‌های تصویر یک برچسب[2] زده می‌شود به طوری که پیکسل‌هایی که دارای یک برچسب هستند در یک یا چند ویژگی خاص متشابه هستند. این ویژگی‌ها را با توجه به ماهیت اطلاعاتی که از یک تصویر می‌خواهیم بدست آوریم، متفاوت است [۲۳, ۲۴].

[1] Segmentation
[2] Label

رویکردهای متفاوتی در زمینه قطعه‌بندی تصاویر وجود دارد که از مهم‌ترین آنها می‌توان به روش‌های مبتنی بر لبه‌های تصویر[۱]، روش‌های مبتنی بر ناحیه تصویر[۲]، روش‌های بر پایه آستانه‌گذاری[۳] و روش‌های مبتنی بر خوشه‌بندی[۴] اشاره کرد [۲۵-۲۷]. در ادامه به توضیح هر یک از این روش‌های خواهیم پرداخت.

## ۲-۱-۱ قطعه‌بندی بر اساس لبه‌های تصویر

یکی از رایج‌ترین عملیات‌ها در زمینه پردازش تصویر، تشخیص و موقعیت لبه‌ها درون یک تصویر می‌باشد. اگر تشخیص لبه‌ها درون یک تصویر به‌درستی انجام پذیرد، ادامه فرآیند قطعه‌بندی تصویر راحت‌تر و با دقت بالاتری انجام می‌پذیرد. فرآیند تشخیص لبه‌ها درون یک تصویر، یکی از پرکابردترین علمیات‌ها در حوزه پردازش تصویر است. از تشخیص لبه‌ها در فازهای میانی یک عملیات پردازش تصویر استفاده می‌شود و با استفاده از آن ویژگی‌های مختلفی از تصویر استخراج می‌شود. سیستم‌های DLA [۲۸]، آنالیز تصاویر پزشکی [۲۹]، آنالیز و استخراج ویژگی از تصاویر ماهواره‌ای [۳۰] و تشخیص خطوط درون جاده [۳۱] از جمله مهم‌ترین کاربردهای فرآیند تشخیص لبه بشمار می‌آیند. همانطور که گفته شد، یکی از مهم‌ترین کاربردهای تشخیص لبه‌ها درون تصویر، در سیستم‌های DLA است. این فرآیند بویژه در مرحله تشخیص خطوط حاوی متن، نقش بسزایی را ایفا می‌کند. روش‌های گوناگونی برای انجام فرآیند تشخیص لبه‌ها درون تصویر وجود دارد. از جمله این روش‌ها لبه‌یاب پریویت۵، لبه‌یاب سوبل۶، لبه‌یاب کنی۷ است [۳۲-۳۴].

---

[1] Edge based segmentation
[2] Region based segmentation
[3] Thresholding based segmentation
[4] Clustering based segmentation
[5] Prewitt edge detector
[6] Sobel edge detector
[7] Canny edge detector

## ۲-۱-۲ قطعه‌بندی بر اساس ناحیه تصویر

در این دسته از روش‌های قطعه‌بندی، نواحی به طور مستقیم تشخیص داده می‌شوند. در پیکسل‌هایی که موقعیت هر یک از اشیاء درون تصویر را نشان می‌دهند، خصوصیات مشترکی وجود دارد که این خصوصیات مشترک در بین پیکسل‌های یک ناحیه، با پیسکل‌های نواحی دیگر متفاوت است [۳۵].

## ۲-۱-۳ قطعه‌بندی بر اساس آستانه‌گذاری

در این دسته از روش‌های قطعه‌بندی، ابتدا تصاویر سه کاناله به یک کانال خاکستری تبدیل می‌شود. در یک تصویر خاکستری، به هر پیکسل عددی بین ۰ تا ۲۵۵ اختصاص می‌یابد. سپس با مقایسه مقدار عددی هر پیکسل با حد آستانه، قطعه‌بندی تصویر انجام می‌پذیرد. این نوع قطعه‌بندی برای تصاویری کاربرد دارد که در سطوح خاکستری دارای تغییرات زیادی نیست. اگر نیاز باشد که یک تصویر خاکستری به دودویی تبدیل شود، این کار پس از تعیین یک حد آستانه انجام می‌پذیرد. بدین صورت که تمامی پیکسل‌هایی که مقدار عددی آنها در تصویر خاکستری کمتر از حد آستانه باشد، مقدار صفر و مابقی پیکسل‌ها مقدار ۱ (یا ۲۵۵) را به خود اختصاص می‌دهند [۳۵].

## ۲-۱-٤ قطعه‌بندی بر اساس خوشه‌بندی

به طور کلی خوشه‌بندی به فرآیندی گفته می‌شود که در آن داده‌هایی که ویژگی‌های آنها با یک دیگر بیشترین شباهت را دارند در یک دسته قرار می‌گیرند. بدین صورت یک مجموعه داده بر اساس شباهت‌های داده‌های درون خوشه و تفاوت‌ها آنها با داده‌های دیگر خوشه‌ها، به چندین خوشه تقسیم می‌شود. تصویر نیز از این قاعده مستثنی نیست؛ در تصویر می‌توان هر داده را به یک پیکسل و یا یک شی درون آن تصویر نسبت داد و ویژگی آنها را می‌توان فاصله بین پیکسل‌ها و یا اشیاء، رنگ و عمق تصویر در نظر گرفت [۲۷, ۳۶].

## ۲-۲ کارهای انجام شده

همانطور که گفته شد در حوزه سیستم‌های DLA کارهای مفیدی صورت گرفته است. در ادامه به بیان تعدادی از روش‌ها در این حوزه خواهیم پرداخت و سپس مروری بر فعالیت‌های پیشین در این زمینه خواهیم داشت.

### ۲-۲-۱ تشخیص مناطق حاوی متن

همانطور که قبلا گفته شد، اولین قدم در سیستم‌های DLA، تشخیص مناطقی از تصویر است که دارای متن هستند. با حذف مناطق غیرمتنی از تصویر، دقت نهایی سیستم DLA بالا می‌رود. همچنین وجود تصاویر چندین ستونه، اشکال متنوع و مختلف، وجود گراف درون تصویر و غیره باعث پیچیدگی تصویر شده و دقت نهایی سیستم DLA را پایین می‌آورند [۷].

در مرجع [۳۷]، روشی برای تشخیص مناطق حاوی متن ارائه شد. همچنین با بهبود عملیات آستانه‌گذاری، در تشخیص دقیق‌تر این مناطق موفق عمل کرده است. اما این الگوریتم در تشخیص عناصر غیرمتنی مثل ترسیم از یک شی، به خوبی عمل نمی‌کند. لذا در مرجع [۳۸]، با انجام اصلاحاتی در ساختار اصلی الگوریتم، نظیر تکنیک‌های مبتنی بر لبه که نسبت به نویز مقاوم هستند، سعی شده تا عناصر غیرمتنی نیز از درون تصویر استخراج شوند. همچنین در مرجع [۳۷]، از چندین عملیات مورفولوژیکی استفاده شده است.

عملیات‌های مورفولوژیکی موجود در زمینه پردازش تصویر، از نظریه‌هایی با پیشتوانه ریاضی قویی برخوردار هستند [۳۷]. یکی از مهم‌ترین کاربردهای این نوع از عملیات‌ها، تحلیل و پردازش ساختارهای هندسی[1] است. با بکارگیری این نوع از عملیات‌ها، یک الگوریتم با استفاده از اطلاعات مربوط به ساختار[2] و شکل[3] هر شی، بسیار راحت‌تر قادر به شناسایی آنها خواهد بود [۲۴].

[1] Geometrical structure
[2] Structure
[3] Shape

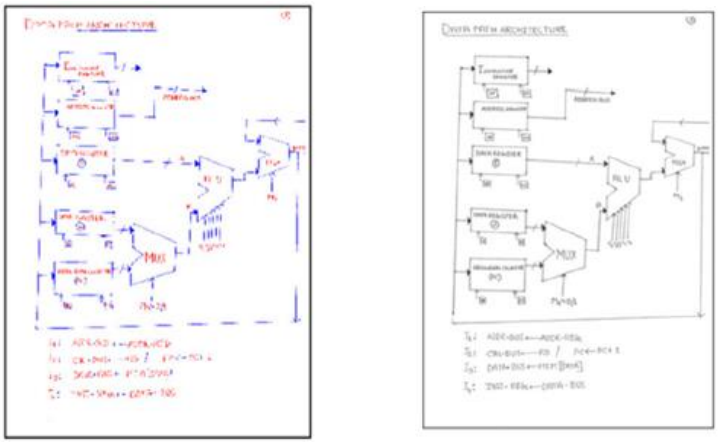
برای استفاده از عملیات‌های مورفولوژیکی، باید به ماهیت فرآیند پردازشی که قرار است انجام شود، دقت ویژه‌تری داشت. نظریه‌های قدرتمندی که در پشت روش‌های مورفولوژی وجود دارد، انعطاف‌پذیری این روش‌ها را بسیار بالا برده است که این انعطاف‌پذیری باعث افزایش کاربردهای این دسته از عملیات‌ها شده است. از جمله کاربردهایی که از عملیات‌های موفولوژیکی استفاده می‌کنند می‌توان به شناسایی خطوط و شناسایی ساختارهای هندسی متفاوت در تصاویر مختلف اشاره نمود. پایه و اساس روش‌هایی که مبتنی بر مورفولوژی هستند، نظریه مجموعه‌ها است [۳۵]. عملگرهایی مانند عملگر سایش[۱]، گسترش[۲]، انتقال[۳] و انعکاس[۴] در عملیات مورفولوژی بکار می‌رود. بسیاری از کاربردهای موجود در الگوریتم‌های فرآیندهای پردازشی، توسط این عملگرها قابل انجام است.

یکی از عملیات‌های مورفولوژی، عملیات نازک‌سازی است [۳۹]. این عملیات معمولا در مرحله پیش‌پردازش یک فرآیند صورت می‌گیرد. یکی از ملزومات اساسی در آماده‌سازی ساختارهای موجود در تصویر، عملیات نازک‌سازی است. در این عملیات، اندازه ساختارهای موجود در یک تصویر کاهش پیدا می‌کند. در این کاهش اندازه، معمولا آن قسمت‌هایی از یک شکل یا ساختار حذف می‌شوند که بیشتر به پس‌زمینه تصویر شبیه و نزدیک هستند. این کاهش اندازه، به‌صورت لایه‌به‌لایه و از خارجی‌ترین پیکسل‌های متعلق به یک شکل صورت می‌پذیرد [۳۹].

برای انجام عملیات نازک‌سازی، روش‌های متنوعی ارائه شده است، الگوریتم‌های مبتنی بر تکرار[۵]، الگوریتم‌های غیر تکراری[۶] و الگوریتم‌هایی که از رویکرد موازی[۷] استفاده کرده‌اند، از جمله این روش‌ها هستند [۳۹].

[1] Erosion
[2] Dilation
[3] Transition
[4] Reflection
[5] Iterative Thinning Algorithms
[6] Non-Iterative Algorithms
[7] Fully Parallel Approaches

در مرجع [۴۰]، برای تشخیص مناطق متنی و غیر متنی، ابتدا با استفاده از الگوریتم‌های مبتنی بر الگوهای دودویی محلی (LBP)[۱]، برای هر یک از CCهای درون تصویر یک بردار ویژگی می‌سازد. برای ساخت این بردار ویژگی روش‌های متعددی مانند LBP پایه[۲]، LBP بهبود یافته (ILBP)[۳]، LBP یکنواخت (ULBP)[۴]، LBP ثابت نسبت به چرخش (RILBP)[۵]، LBP مقاوم و یکنواخت (RULBP)[۶] و LBP ثابت به چرخش و یکنواخت (RIULBP)[۷] وجود دارد [۴۱-۴۴]. در این مرجع، بیان شده است که بردار ویژگی استخراج شده توسط هر یک از الگوریتم‌های مبتنی بر LBP دارای ابعاد و طول زیادی است که علاوه بر پیچیدگی محاسباتی بالا به همراه محدودیت‌های زمانی و فضایی، باعث کاهش دقت در الگوریتم دسته‌بندی نیز می‌شود. لذا با استفاده از نوعی بازی مشارکتی اقدام به انتخاب ویژگی می‌کند. با این کار محتمل‌ترین مناطق که بیشترین اطلاعات را دارند، استخراج می‌شوند تا دسته‌بندی راحت‌تر انجام شود. همانطور که در شکل ۱-۲ نشان داده شده است، ورودی این سیستم باید دارای یک کانال باشه و در نهایت تصویر را بر اساس دو رنگ قرمز و آبی که به‌ترتیب نمایشگر مناطق متنی و غیرمتنی هستند، تقسیم می‌کند.



الف: تصویر اصلی ب: تصویر خروجی

شکل ۱-۲: ورودی و نتیجه نهایی تشخیص مناطق متنی و غیرمتنی در مرجع[۴۰]

[1] Local Binary Pattern
[2] Basic LBP
[3] Improved LBP
[4] Uniform LBP
[5] Rotation Invariant
[6] Robust and Uniform LBP
[7] Rotation Invariant and Uniform LBP

به‌طور کلی رویکردها در زمینه آنالیز قالب‌بندی اسناد، به دو نوع پایین به بالا[1] و بالا به پایین[2] تقسیم می‌شوند. در روش‌های پایین به بالا همانند روشی که در مرجع [۷] برای تشخیص مناطق متنی و غیرمتنی در تصاویر اسناد به زبان هندی بکار رفته است، ابتدا محدوده هر CC را با کشیدن یک کادر محصورکننده[3] حول آن مشخص می‌کند. سپس کادرهای نزدیک به هم در راستای افق و عمود را به یکدیگر متصل می‌سازد و در نهایت با استفاده از خصوصیاتی نظیر مساحت مناطق حاصله، به تشیخص مناطق متنی و غیرمتنی می‌پردازد. همانطور که در شکل ۲-۲ مشاهده می‌شود، پس از اتصال باکس‌ها به‌یکدیگر، مناطق متنی و غیرمتنی در تصویر اصلی مشخص شده‌اند.


الف: تصویر اصلی ب: کشیدن کادر حول CCها

ج: استخراج مناطق غیرمتنی د: استخراج مناطق متنی

شکل ۲-۲: مراحل تشخیص مناطق متنی و غیرمتنی در مرجع [۷]

[1] Bottom-Up
[2] Top-Down
[3] Bounding Box

در مرجع [۶] نیز از روش پایین به بالا برای تشخیص مناطق متنی و غیرمتنی استفاده شده است. در این مرجع [۶] که همه مراحل یک سیستم DLA را در خود جای داده است، برای بخش تشخیص مناطق متنی و غیرمتنی از یک شبکه عصبی کانولوشنی عمیق (DCNN)[1] آموزش دیده، بهره گرفته است. این نوع مدل‌ها که در حوزه یادگیری انتقال (TL)[2] قرار می‌گیرند بسیار مفید هستند. مدل‌هایی که با استفاده از یک دیتاست آموزش دیده‌اند و برای یک کار دیگر با دیتاستی متفاوت استفاده می‌شوند [۴۵]. در شکل ۳-۲ نمونه‌ای از خروجی این مرحله در این مرجع [۶]، نشان داده شده است.

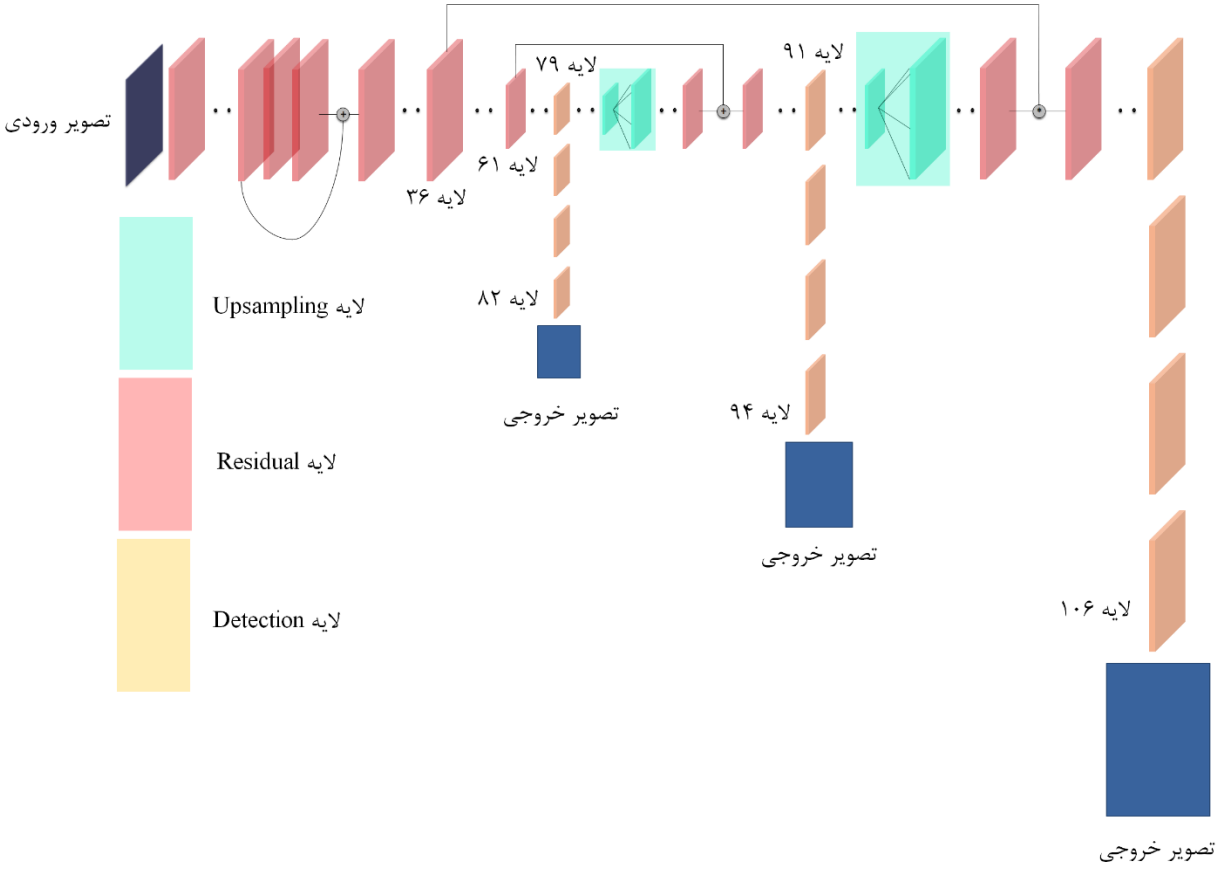

الف: تصویر اصلی ب: تصویر خروجی

شکل ۳-۲: خروجی مدل شبکه عصبی کانولوشنی عمیق برای تشخیص مناطق متنی و غیرمتنی [۶]

یکی دیگر از روش‌های مبتنی بر یادگیری عمیق که در سال‌های اخیر ارائه شده است، روش YOLO[3] است [۴۶]. این روش که اولین بار در سال ۲۰۱۶ ارائه شد، جز روش‌های بلادرنگ[4] محسوب می‌شود که در مقایسه با دیگر روش‌های تشخیص اشیا، عملکرد بهتری در این زمینه دارد. YOLOv3 [۴۷] یکی از

[1] Deep Convolutional Neural Networks
[2] Transfer Learning
[3] You Only Look Once
[4] Real-time

روش‌های جدید مبتنی بر YOLO است که دارای ۵۳ لایه آموزش دیده بر روی ImageNet، به همراه ۵۳ لایه دیگر برای تشخیص اشیا است. بدلیل وجود ۱۰۶ لایه در YOLOv3، این ساختار به نسبت دیگر ساختارهای YOLO مثل YOLOv2، کمی کندتر است اما از دیگر جوانب، عملکرد بهتری دارد. ساختار کلی این الگوریتم در شکل ۴-۲ نشان داده شده است.

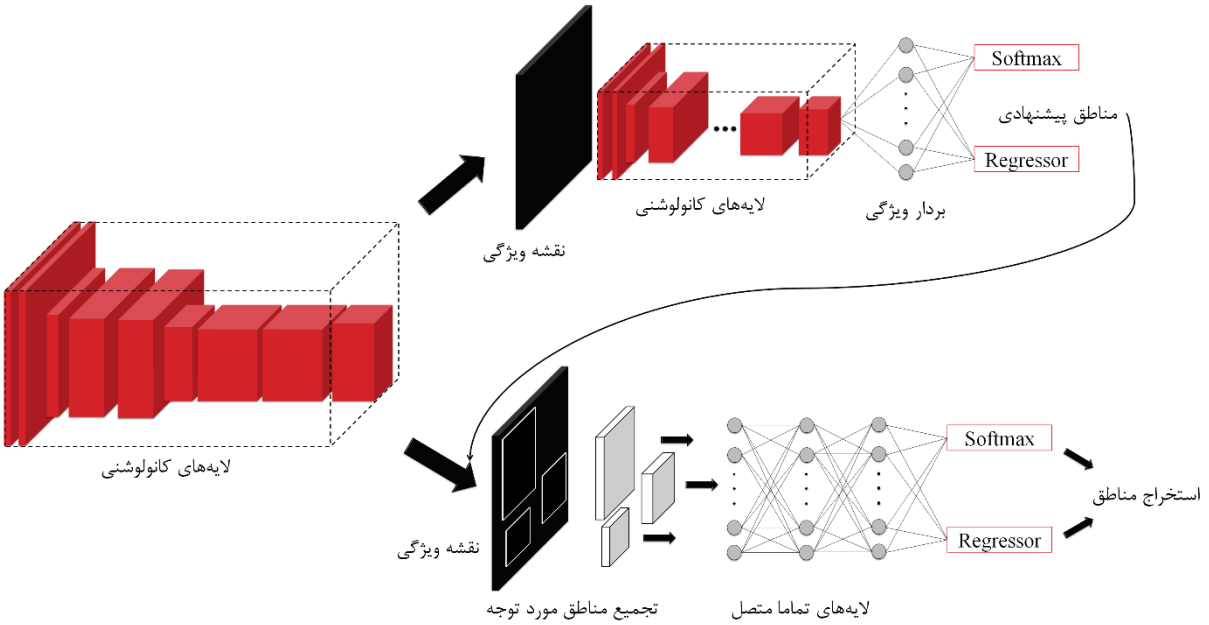



شکل ۴-۲: ساختار کلی الگوریتم YOLOv3 [۴۷]

یکی از روش‌های مناسب در زمینه تشخیص مناطق متنی و غیر متنی، Faster R-CNN[1] است [۴۸]. تفاوت عمده این روش با روش‌های پیشین خود همانند R-CNN و Fast R-CNN، در نحوه استخراج مناطق است. این دو روش با بکارگیری جستجوی انتخابی[2] مناطق مورد نظر را استخراج می‌کردند در حالی که در Faster R-CNN جستجوی انتخابی جای خود را به RPN[3] داده است. همان‌طور که در شکل ۵-۲ نشان داده شده

[1] Faster Region-based Convolution Neural Network
[2] Selective Search
[3] Region Proposal Network

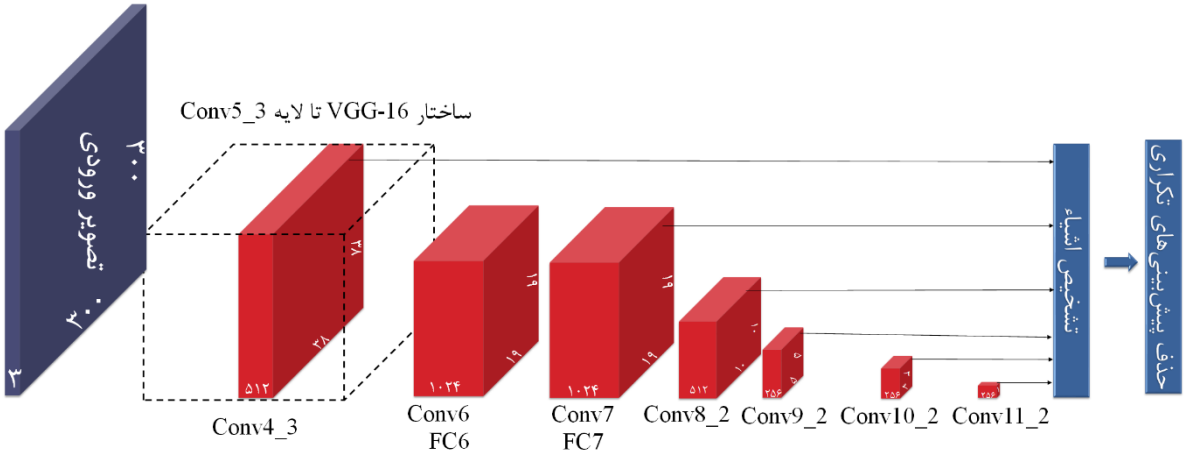
است، در این روش، تصویر ورودی به تعدادی لایه کانولوشنی داده می‌شود تا نقشه ویژگی[1] تصویر استخراج شود. سپس با بکارگیری RPN بر روی این نقشه ویژگی، مناطقی که وجود اشیا در آنها محتمل‌تر است، بدست می‌آیند و با کادرهایی خاص مشخص می‌شوند. سپس این مناطق به سایز اصلی تصویر برگردانده شده و در نهایت کادرها طبقه‌بندی می‌شوند.




شکل ۵-۲: ساختار کلی الگوریتم Faster R-CNN [۴۹]

یکی دیگر از روش‌های مبتنی بر یادگیری عمیق (SSD)[2] است [۵۰]. همان‌طور که در شکل ۶-۲ نشان داده شده است، در این روش از شبکه کانولوشنی برای محاسبه نقشه ویژگی تصویر ورودی استفاده می‌شود. سپس با بکارگیری شبکه‌های کانولوشنی کوچک ۳ در ۳ بر روی نقشه ویژگی، مناطق متنی و غیرمتنی را تشخیص و دور آنها کادر می‌کشد.

[1] Feature map
[2] Single Shot Detector




شکل ۶-۲: ساختار کلی الگوریتم SSD [۵۰]

از دیگر روش‌های تشخیص مناطق حاوی متن، استفاده از Layout Parser است [۵۱]. Layout Parser ابزاری مبتنی بر یادگیری عمیق برای کارهای تجزیه و تحلیل طرح اسناد است. این ابزار که از طریق GitHub قابل دسترسی است، ابزاری نسبتا قدرتمند است که توانایی استخراج مناطق متنی و غیرمتنی از درون تصاویر حتی نسبتا پیچیده را نیز دارد.

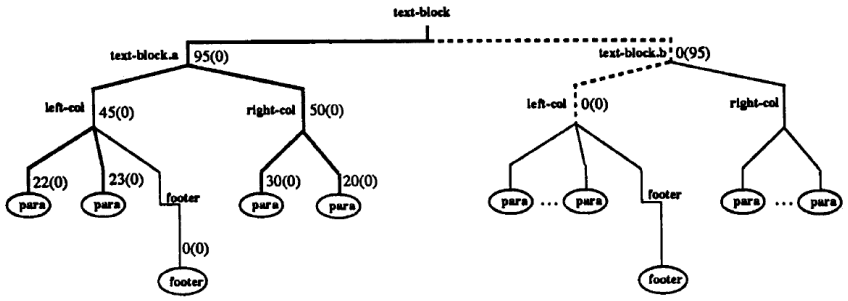
یکی از روش‌های مناسب در زمینه تشخیص مناطق متنی و غیر متنی، تبدیل هاف[1] است. با استفاده از تبدیل هاف می‌توان اشکال هندسی نظیر دایره، بیضی، مستطیل، خط و به‌طور کل هر شکل دیگر هندسی را به‌راحتی تشخیص داد به شرطی که بتوان برای آن اشکال یک فرمول ریاضی ارائه کرد. به عنوان مثال، خط را می‌توان با استفاده از دو پارامتر شیب و عرض از مبدا تعریف کرد. لذا می‌توان با استفاده از تبدیل هاف آن را استخراج نمود [۳۵].

برای تشخیص اشیا درون تصویر با استفاده از تبدیل هاف، ابتدا می‌بایست لبه‌های درون تصویر استخراج شوند. در مرحله بعد، کار شناسایی اشیا با استفاده از این لبه‌ها انجام می‌پذیرد [۳۵]. در تبدیل هاف، فضای جدید به وسیله یک سری از سلول‌ها که به انباشتگر[2] معروف‌اند، توصیف می‌شود. در یک تصویر دودویی، برای هر یک از مقادیر مربوط به لبه، یک عمل نگاشت یا همان فرآیند رای‌گیری به فضای انباشتگر انجام

[1] Hough Transform
[2] Accumulator

می‌شود. در پایان با در نظر گرفتن مختصات نقاط بیشینه در فضای انباشتگر (مختصات بیشترین رأی در فضای انباشتگر) و جایگذاری آن‌ها در معادله مدل مربوط به آن شی، شکل مورد نظر در تصویر شناسایی می‌شود [۳۵].

استفاده از درخت تصمیم یکی دیگر از راهکارهای تشخیص مناطق متنی و غیرمتنی درون تصویر است. یکی از روش‌های تقسیم‌بندی تصویر مبتنی بر درخت تصمیم، برش X Y[1] نام دارد. که در آن هر صفحه از یک سند در ریشه درخت قرار می‌گیرد و برگ‌ها مناطق نهایی تقسیم‌بندی شده هستند. در هر گره[2] این درخت ناحیه مورد نظر را به دو مستطیل کوچک‌تر تقسیم می‌کند و این کار تا آنجا ادامه می‌یابد که دیگر ناحیه‌ای را نتوان تقسیم نمود [۵۲]. شکل ۷-۲ یک نمونه از این نوع درخت را به نمایش درآورده است.




شکل ۷-۲: درخت تصمیم برش X Y در مرجع [۵۲]

در مرجع [۵۳]، یک روش جدید برای تشخیص مناطق متنی و غیرمتنی از درون تصاویر صحنه‌های طبیعی[3] ارائه شده است. دستیابی به تشخیص درست با دقت بالا، منوط به انتخاب ویژگی‌ها و طبقه‌بندهای بهینه است. این مرجع [۵۳]، توانسته با بهبود دادن الگوریتم (MSER)[4]، محتمل‌ترین مناطق متنی درون تصویر را تشخیص دهد. در مرحله بعدی، یازده ویژگی را از درون این مناطق استخراج می‌کند. این ویژگی‌ها شامل ویژگی‌های مرتبط با متن، مبتنی بر لبه، مبتنی بر رنگ و مبتنی بر شکل هندسی هستند. سپس با استفاده از پارامترهای BFS و CfsSubsetEval درون ابزار وکا[5] [۵۴]، روش پیشنهادی این مرجع [۵۳]، از بین این

[1] X Y-Cut
[2] Node
[3] Natural scene images
[4] Maximally Stable Extremal Region
[5] Weka tool

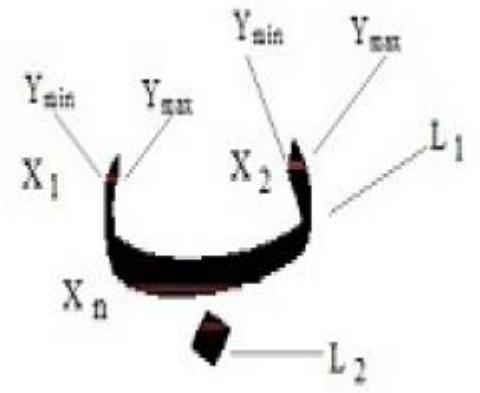
یازده ویژگی، مجموعه ویژگی بهینه را انتخاب می‌کند تا بتواند بخش متنی و غیرمتنی را از هم تفکیک کند. در نهایت با بکارگیری ابزار وکا بر روی بخش آموزش دادگان ICDAR 2013 [۵۵]، طبقه‌بندهای زیادی توسط این روش آموزش می‌بینند.

## ۲-۲-۲ استخراج خطوط حاوی متن از درون مناطق متنی

همانطور که در بخش قبل گفته شد، سیستم‌های DLA، پس از تشخیص مناطق متنی، به استخراج خطوط درون هر منطقه می‌پردازد. در سال‌های اخیر روش متعددی برای استخراج خطوط حاوی متن معرفی شده‌اند. این روش‌ها به چندین دسته مختلف تقسیم می‌شوند. روش‌های مبتنی بر اطلاعات مستخرج از CC، روش‌های بر اساس نمایه تصویر[۱]، روش‌های مبتنی بر الگوریتم لکه‌گیری[۲]، روش‌های کادر محصورکننده، روش‌های برپایه تبدیل هاف، روش‌های تلفیقی و روش‌های مبتنی بر یادگیری عمیق[۳] چند نمونه از این دسته‌ها هستند. در ادامه به توضیح هر کدام خواهیم پرداخت.

در مرجع [۵۶]، برای اتصال CCها درون یک خط از اطلاعات مربوط به لبه آنها بهره گرفته است. این مرجع [۵۶]، از سه نوع ماژول برای استخراج خطوط حاوی متن درون تصویر استفاده کرده است. در ماژول اول تمامی CCهای درون تصویر استخراج می‌شود و سپس هر یک از این CCها به ماژول دوم داده می‌شوند تا اطلاعات مربوط به لبه آنها استخراج شود. در شکل ۸-۲ نمونه‌ای از نوع اطلاعات استخراج شده با استفاده از ماژول دوم نشان داده شده است. در نهایت ماژول سوم با توجه به اطلاعات بدست آمده از ماژول دوم، CCهایی که در راستای افق و در نزدیکی هم هستند را به یکدیگر متصل می‌کند. این مرجع [۵۶]، روش پیشنهادی خود را بر روی دو مجموعه داده‌ای که خود ساخته است آزمایش کرده و به دقت‌های ۸۷.۳۶٪ و ۸۴.۷۵٪ بر روی این دو دیتاست رسیده است.

[1] Project Profile
[2] Smearing
[3] Deep learning




شکل ۸-۲: اطلاعات مربوط به لبه استخراج شده از یک CC در مرجع [۵۶]

در مرجع [۵۷]، یک روش جدید برای استخراج خطوط از درون تصاویری که توسط دوربین گرفته شده‌اند، ارائه شده است. این مرجع [۵۷]، توانسته بر روی تصاویر سه کاناله که دارای پس‌زمینه‌های طبیعی هستند، به دقت خوبی دست پیدا کند. در این مرجع، ابتدا مختصات هر CC را با استفاده از الگوریتم MSER، تخمین می‌زند [۵۸]. جهت هر یک از CCها با استفاده از نمایه تصویر، استخراج و سپس مناطق کاندیدی که بیانگر موقعیت خطوط است از درون تصویر استخراج می‌شود. در نهایت با استفاده از تکنیک پایین به بالا، مناطق متنی استخراج می‌شوند.

الف: تصویر اصلی ب: تصویر خروجی

شکل ۹-۲: نمونه‌ای از استخراج مناطق کاندید برای هر خط توسط مرجع [۵۷]

شکل ۹-۲ نمونه‌ای از مناطق کاندید برای خطوط توسط این مرجع [۵۷] را نشان داده است. این مرجع [۵۷] توانسته چالش‌هایی مثل خطوط دارای انحنا و همچنین تصاویری که پس‌زمینه‌های رنگی دارند را تا حدودی حل کند.

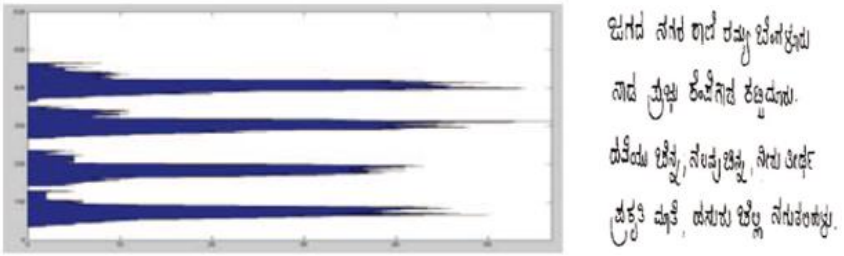
در مرجع [۵۹]، یک روش برای تشخیص خطوط حاوی متن درون تصویر ارائه شده است که با استفاده از نمودار ورونوی[1] موقعیت خطوط را تخمین می‌زند. در نمودار ورونوی، هر CC به عنوان مرکز یک سلول در نظر گرفته می‌شود. اندازه هر سلول، بسته به وجود CCهای دیگر در اطراف این سلول متغیر خواهد بود. در شکل ۱۰-۲ نمونه‌ای از نمودار ورونوی بر روی یک تصویر حاوی CCهای کاراکتری[2] نشان داده شده است. پس از مشخص شدن ناحیه مربوط به هر سلول با مرکزیت یک CC، این مرجع [۵۹]، برای اتصال CCها و استخراج خطوط، از الگوریتم خوشه‌بندی بر اساس نزدیک‌ترین همسایه بهره گرفته است و بدین صورت دو ناحیه رو به یکدیگر متصل می‌سازد.


شکل ۱۰-۲: نواحی تقسیم‌شده توسط نمودار ورونوی در یک تصویر حاوی CCهای کاراکتری در مرجع [۵۹]

بعضی دیگر از روش‌ها برای استخراج خطوط درون تصویر، از تکنیک‌های مبتنی بر نمایه تصویر استفاده می‌کنند. این تکنیک‌ها به‌طور کلی به دو دسته نمایه تصویر افقی و نمایه تصویر عمودی تقسیم می‌شوند [۶۰-۶۵]. در تکنیک‌های مبتنی بر نمایه تصویر افقی، امکان استخراج خطوط وجود دارد. در این نوع از تکنیک‌ها، که بر روی تصاویر دودویی (و یا در بعضی از موارد تک کانال خاکستری) قابل پیاده سازی است، تعداد پیکسل‌های سیاه (یا سفید بسته به رنگ پس‌زمینه) را در هر سطر بدست می‌آورد و در یک نمودار

[1] Voronoi diagram
[2] Character

نشان می‌دهد. سپس با استفاده از گپ[1] بین خطوط، کار استخراج را انجام می‌دهد. در شکل ۱۱-۲ نمونه‌ای از نمایه تصویر افقی برای نوعی زبان هندی نشان داده شده است [۶۴]. همان‌طور که در شکل ۱۱-۲ نشان داده شده است، بین دو خط پیکسل‌های مشکلی کمتری وجود دارند. با استفاده از همین ساختار می‌توان خطوط را از یکدیگر جدا کرد.



الف: تصویر دودویی ب: نمایه تصویر افقی

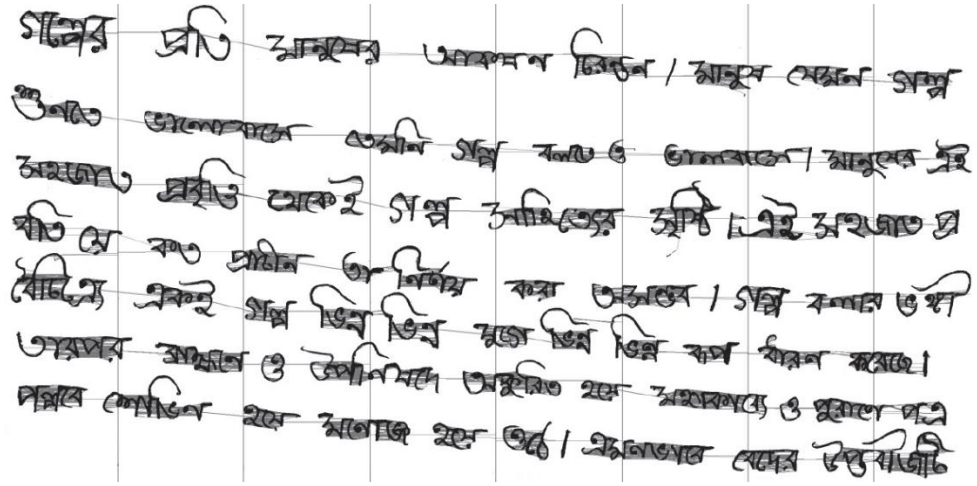
شکل ۱۱-۲: نمونه‌ای از نمایه تصویر افقی در مرجع [۶۴]

از این تکنیک معمولا برای متون چاپی استفاده می‌شود، جایی که خطوط با هم هیچ تداخلی نداشته باشند. همچنین این روش برای تصاویری که خطوط آنها دارای انحنا باشد و یا حتی خطوط مقدار زیادی کج باشند مناسب نیست. بعلاوه اگر فاصله خطوط از هم بسیار کم باشد باز هم این الگوریتم به خوبی جواب نمی‌گیرد. البته تکنیک‌ها نمایه تصویر عمودی، برای استخراج مناطق متنی در تصاویر چند ستونه مناسب هستند که می‌توان از آنها بهره برد [۱۹]. شکل ۱۲-۲ نمونه‌ای از این نوع نمایه تصویر را نشان داده است.

[1] gap

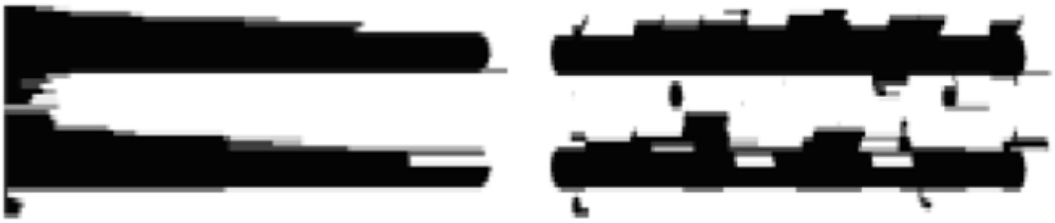

شکل ۱۲-۲: نمونه‌ای از نمایه تصویر عمودی برای تشخیص تصاویر چند ستونه در مرجع [۱۹]

روش‌های مبتنی بر لکه‌گیری، با استفاده از شباهت‌های مناطق مختلف تصویر به یکدیگر و مکان‌های متنی، کار تشخیص و جداسازی خطوط را انجام می‌دهند. این تکنیک قدرت تشخیص بر روی هر دو نوع متون چاپی و دست‌نویس را دارد، اما مشابه تکنیک نمایه تصویر، اگر خطوط و یا حتی کلمات درون تصویر با هم، همپوشانی داشته باشند، دقت این تکنیک پایین می‌آید [۱۹, ۶۶]. پس از استفاده از تکنیک لکه‌گیری، حروف یک کلمه به یکدیگر متصل می‌شوند و سپس با استفاده از الگوریتم‌هایی مانند پیدا کردن نزدیک‌ترین همسایه، خطوط درون تصویر استخراج می‌شوند. شکل ۱۳-۲ نمونه‌ای از خروجی بخش مربوط به تکنیک لکه‌گیری در مرجع [۶۶] را نشان داده است.



شکل ۱۳-۲: استفاده از تکنیک لکه‌گیری در تصویر در مرجع [۶۶]

یکی دیگر از روش‌های تشخیص خطوط حاوی متن، استفاده از کادر محصورکننده است. در این روش، حول هر کاراکتر و یا کلمه یک کادر می‌کشند که تمامی پیکسل‌های آن کاراکتر و یا کلمه را در بر بگیرد. سپس مشابه روشی که پس از الگوریتم لکه گیری استفاده می‌شود می‌توان کلمات را به یکدیگر متصل ساخت و خط را استخراج کرد. روش دیگر مشابه تکنیک نمایه تصویر افقی است که هیستوگرام تصویر را کشیده و سپس خطوط را بر اساس آنجاهایی که کم‌ترین پیکسل‌ها را دارند، از یکدیگر جدا نمود. در مرحله بعد سعی می‌کند مرکز هر خط را پیدا کند و کادر محصور کننده‌ای را حول آن خط بکشد [۶۳]. همانطور که در شکل ۱۴-۲ نشان داده شده است، ابتدا با روشی همانند لکه‌گیری، کاراکترها و سپس کلمات را به هم می‌چسباند و سپس با کشیدن هیستوگرام تصویر، مرز جداساز خطوط مشخص می‌شود.

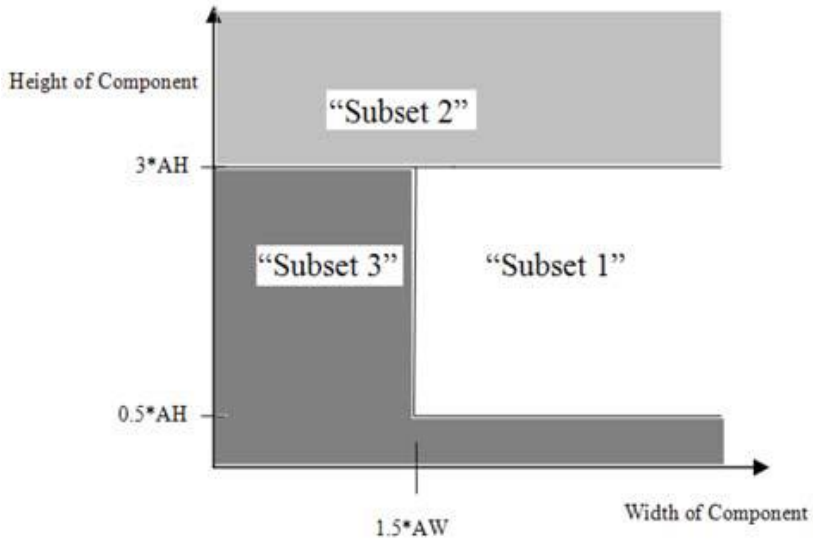


الف: اتصال کاراکترهای یک خط ب: کشیدن هستوگرام برای تشخیص مرز جداساز

شکل ۱۴-۲: نحوه تشخیص مرز جداساز دو خط در مرجع [۶۳]

همانطور که در بخش قبل گفته شد، یکی از روش‌های کارآمد در زمینه تشخیص خطوط و به‌طور کل در حوزه سیستم‌های DLA، تبدیل هاف است. در مرحله دوم تبدیل هاف، که نگاشت تبدیل هاف نام دارد، از

یک تبدیل هاف مبتنی بر بلوک، برای شناسایی خطوط حاوی متن درون تصویر استفاده می‌شود که زیر مجموعه‌ای از CCهای درون تصویر را در نظر می‌گیرد. شکل ۱۵-۲ نمونه‌ای از این نوع زیر مجموعه را نشان داده است [۶۷].




شکل ۱۵-۲: فضای CCها با سه زیر مجموعه در مرجع [۶۷]

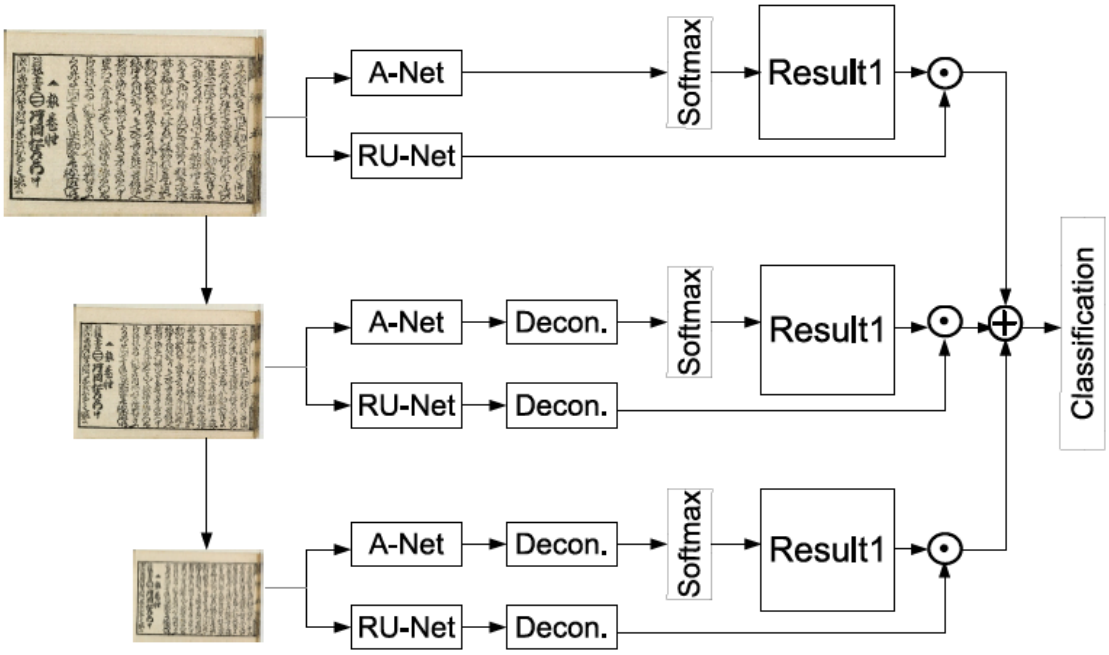
دلایل انتخاب این زیر مجموعه‌ها بدین صورت است:

۱. تضمین این که اجزایی که در بیش از یک خط ظاهر می‌شوند، در حوزه هاف رأی نمی‌دهند.

۲. اجزایی مانند لهجه‌ها که اندازه کوچکی دارند باید از این مرحله حذف شوند. زیرا می‌توانند با اتصال تمام لهجه‌های بالای خط متن اصلی باعث تشخیص خط متن اشتباه شوند.

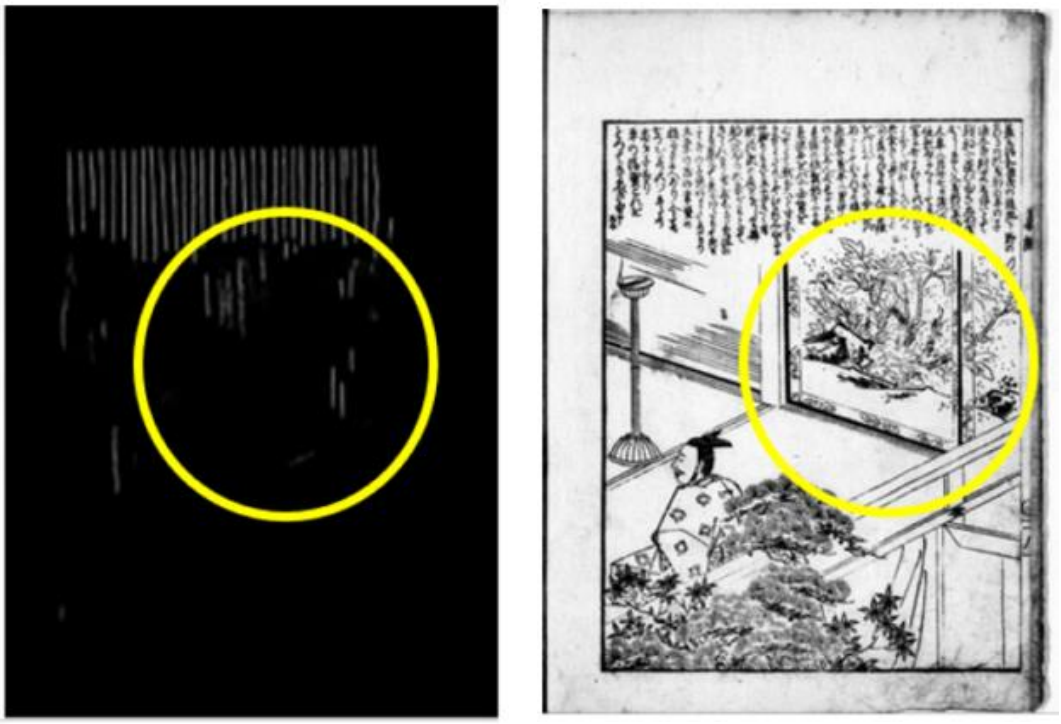
در سال‌های اخیر با افزایش قدرت سخت‌افزارها بحث یادگیری عمیق دوباره مطرح گردیده و در اکثر حوزه‌های هوش مصنوعی نقش مهمی را ایفا می‌کند. در زمینه سیستم‌های تشخیص خطوط حاوی متن درون تصویر نیز پژوهشگران زیادی سعی در استفاده از یادگیری عمیق داشته‌اند تا به دقت‌های بالاتری دست پیدا کنند. در مرجع [۶۸]، از شبکه ARU-Net برای استخراج خطوط حاوی متن درون تصویر استفاده شده است [۶۹]. شبکه ARU-Net یک مدل یادگیری عمیق است که برای تقسیم‌بندی مناطق

اسناد تاریخی استفاده می‌شود. همانطور که در شکل ۱۶-۲ نشان داده شده است، ساختار این شبکه با استفاده از دو شبکه A-Net و RU-Net بنا شده است.




شکل ۱۶-۲: ساختار شبکه ARU-Net برای استخراج خطوط در مرجع [۶۸]

در فرآیند استخراج خطوط، احتمال از دست رفتن اطلاعات و عدم تشخیص بعضی از خطوط وجود دارد. لذا در این مرجع [۶۸]، از شبکه LeNet که یک شبکه عصبی عمیق نسبتا ساده می‌باشد برای افزایش دقت و تشخیص مناطق متنی تصویر استفاده شده است. در شکل ۱۷-۲ نمونه‌ای از نتیجه استخراج خطوط توسط مرجع [۶۸] نشان داده شده است.



الف: تصویر اصلی ب: نتیجه استخراج خطوط

شکل ۱۷-۲: استخراج خطوط توسط مرجع [۶۸]

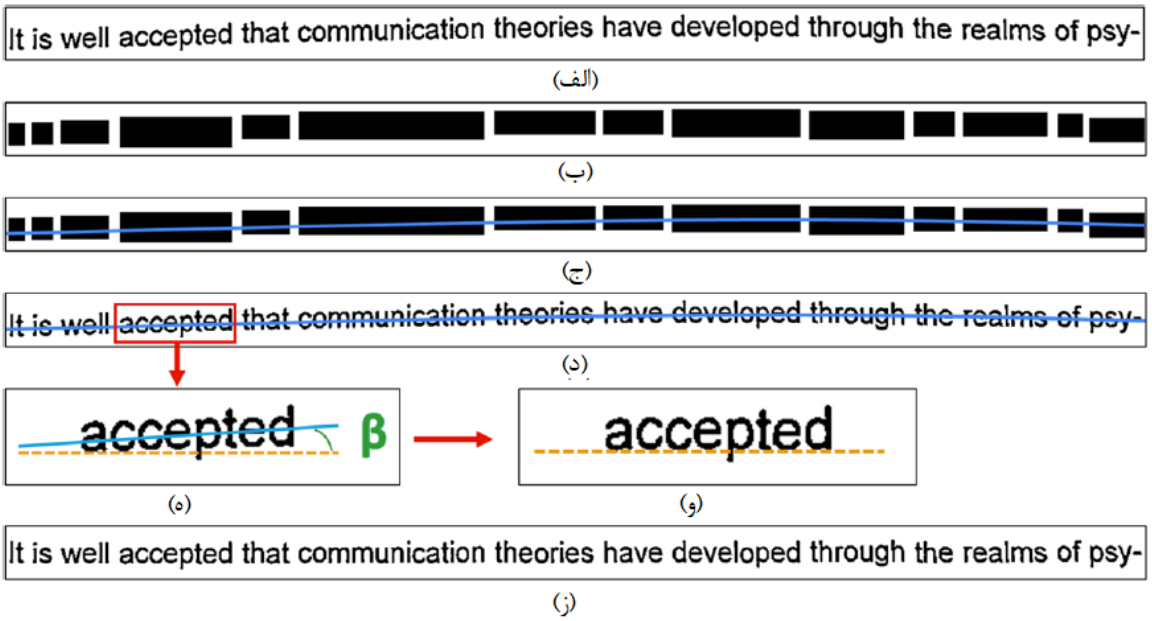
یادگیری عمیق یکی از تکنیک‌های کارآمد در زمینه تشخیص خطوط حاوی متن از درون تصویر است. اما برای دستیابی به یک دقت بالا، یک شبکه عصبی عمیق نیاز به آموزش به وسیله یک حجم عظیمی از تصاویر دارد. در زبان فارسی ما با کمبود مجموعه داده‌های استاندارد که حاوی تصاویری از اسناد چاپی باشند، مواجهه هستیم و لذا امکان استفاده از این نوع یادگیری برای ما وجود ندارد.

### ۲-۲-۲-۱ وجود خطوط دارای انحنا درون تصویر

همان‌طور که در فصل اول بیان شد، یکی از مشکلات اساسی در زمینه تشخیص خطوط حاوی متن درون تصویر، وجود خطوط است که دارای شکل انحنایی هستند. استخراج صحیح این نوع خطوط یکی از چالش‌های اصلی پژوهشگران در سال‌های اخیر بوده است. همان‌طور که در شکل ۱-۱ نشان داده شده است، شکل این خطوط منحنی است که استفاده از روش‌هایی نظیر کادر محصور کننده و یا نمایه تصویر

را سخت و یا غیر ممکن می‌سازد. در سال‌های اخیر روش‌های نو ظهوری برای حل این مشکل ارائه شده‌اند که در ادامه به توضیح چند مورد از آنها خواهیم پرداخت.

در مرجع [۷۰]، یک روش جدید برای استخراج این نوع خطوط ارائه شده است. در مرحله پیش‌پردازش پس از رفع نویز، و حذف پس‌زمینه تصویر با کمک تکنیک dilation مناطق متنی تصویر بدست می‌آیند. در نهایت با اتصال CCهای هر خط به یکدیگر، خطوط درون تصویر استخراج می‌شوند. همان‌طور که در شکل ۱۸-۲ نشان داده شده است، این مرجع [۷۰]، برای استخراج خطوط از تکنیک خط پایه[1] بهره گرفته است و سپس سعی در تصحیح زاویه انحنا دارد.




شکل ۱۸-۲: مراحل استخراج و تصحیح زاویه خط با کمک تکنیک خط پایه در مرجع [۷۰].
الف: تصویر اصلی. ب: اتصال CCهای نزدیک به هم و استخراج کلمات. ج: تخمین خط پایه. د: ردیابی خط پایه در تصویر اصلی. ه: برآورد زاویه هر کلمه نسبت به خط پایه. و: تصحیح زاویه و صاف کردن کلمه. ز: تصویر نتیجه پس از اصلاح زوایا

[1] baseline

## ۳-۲ جمع‌بندی

در این فصل ابتدا به بیان مفهوم قطعه‌بندی، که فرایند اصلی آشکارسازی مناطق متنی و همچنین خطوط را در بسیاری از رویکردها و روش‌های موجود تشکیل می‌دهد پرداخته شد. در ادامه به انواع روش‌های قطعه‌بندی اشاره گردید. سپس روش‌های مبتنی بر پیشینه تحقیق و نهایتا، تعدادی از پژوهش‌ها و فعالیت‌های پیشین  را مورد بررسی قرار دادیم. در ادامه و در فصل بعد به بیان روش پیشنهادی خواهیم پرداخت.

# فصل ۳ : روش پیشنهادی

## ۳-۱ مقدمه

در فصل‌های پیش به تعریف مسئله پرداخته شد و در ادامه، کارهای انجام شده مورد بررسی قرار گرفت. در این فصل، روش پیشنهادی با جزئیات مورد بررسی قرار می‌گیرد. روش پیشنهادی از دو مرحله مجزا تشکیل شده است. از این رو، ابتدا به توضیح بخش‌های مختلف مرحله اول پرداخته‌ایم و سپس به‌سراغ مرحله دوم رفته‌ایم. در مرحله اول، پس از اصلاح و رفع نویزهای درون تصویر در مرحله پیش‌پردازش، با استفاده از چهار مدل مبتنی بر یادگیری عمیق، مناطق متنی و غیرمتنی درون هر تصویر را تخمین می‌زنیم. سپس با بکارگیری سیستم رای‌گیری مناطق دقیق مربوط به هر دسته را استخراج می‌نماییم. در مرحله بعد برای پیکسل‌هایی که سیستم رای‌گیری نمی‌تواند در موردشان با قطعیت نظر بدهد از روش پنجره و همچنین مدل OCRopus استفاده می‌کنیم.

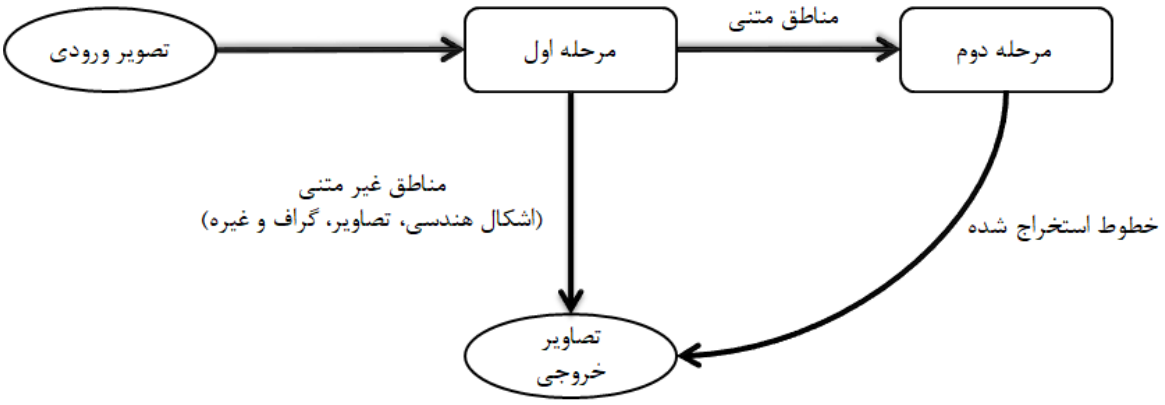
پس از استخراج مناطق متنی درون تصویر، مرحله دوم شروع می‌شود. در این مرحله ابتدا در مرحله پیش‌پردازش تصویر را به حال دلخواه تبدیل می‌نماییم. سپس مفهومی را به‌نام قلم نهایی تصویر تعریف نموده و اندازه آن را محاسبه می‌نماییم. با بکارگیری اندازه قلم نهایی اجزای متصل در هر خط را به یکدیگر متصل می‌سازیم تا در نهایت بهترین منطقه برای هر خط استخراج می‌شود. در ادامه به توضیح مفصل ه کدام از این مراحل می‌پردازیم.

## ۳-۲ روش پیشنهادی

همانطور که گفته شد، ورودی یک سیستم DLA یک تصویر حاوی متن است که در نهایت خطوط درون آن، استخراج می‌شوند. لذا ابتدا نیاز است که مناطق غیرمتنی مانند شکل، گراف و غیره، از درون تصویر حذف شوند و سپس خطوط درون مناطق متنی تصویر استخراج شوند. به‌همین منظور، روش پیشنهادی شامل دو مرحله است. همان‌طور که در شکل ۱-۳ نشان داده شده است، در مرحله اول مناطق متنی و

غیرمتنی درون تصویر جدا شده و در مرحله دوم الگوریتم، خطوط حاوی متن را از درون مناطق متنی استخراج می‌کند.

در مرحله اول، پس از پیاده‌سازی چهار مدل مبتنی بر یادگیری عمیق برای استخراج مناطق حاوی متن از درون تصویر، با استفاده از یک الگوریتم رای‌گیری مناطق محتمل‌تر متنی استخراج می‌شوند در مرحله دوم، آشکارسازی خطوط بر روی مناطق متنی، انجام می‌شود. استخراج خطوط با بکارگیری روشی نوین مبتنی بر اندازه قلم نهایی تصویر انجام می‌گیرد.



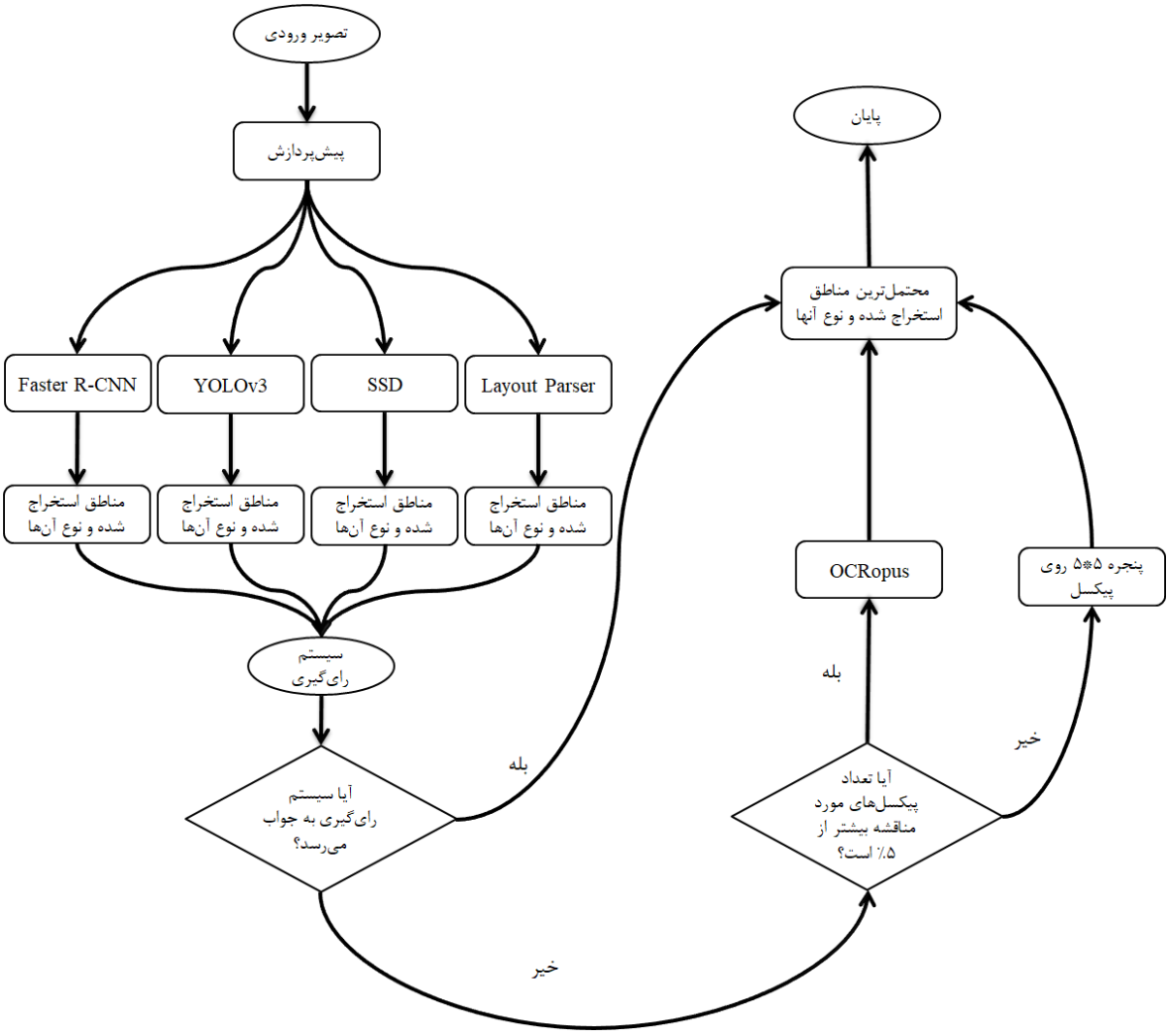

شکل ۱-۳: مراحل کلی سیستم DLA پیشنهادی

## ۳-۳ مرحله اول (استخراج مناطق متنی)

ما برای پیاده‌سازی روش پیشنهادی در مرحله اول، از چهار مدل از قبل آموزش دیده[1] Faster R-CNN، YOLOv3، SSD و Layout Parser بهره گرفتیم [۷۱, ۷۲]. این چهار مدل مبتنی بر یادگیری عمیق هستند که دقت و سرعت بالایی نسبت به دیگر روش‌های این حوزه دارند که در فصل دوم به شرح آن‌ها پرداخته شد. همان‌طور که در شکل ۲-۳ نشان داده شده، ابتدا تصاویر به صورت همزمان به عنوان ورودی به هر یک از این چهار روش داده می‌شود. خروجی هر یک از این چهار روش، یک دیتافریم است که اطلاعاتی نظیر مختصات کادر استخراج شده و نوع متنی یا غیرمتنی بودن منطقه را در خود جای داده

[1] Pre-trained models

است. پس از استخراج مناطق توسط این چهار روش، با بکارگیری یک روش رای گیری، محتمل‌ترین مناطق متنی و غیرمتنی بدست می‌آیند. همچنین در صورت برابر بودن آرا به سراغ مدل OCRopus که در فصل دوم شرح داده شده یا استفاده از پنجره ۵*۵ می‌رویم. مدل OCRopus با وجود دقت خوبی که دارد اما بسیار کندتر از دیگر روش‌ها است و فقط در شرایط خاص که تعداد پیکسل‌های دارای مناقشه زیاد است، به سراغ این روش می‌رویم. در ادامه به توضیح مراحل روش پیشنهادی خواهیم پرداخت.




شکل ۲-۳: مراحل مرحله اول روش پیشنهادی

## ۳-۳-۱ پیش‌پردازش

همان‌طور که گفته شد، چهار مدل ابتدایی که در رای‌گیری دخیل هستند، با مجموعه‌ای از دادگان از قبل آموزش دیده‌اند. اما برای استفاده از آن‌ها نیاز است که پیش‌پردازش‌های لازم را بر روی آن‌ها انجام دهیم. برای پیاده‌سازی و استفاده از مدل Layout Parser ابتدا نیاز است که Detectron2 را بر روی سیستم خود یا google colab نصب نماییم. Detectron2 سیستم نرم‌افزاری است که توسط محققین هوش مصنوعی شرکت فیس‌بوک[1] ارائه شده است و پیشرفته‌ترین الگوریتم‌های تشخیص اشیاء را پیاده‌سازی کرده است [۷۳]. در مرحله بعدی، نیاز به نصب کتابخانه layoutparser است که با نصب آن قادر به تشخیص مناطق حاوی متن خاوهیم بود. Parser ابزاری مبتنی بر یادگیری عمیق برای کارهای تجزیه و تحلیل طرح اسناد است. این ابزار که از طریق GitHub قابل دسترسی است، ابزاری نسبتا قدرتمند است که توانایی استخراج مناطق متنی و غیرمتنی از درون تصاویر حتی نسبتا پیچیده را نیز دارد.

علاوه بر مدل Layout Parser، ما از سه روش مبتنی بر یادگیری عمیق دیگر نیز بهره گرفتیم. مدل‌های پیاده‌سازی شده توسط این سه روش که عبارت‌اند از Faster R-CNN، YOLOv3 و SSD که همانند مدل Layout Parser از قبل آموزش دیده شده‌اند. این مدل‌ها توسط شرکت Monk AI پیاده‌سازی و آموزش دیده شده‌اند و در دسترس عموم قرار گرفته‌اند [۷۲]. Monk که یک مجموعه ابزار بینایی رایانه‌ای است، بیش از ۲۰ مدل را برای تشخیص اشیا پیاده‌سازی کرده و با آموزش آن‌ها، کار را برای دیگر محققین در این زمینه آسان‌تر کرده است. پس از انجام و نصب موارد گفته شده، مدل از قبل آموزش دیده شده مورد نظر را دریافت و همچنین آدرس کتابخانه‌هایش را به مسیر سیستم[2] اضافه می‌نماییم.

[1] Facebook
[2] System path

## ۳-۳-۲ ساخت، اجرا و آماده‌سازی سیستم جهت استفاده مناسب از مدل‌ها

همان‌طور که در الگوریتم ۱ نشان داده شده، مدل‌ها تلاش می‌کنند که محتمل‌ترین مناطق متنی و غیرمتنی تصویر را استخراج نمایند. پس از استخراج این مناطق توسط هر چهار مدل، نیاز به انجام تغییرات بر روی نتایج بدست آمده است. ما از هر مدل، اطلاعات یکسانی را استخراج می‌کنیم و درون یک دیتافریم ذخیره می‌نماییم. این دیتافریم شامل مختصات دو نقطه از کادر محصورکننده (مختصات بالاترین نقطه سمت چپ و پایین‌ترین نقطه سمت راست مشابه شکل ۳-۳ کادر محصورکننده قرمز) و نوع منطقه است.

الگوریتم ۱: مراحل پیاده‌سازی، اجرا و پس‌پردازش مدل‌های آنالیز قالب‌بندی

**Result:** text and non-text matrices
**Function** `FasterR-CNN(`$Input-image$`):`
  Input-image $\overrightarrow{\text{FasterR-CNN}}$ text-regions;
  text-regions $\overrightarrow{x1, y1, x2, y2, region-type}$ DataFrame;
  Text-matrix, Non-text-matrix = zero matrix with the same size of input-image $\rightarrow$ (h,w);
  **for** *each pixel of Input-image* **do**
    **if** *the pixel is in one of the text-region of the DataFrame* **then**
      Text-matrix [pixel's row, pixel's column] = 1
    **end**
    **if** *the pixel is in one of the non-text-region of the DataFrame* **then**
      Non-text-matrix [pixel's row, pixel's column] = 1
    **end**
  **end**
  **return** Text-matrix, Non-text-matrix;
**Function** `YOLOv3(`$Input-image$`):`
  .. like Faster R-CNN ..;
  **return** Text-matrix, Non-text-matrix;
**Function** `SSD(`$Input-image$`):`
  .. like Faster R-CNN ..;
  **return** Text-matrix, Non-text-matrix;
**Function** `LayoutParser(`$Input-image$`):`
  .. like Faster R-CNN ..;
  **return** Text-matrix, Non-text-matrix;
**Function** `Main:`
  Input-image = from dataset read an image;
  F-Text-matrix, F-Non-text-matrix = FasterR-CNN(Input-image);
  Y-Text-matrix, Y-Non-text-matrix = YOLOv3(Input-image);
  S-Text-matrix, S-Non-text-matrix = SSD(Input-image);
  L-Text-matrix, L-Non-text-matrix = LayoutParser(Input-image);
  Text-matrix = Combining all four text-matrices;
  Non-text-matrix = Combining all four non-text-matrices;


شکل ۳-۳: کادرهای محصورکننده تشخیص داده شده بر روی تصویر

در نتایج بدست آمده احتمال همپوشانی کامل دو منطقه وجود دارد، علی الخصوص در مدل Layout Parser که این میزان همپوشانی بسیار بیشتر است. همپوشانی کامل دو منطقه استخراج شده بدین صورت است که منطقه‌ای که به عنوان متن در نظر گرفته شده است به طور کامل درون یک منطقه متنی دیگر باشد. یا یک منطقه غیرمتنی با یک منطقه متنی همپوشانی داشته باشد. ما برای هر تصویر خروجی از هر چهار مدل، دو ماتریس هم‌اندازه تصویر اصلی می‌سازیم که مقادیر اولیه درایه‌های آن‌ها صفر است. این دو ماتریس نشان‌دهنده پیکسل‌هایی با برچسب متنی یا غیرمتنی هستند. به عنوان مثال اگر پیکسلی در یک منطقه متنی قرار داشته باشد، درایه متناظر آن در ماتریس متنی، برابر با عدد یک خواهد شد. همچنین اگر پیکسلی متعلق به بیش از یک منطقه باشد، همان مقدار یک را در درایه متناظر آن در ماتریس قرار خواهیم داد و اگر این دو منطقه متنی و غیرمتنی باشند، درایه متناظر آن پیکسل در هر دو ماتریس برابر با یک خواهد شد.

به عنوان نمونه، برای تصویر نشان داده شده در شکل ۴-۳، الگوریتم پیشنهادی ابتدا دو ماتریس متنی و غیرمتنی هم‌اندازه با تصویر اصلی خواهد ساخت و تمامی مقادیر درایه‌های آنها را صفر قرار می‌دهد. در مرحله بعد، منطقه‌ی پیکسل‌ها را مشخص کرده و درایه متناظر آن‌ها را در ماتریس متنی یا غیرمتنی را برابر با یک قرار خواهد داد. به عنوان مثال همانند شکل ۴-۳ پیکسل $(x_i,y_j)$ چون در هر دو منطقه متنی و غیرمتنی قرار دارد، درایه متناظرش در هر دو ماتریس برابر با یک خواهد شد. این دو ماتریس را برای خروجی هر چهار مدل ساخته و مقادیر درایه‌های آن‌ها را به همین نحوه قرار خواهیم داد. در نهایت با ترکیب چهار ماتریس متنی و جمع مقادیر درایه‌های آنها یک ماتریس نهایی برای مناطق متنی خواهیم داشت و به طور مشابه ماتریس متعلق به مناطق غیرمتنی را هم می‌سازیم.


شکل ۴-۳: چالش همپوشانی مناطق

## ۳-۳-۳ سیستم رای‌گیری

پس از مشخص شدن مقادیر درایه‌های دو ماتریس متنی و غیرمتنی برای یک تصویر، به سراغ رای‌گیری می‌رویم. در این مرحله می‌بایست محتمل‌ترین مناطق متنی و غیرمتنی را به کمک دو ماتریس بخش قبل

استخراج نماییم. مقادیر درون درایه‌های این دو ماتریس، حداقل صفر و حداکثر چهار هستند. به عنوان مثال اگر مقدار درایه‌ای مانند (i,j) در درون ماتریس غیرمتنی، برابر با سه و در درون ماتریس متنی دو باشد، یعنی سه روش پیکسل (xi,yj) تصویر اصلی را غیرمتنی و دو روش آن را متنی تشخیص داده‌اند. تعداد روش‌های رای‌گیری چهار است، اما گاهی مجموع مقادیر متنی و غیرمتنی از چهار بیشتر می‌شود. دلیل به وجود آمدن این اتفاق وجود همپوشانی مناطق غیر هم‌نوع در یکی از روش‌ها است. در صورتی که یک پیسکل در هر دو ماتریس متنی و غیرمتنی یک روش جای گیرد این اتفاق رخ خواهد داد.

در سیستم رای‌گیری باید مشخص کنیم که هر پیکسل و به‌تبع آن هر درایه از دو ماتریس متنی و غیرمتنی که مقادیری به جز صفر دارند، در کدام یک از دو نوع منطقه متنی و غیرمتنی قرار خواهد گرفت. در صورتی که اختلاف مقادیر درایه‌های دو ماتریس در یک پیکسل، بزرگتر یا مساوی یک باشد، سیستم رای‌گیری آن پیکسل را در زمره منطقه‌ای قرار خواهد داد که مقدار درایه ماتریسش بزرگتر باشد. به عنوان نمونه اگر مقدار پیکسل ($x_i,y_j$) در درایه متناظرش در ماتریس متنی برابر با سه و در ماتریس غیرمتنی برابر با یک باشد، این پیکسل به عنوان یکی از پیکسل‌های منطقه متنی قلمداد خواهد شد.

اما در غیر این صورت سیستم رای‌گیری به مشکل بر می‌خورد. دلیل این اتفاق ضعیف‌تر بودن مدل Layout Parser به نسبت سه مدل دیگر است. در این مدل علاوه بر همپوشانی زیاد مناطق، میزان دقت تشخیص درست مناطق نیز پایین‌تر است. لذا پس از مشخص شدن نوع مناطق، بعضی از پیکسل‌ها تعیین تکلیف نمی‌شوند که به آنها پیکسل مورد مناقشه می‌گوییم. در صورتیکه تعداد پیکسل‌های مورد مناقشه بیش از ۵٪ پیکسل‌های تعیین تکلیف شده باشد، از یک مدل دیگر نیز کمک می‌گیریم.

این مدل که OCRopus نام دارد، دقت بالایی در تشخیص مناطق حاوی متن دارد ولی همان‌طور که در جدول ۱-۳ نشان داده شده است، نسبت به چهار روش گفته شده بسیار کندتر است. بعلاوه تصاویر ورودی این مدل حتما می‌بایست وضوح 300 dpi داشته باشند، در غیر این صورت امکان استفاده از این مدل وجود ندارد. به همین سبب ما در سیستم رای‌گیری از این مدل استفاده نکرده‌ایم و تنها در صورت نیاز به سراغ

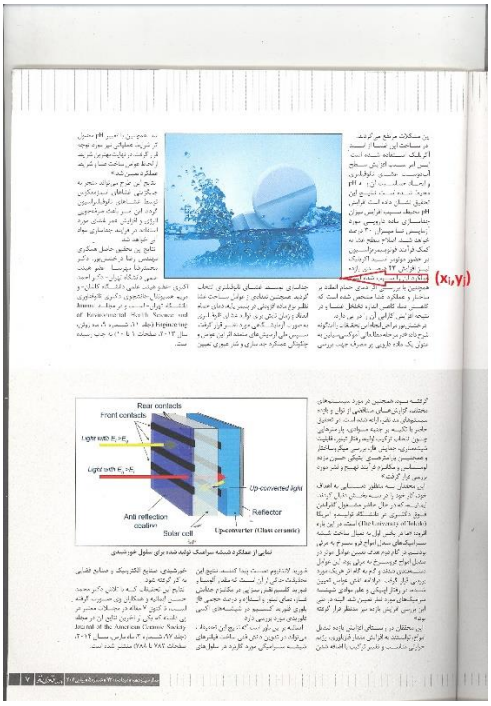
این مدل خواهیم رفت. بدین صورت قادر خواهیم بود تا محتمل‌ترین مناطق متنی و غیرمتنی درون تصویر را تشخیص و جدا کنیم. نکته حائز اهمیت، اجرای همزمان چهار مدل اصلی سیستم رای‌گیری است. یعنی هر تصویر در کمتر از ۲.۵ ثانیه وارد سیستم رای‌گیری می‌شود که باعث افزایش چشم‌گیر سرعت روش‌پیشنهادی در مقایسه با استفاده از مدل OCRopus است.

جدول ۱-۳: میانگین زمان اجرا برای هر تصویر در پنج مدل استفاده شده روش پیشنهادی

| نام مدل | میانگین زمان اجرا برای هر تصویر (s) |
|---|---|
| Layout Parser | ۱.۴۴ |
| SSD | ۰.۵۰۷ |
| YOLOv3 | ۱.۳۹ |
| Faster R-CNN | ۲.۲۳۸ |
| OCRopus | ۱۸.۰۵۱ |

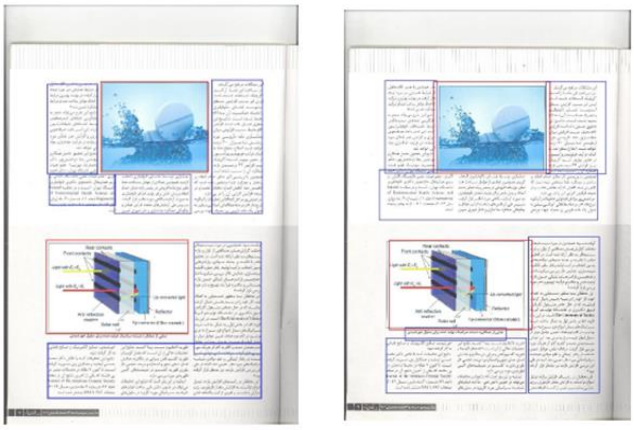
الگوریتم ۲: مراحل رای‌گیری

```
for each pixel of text-matrix do
    if Text-matrix [pixel's row, pixel's column] > Non-text-matrix [pixel's
     row, pixel's column] then
        pixel considered as text-region
    end
    if Text-matrix [pixel's row, pixel's column] < Non-text-matrix [pixel's
     row, pixel's column] then
        pixel considered as non-text-region
    end
    if Text-matrix [pixel's row, pixel's column] == Non-text-matrix
     [pixel's row, pixel's column] then
        Using OCRopus
    end
end
```

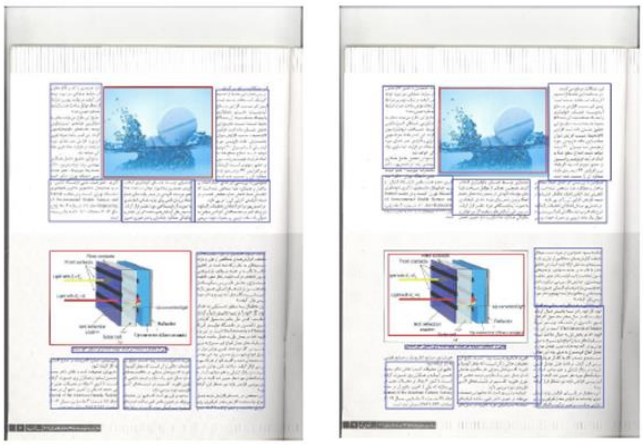



شکل ۵-۳: نمونه‌ای از پیکسل‌های مورد مناقشه

در صورتیکه تعداد پیکسل‌های مورد مناقشه کمتر از ۵٪ پیکسل‌های تعیین تکلیف شده باشد، از مدل OCRopus بهره گرفته نمی‌شود. در این موارد برای پیکسل‌های مورد مناقشه، پنجره‌ای ۵*۵ به مرکز پیکسل مورد نظر در نظر گرفته و نتیجه رای‌گیری روی این پنجره را مبنا قرار می‌دهیم. به عنوان نمونه شکل ۵-۳ را در نظر بگیرید که در آن تعداد پیکسل‌های مورد مناقشه در کل تصویر بسیار اندک است، لذا نیازی به استفاده از OCRopus نیست. سیستم ابتدا همانند شکل ۶-۳ خروجی هر یک از چهار روش را بدست می‌آورد. حال یکی از پیکسل‌های مورد مناقشه را در نظر می‌گیریم. سیستم پنجره‌ای ۵*۵ به مرکز پیکسل مورد نظر باز می‌کند و مقادیر درایه‌های متناظر پنجره را در دو ماتریس‌های متنی و غیرمتنی، بدست می‌آورد. نتایج پیش‌بینی چهار روش، در این پنجره ۵*۵ در جدول ۲-۳ و جدول ۳-۳ آورده شده است. همان‌طور که مشاهده می‌شود، مجموع تعداد آرای غیرمتنی بیشتر است، لذا این پیکسل در زمره پیکسل‌های غیرمتنی قرار خواهد گرفت. پس از انجام این‌کار تمامی پیکسل‌های درون تصویر تعیین وضعیت شده و بخش متنی و غیرمتنی درون تصویر جدا می‌شوند.



الف: خروجی Layout Parser ب: خروجی Yolov3



ج: خروجی SSD د: خروجی Faster R-CNN

شکل ۶-۳: خروجی چهار مدل ارائه شده بر روی یک نمونه تصویر

جدول ۲-۳: جمع مقادیر درایه‌های متناظر پنجره ۵*۵ در چهار ماتریس غیرمتنی

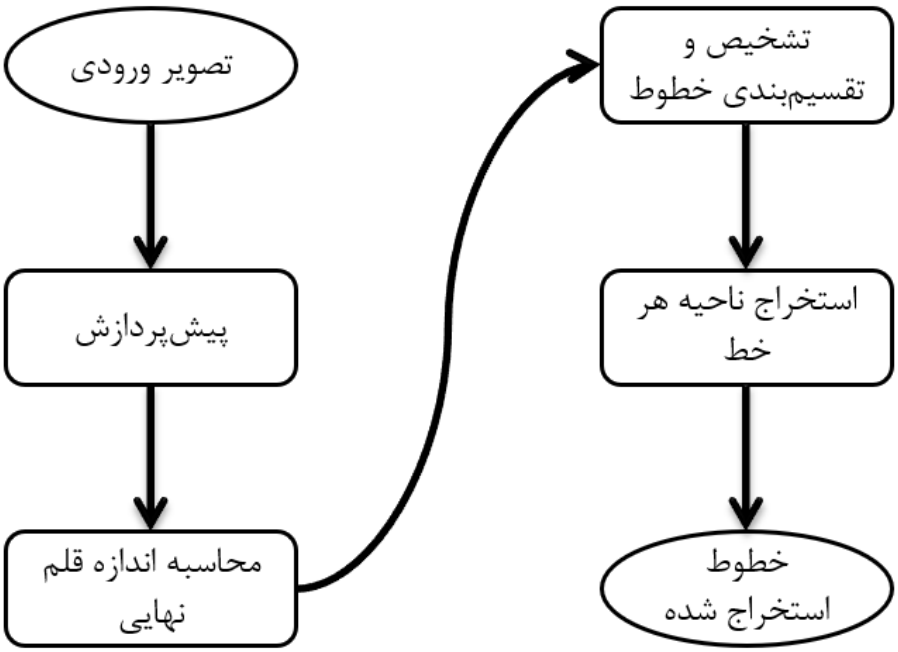
| Non-text sum=44 | j-2 | j-1 | j | j+1 | j+2 |
|---|---|---|---|---|---|
| i-2 | 3 | 3 | 2 | 1 | 1 |
| i-1 | 3 | 3 | 2 | 1 | 1 |
| i | 3 | 3 | 2 | 1 | 1 |
| i+1 | 2 | 2 | 1 | 1 | 1 |
| i+2 | 2 | 2 | 1 | 1 | 1 |

جدول ۳-۳: جمع مقادیر درایه‌های متناظر پنجره ۵*۵ در چهار ماتریس متنی

| text<br>sum=34 | j-2 | j-1 | j | j+1 | j+2 |
|---|---|---|---|---|---|
| i-2 | 0 | 0 | 0 | 1 | 1 |
| i-1 | 1 | 1 | 2 | 2 | 2 |
| i | 1 | 1 | 2 | 2 | 2 |
| i+1 | 1 | 1 | 2 | 2 | 2 |
| i+2 | 1 | 1 | 2 | 2 | 2 |

## ۳-٤ مرحله دوم (استخراج خطوط)

همان‌طور که فصل دوم بیان شده، یکی از انواع الگوریتم‌ها در زمینه استخراج خطوط، الگوریتم‌های پایین به بالا هستند که الگوریتم ارائه شده در مرحله دوم، دارای همین طراحی است. همان‌طور که در شکل ۳-۷ نشان داده شده، الگوریتم پیشنهادی، ابتدا تمامی CCهای را پیدا می‌کند و سپس CCهایی که به صورت افقی در یک راستا قرار دارند را به یکدیگر متصل می‌کند. درنهایت الگوریتم برای هر خط بهترین ناحیه را استخراج و در یک تصویر جدا ذخیره می‌کند. همان‌طور که در بخش‌های قبل نیز گفته شد، CC به مجموعه‌ای از پیکسل‌های سفید (یا سیاه) گفته می‌شود که به یکدیگر متصل هستند و یک نقطه، اعراب، حرف، یک و یا بخشی از یک کلمه را تشکیل می‌دهند. به عنوان مثال کلمه سلام از دو CC تشکیل می‌شود و "سلا" و "م" هر یک بیانگر یک CC خواهند بود. در ادامه این فصل به توضیح بخش‌های این مرحله می‌پردازیم.




شکل ۷-۳: مراحل مرحله دوم روش پیشنهادی

## ۳-٤-۱ پیش‌پردازش

مرحله پیش‌پردازش یکی از مراحل مهم و اساسی در زمینه تشخیص خطوط حاوی متن از درون تصویر است. مرحله پیش‌پردازش یک سیستم تشخیص خط، شامل عملیاتی است که بر روی تصویر ورودی انجام داده می‌شود تا یک تصویر مناسب، برای شناسایی خطوط حاصل شود. همان‌طور که در الگوریتم ۳ نشان داده شده است، در این مرحله، ابتدا اگر تصویر سه کاناله (RGB)[1] باشد آن را به یک کانال خاکستری تبدیل می‌کنیم و سپس با استفاده از تعیین یک حد آستانه، تصویر را دودویی می‌کنیم. در بخش دوم از مرحله پیش‌پردازش، رفع نویز را خواهیم داشت. نهایتا خطوط زائد درون تصویر را تا حد امکان حذف می‌کنیم.

[1] Red, Green and Blue

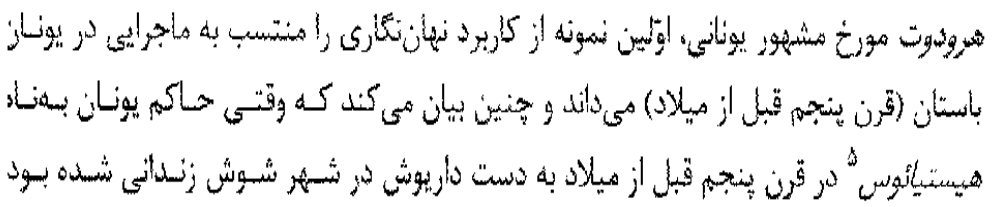
الگوریتم ۳: شبه کد مربوط به مراحل پیش‌پردازش

```
Result: The clean image
Function Binary(Input − image):
    Input-image rgbTogray grayscale-image;
    grayscale-image OtsuMethod threshold;
    threshold, grayscale-image grayTobinary binary-image;
    return binary-image;
Function
 median-filter(binary − image,Neighborhood − size : [M,M]):
    for i=1:Number of binary image pixels do
        Value of pixel(i) = median value in a M-by-M neighborhood
         around pixel(i)
    end
    return modified-binary-image;
Function
 Initial-labeling(modified − binary − image,epsilon − value : ε):
    Initial labeling to each CC;
    measuring the MajorAxisLength and MinorAxisLength of each label;
    for i=1:Number of labels do
        if MajorAxisLength(i) / MinorAxisLength(i) > ε or MajorAxisLength(i) == 1 or
         MinorAxisLength(i) == 1 then
            remove label(i)
        end
    end
    return clean-image;
Function Main:
    binary-image=Binary(Input − image);
    modified-binary-image=median-filter(binary − image,[3,3]);
    clean-image=Initial-labeling(modified − binary − image,15);
```

### ۳-٤-۱-۱ دودویی‌سازی

با استفاده از یک تصویر دودویی، فرآیندهای پردازش بر روی تصویر بسیار راحت‌تر و با سرعت بالاتری انجام می‌پذیرند. دودویی‌سازی تصویر در روال الگوریتم تشخیص خطوط حاوی متن بسیار موثر است [۷۴, ۷۵]. برحسب کاربرد الگوریتم، در بسیاری از موارد نیاز به تبدیل تصویر از نوعی به نوع دیگر است. برای دودویی‌سازی تصویر می‌بایست ابتدا تصاویر با سه کانال RGB به تصاویر تک کاناله خاکستری تبدیل شوند. سپس با استفاده از یکی از روش‌های آستانه گذاری، یک حد آستانه برای تصویر پیدا کرده و بر اساس آن حد آستانه، تصاویر خاکستری را به دودویی تبدیل شود.

در بسیاری از روش‌های دودویی‌سازی همچون روش این رساله، برای پیدا کردن بهینه‌ترین مقدار حد آستانه، از روش آستانه‌گذاری به نام Otsu استفاده شده است. روش Otsu، بر اساس بیشینه کردن واریانس تصاویر عمل می‌کند. یعنی این روش مقداری را برای آستانه انتخاب می‌کند که قطعه‌بندی انجام شده در ناحیه با واریانس (اختلاف)زیاد نسبت به هم به‌دست می‌آید [۳۵]. در شکل ۸-۳ نمونه‌ای از تبدیل یک تصویر خاکستری به یک تصویر دودویی نشان داده شده است.


هرودوت مورخ مشهور یونانی، اولین نمونه از کاربرد نهان‌نگاری را منتسب به ماجرایی در یونان باستان (قرن پنجم قبل از میلاد) می‌داند و چنین بیان می‌کند که وقتی حاکم یونان به‌نام هیستیائوس[5] در قرن پنجم قبل از میلاد به دست داریوش در شهر شوش زندانی شده بود

شکل ۸-۳: نمونه‌ای از تبدیل یک نصویر خاکستری به تصویر دودویی

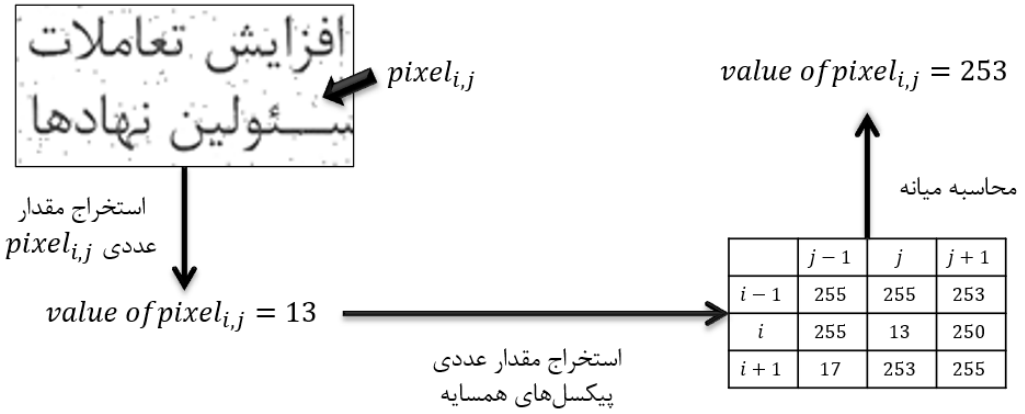
### ۳-٤-۱-۲ رفع نویز

به‌طور کلی نویز در تصویر، مشخصه‌هایی از تصویر است که موجب منحرف شدن الگوریتم در تشخیص مشخصه‌های اصلی تصویر می‌شود. با وجود نویز در تصویر، تحلیل و تفسیر تصاویر با مشکل روبرو می‌شود. لذا یکی از پیش‌پردازش‌های اساسی در حوزه پردازش تصویر، رفع نویز است [۳۵].

گاهی اوقات تصویر به دلیل کیفیت بد اسکنر، نویزدار شود. در این حالت، با استفاده از یک فیلتر مناسب می‌توان نویز درون تصویر را حذف نمود. از فیلترهای رفع نویز معروف می‌توان به فیلتر میانگین[1]، فیلتر میانه[2]، فیلتر گوسین[3] و فیلتر وینر[4] اشاره کرد [۷۶-۷۹]. اعمال هر فیلتر علاوه بر حذف احتمالی نویز در تصویر، ممکن است اثرات جانبی مشخصی در تصویر ایجاد کند. بنابراین، استفاده درست از فیلتر مناسب در هر فرایند مشخص، بسیار حائز اهمیت است. در این تحقیق ما از فیلتر میانه استفاده نموده‌ایم که

[1] Average Filter
[2] Median Filter
[3] Gaussian Filter
[4] Winner Filter

کمترین اثرات جانبی را نسبت به بقیه فیلترها بر روی تصویر داشته است. فیلتر میانه در حذف نویزها، مخصوصا نویز فلفل نمکی بسیار خوب عمل می‌کند [۸۰]. در این نوع فیلتر یک شعاع همسایگی تعریف می‌گردد و مقدار عددی هر پیکسل در تصویر، برابر با میانه اعداد پیکسل‌ها در این شعاع همسایگی خواهد بود. همان‌طور که در شکل ۹-۳ نشان داده شده است، در $pixel_{i,j}$ (پیکسلی که در سطر iام و ستون jام از تصویر قرار دارد) با تعریف شعاع همسایگی ۱، مربعی سه در سه با مرکزیت این پیکسل ایجاد می‌شود. با اعمال فیلتر میانه بر روی تصویر، مقدار عددی $pixel_{i,j}$ برابر با میانه اعداد درون این مربع می‌شود.




شکل ۹-۳: نحوه کار فیلتر میانه

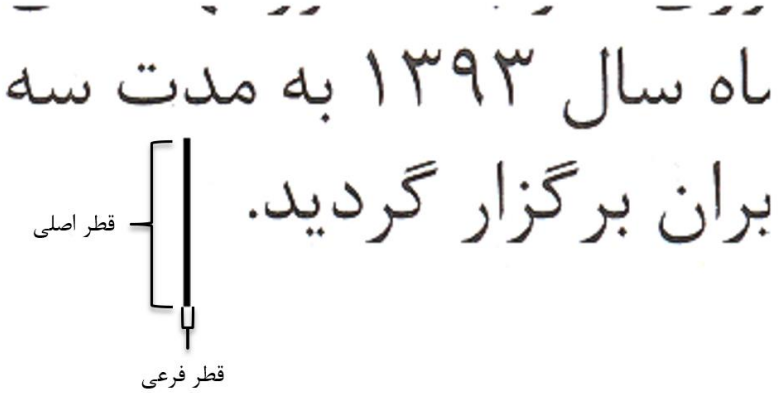
این عملیات بر روی تمامی پیسکل‌های درون تصویر انجام می‌شود و همان‌طور که در شکل ۱۰-۳ نشان داده شده است، این فیلتر به خوبی توانسته است نویزهای موجود در تصویر را بدون از بین بردن اطلاعات مهم در تصویر، حذف نماید.

الف: تصویر نویزدار ب: تصویر رفع نویز شده

شکل ۱۰-۳: رفع نویز با کمک فیلتر میانه

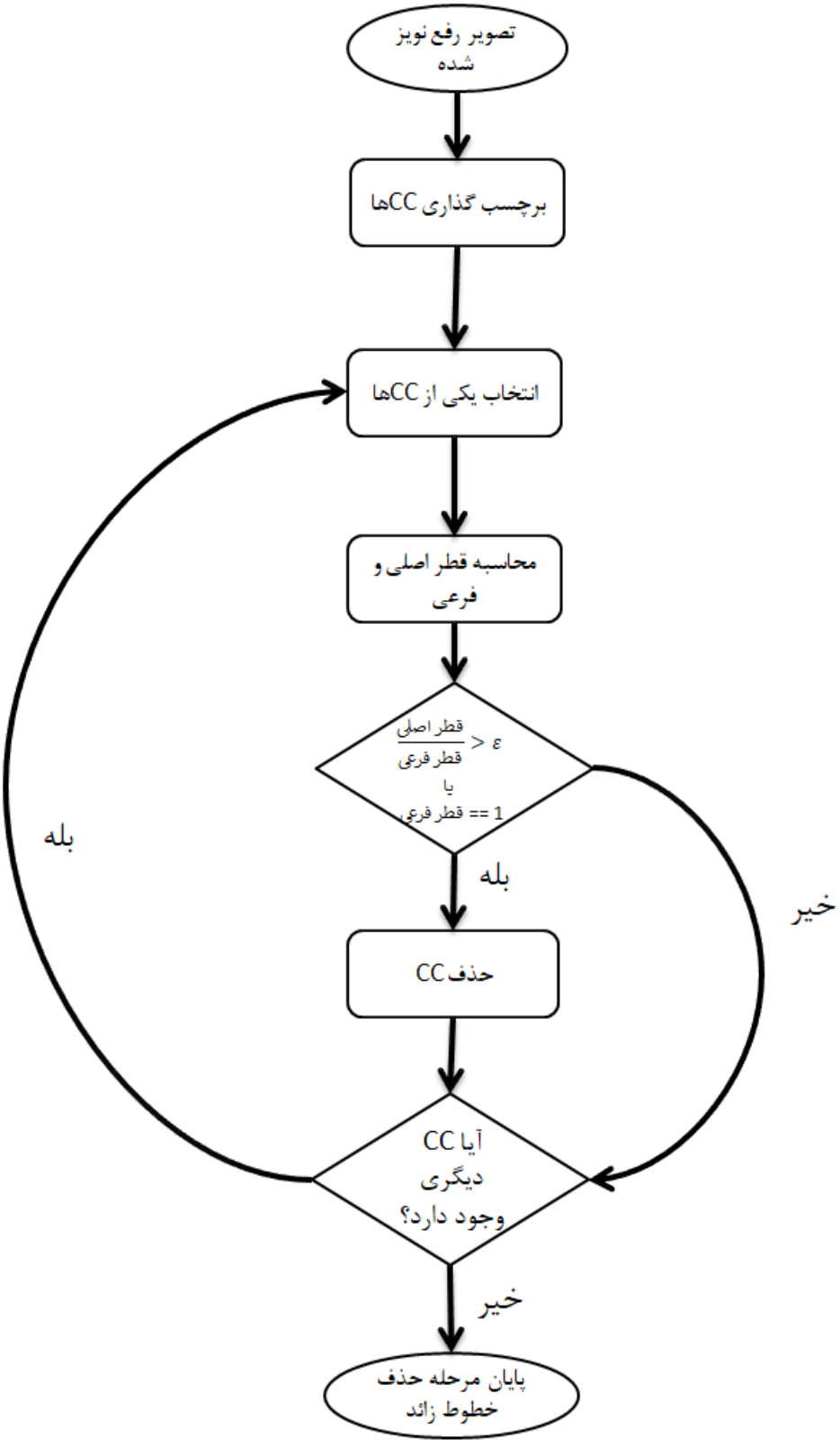
### ۳-۱-٤-۳ حذف خطوط زائد

گاهی اوقات بر روی تصویر، خطوط زائدی وجود دارد که در صورت عدم حذف آنها برای الگوریتم تشخیص دهنده خط، مشکلاتی را به وجود خواهند آورد. در همین راستا، یکی از بخش‌های مرحله پیش‌پردازش مرحله‌ی دوم سیستم DLA خود را به حذف این خطوط اختصاص دادیم. در این بخش به سراغ تک‌تک CCها می‌رویم و پس از برچسب‌گذاری آنها، یک سری اطلاعات مرتبط با اندازه آنها را استخراج می‌نماییم. این اطلاعات اندازه قطر اصلی و اندازه قطر فرعی هر CC هستند. در شکل ۱۱-۳ نمونه‌ای از قطر اصلی و فرعی نشان داده شده است.




شکل ۱۱-۳: نمایش قطر اصلی و فرعی

همان‌طور که در شکل ۱۲-۳ نشان داده شده، در هر CC، اگر قطر فرعی برابر یک پیکسل باشد یا نسبت قطر اصلی به فرعی بزرگتر از $\varepsilon$ باشد، برچسب این CC با عدد صفر مقداردهی می‌شود. یا به عبارتی، برچسب مورد نظر حذف و خطوط زائد به رنگ پس زمینه در می‌آیند. مقدار $\varepsilon$ با توجه به تصاویر دادگان موجود و با آزمون و خطای فراون ۱۵ بدست آمده است.




شکل ۱۲-۳: مراحل حذف خطوط زائد

### ۳-٤-۱-٤ مکمل تصویر

از کاربردهای اساسی مکمل‌سازی تصویر، بهبود اطلاعات درون تصویر است. در این تحقیق پیکسل‌های سفید برای ما مهم هستند، چرا که این پیکسل‌ها مقدار ۱ یا همان ۲۵۵ را پس از دودویی‌سازی تصویر خواهند داشت و می‌توان محاسبات دلخواه را روی آنها انجام داد. لذا در تصاویری که پس‌زمینه سفید و CCها سیاه هستند، نیاز است که از عملیات مکمل‌سازی تصویر بهره گرفت. در شکل ۱۳-۳ روال کلی بخش پیش‌پردازش بر روی یک عکس نشان داده شده است.


الف: تصویر ورودی مرحله پیش‌پردازش

ب: تصویر رفع نویز شده

ج: حذف خطوط زائد

د: مکمل‌سازی

شکل ۱۳-۳: نمونه‌ای از اجرای مراحل پیش‌پردازش بر روی تصویر

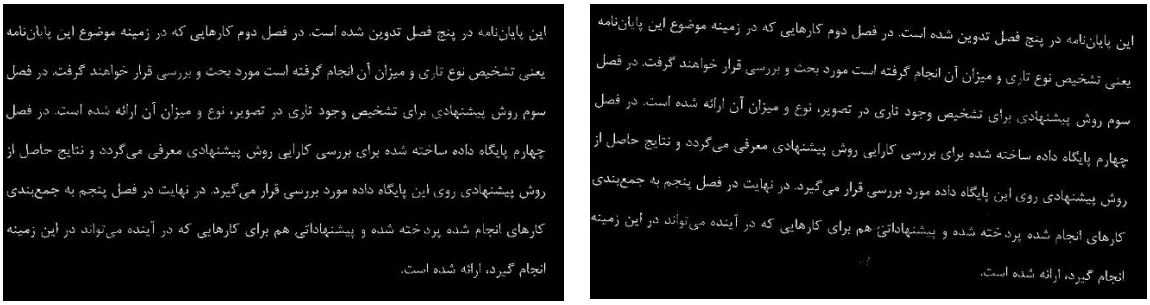
## ۳-٤-۲ تصحیح زاویه تصویر

تصحیح زاویه تصویر یکی از مراحل مهم در تشخیص و جداسازی خطوط حاوی متن است. روش پیشنهادی در این تحقیق با استفاده از تکنیک تصحیح زاویه، تصاویر کج را اصلاح نموده است. همان‌طور که در شکل ۱۴-۳ الف مشاهده می‌شود، تصویر ورودی دارای زاویه زیادی است که دقت الگوریتم تشخص خط را پایین می‌آورد. همان‌طور که در شکل ۱۴-۳ ج مشاهده می‌شود، تعداد سطرهایی که در هیستوگرام مقداری نزدیک به صفر دارند بسیار کمتر از حالت اصلاح شده است. مقدار هر سطر از این هیستوگرام، بیانگر تعداد

پیکسل‌های متنی (مقدار ۲۵۵ یا ۱) در آن سطر است. لذا اگر تصویر زاویه نداشته باشد بدنه اصلی هر خط در سطرهای کمتری قرار می‌گیرند و می‌توان تصویر را با توجه به آن اصلاح نمود.

همانطور که در الگوریتم ۴ مراحل این بخش آورده شده است، برای تصحیح زاویه تصویر، تصویر را از منفی ۱۰ تا مثبت ۱۰ درجه می‌چرخانیم و در هر زاویه تعداد سطرهای نزدیک به صفر در هیستوگرام را بدست می‌آوریم. دلیل بیان عبارت نزدیک به صفر، بخاطر وجود بعضی از نویزها و یا تعداد کمی پیکسل سفید در هر سطر است.

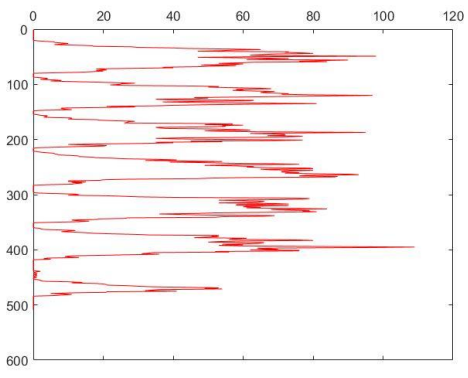
پس از اتسخراج تعداد سطرهای نزدیک به صفر در هر زاویه، بهترین زاویه بدست می‌آید. بدیهی است آن زاویه‌ای که تعداد سطر نزدیک به صفر بیشتری را در هیستوگرامش داشته باشد، زاویه‌ی بهتری است. به صورت مشابه این کار را برای زاویه‌های ۰.۱ تکرار می‌کنیم تا دقیق‌ترین زاویه استخراج شود و در نهایت بر اساس زاویه نهایی تصویر اصلاح می‌گردد.



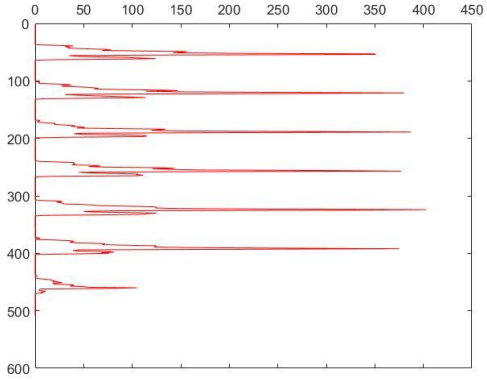

الف: تصویر کج




ب: تصویر اصلاح شده




ج: هیستوگرام افقی تصویر کج




د: هیستوگرام افقی تصویر اصلاح شده

شکل ۱۴-۳: مراحل تصحی زاویه تصویر

الگوریتم ۴: مراحل تصحیح زاویه

```
Result: non-skewed image
Function projection(binary-image):
    W=binary-image's width;
    projection=Number of white pixels per row in binary-image;
    x=0;
    for i=1:len(projection),i++ do
        if projection[i] <= 0.01 * W then
            x+=1
        end
    end
    return x;
End Function
Function main(binary-image):
    best-degree=none;
    x=0;
    for i=-10:10,i++ do
        rotated-image=imrotate(binary-image,i);
        x_new=projection(rotated-image);
        if x < x_new then
            x=x_new;
            best-degree=i;
        end
    end
    Fbest-degree=best-degree;
    for i=Fbest-degree-1:Fbest-degree+1,i+=0.1 do
        rotated-image=imrotate(binary-image,i);
        x_new=projection(rotated-image);
        if px < x_new then
            x=x_new;
            best-degree=i;
        end
    end
    non-skewed-image=imrotate(binary-image,best-degree);
    return non-skewed-image;
End Function
```

## ۳-٤-۳ محاسبه اندازه قلم نهایی

پس از اجرا پیش‌پردازش‌های لازم بر روی تصویر، در این مرحله اندازه قلم نهایی تصویر استخراج می‌شود.

با استفاده از اندازه قلم نهایی استخراج شده از تصویر می‌توان CCهای هم‌راستا را به یکدیگر متصل نمود

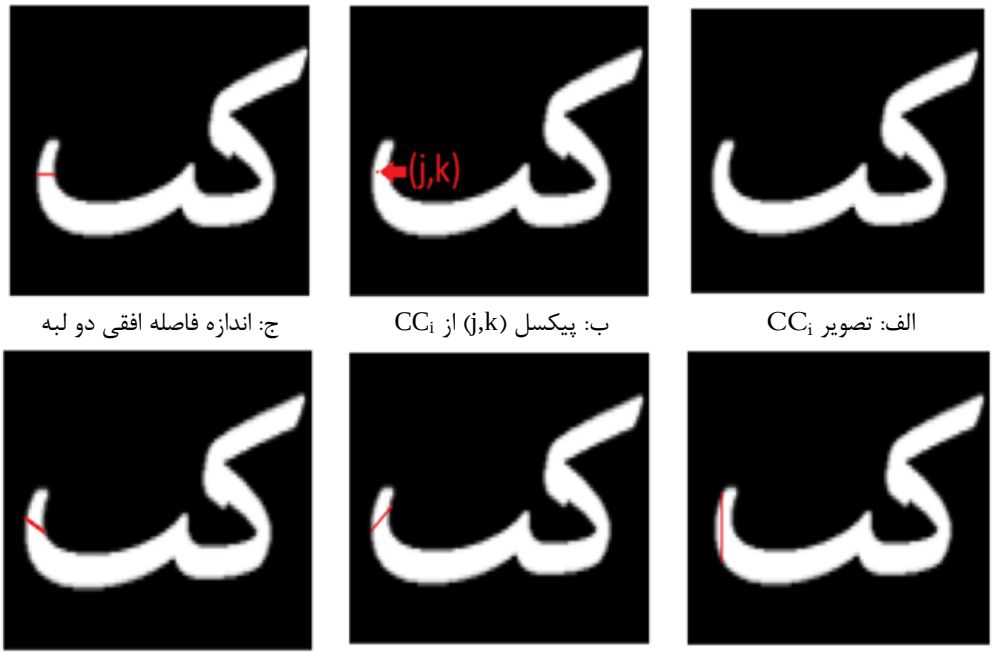
که در بخش بعد به توضیح آن می‌پردازیم. اما همانطور که گفته شد، هدف این بخش، استخراج قلم نهایی تصویر است. الگوریتم پیشنهادی، کار خود را در این مرحله با محاسبه پهنای تک تک قسمت‌های هر یک از CCها شروع می‌کند و پس از استخراج پهنای CCهای درون تصویر، شروع به محاسبه و استخراج یک اندازه قلم از تصویر می‌کند که این اندازه را اندازه نهایی قلم می‌نامیم. در ادامه به توضیح مراحل این بخش خواهیم پرداخت.

### ۳-٤-۳-۱ محاسبه پهنای CC

در این مرحله الگوریتم باید برای تک‌تک CCهای درون تصویر، یک اندازه پهنا استخراج کند که این اندازه نشان دهنده پهنای قالب یک CC است. در الگوریتم ۵ مراحل این بخش به‌صورت شبه کد نشان داده شده است. برای محاسبه اندازه پهنای هر یک از CCهای درون تصویر، همانطور که در شکل ۱۵-۳ نشان داده شده، از جهت‌های مختلف به تک‌تک پیکسل‌های درون آن CC نزدیک شده و پهناهای مربوط به هر پیکسل، در جهات مختلف محاسبه می‌شود. این پهنا، نشان‌دهنده فواصل بین لبه‌ها نیز هستند. پهنای هر پیکسل کمینه پهنا به دست آمده در آن پیکسل است. علاوه بر این، در زبان فارسی پهناهایی که به دفعات تکرار شده اند، بیانگر قطرهای بهینه هستند.

الگوریتم ۵: شبه کد مربوط به محاسبه پهنای CC

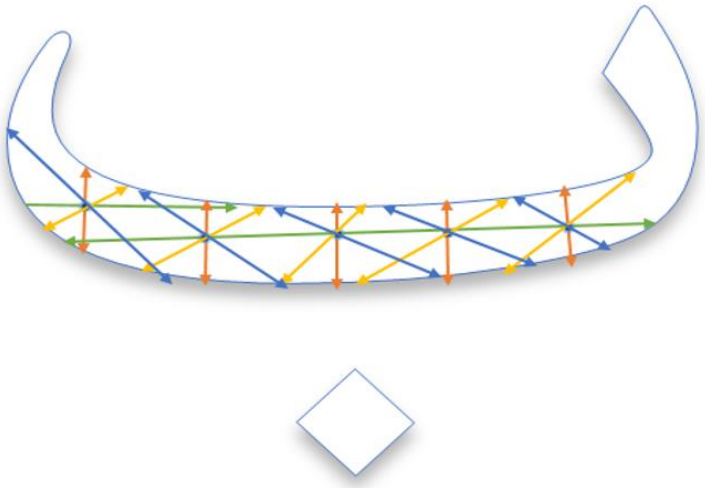
```
for i=1:Number of CCs do
    foreach pixel of CC_i do
        HD= horizontal distance between two edges of CC_i;
        VD= vertical distance between two edges of CC_i;
        OD= oblique distance between two edges of CC_i;
        CC_i's diameter of the pixel= minimum(HD,VD,OD);
    end
    final CC_i's diameter= the largest number of repetitions of the diameter
      among the CC_i;
    foreach pixel of CC_i do
        CC_i's diameter of the pixel=final CC_i's diameter;
    end
end
```




الف: تصویر $CC_i$ ب: پیکسل (j,k) از $CC_i$ ج: اندازه فاصله افقی دو لبه

د: اندازه فاصله عمودی دو لبه ه: اندازه فاصله اوریب ۱ دو لبه و: اندازه فاصله اوریب ۲ دو لبه

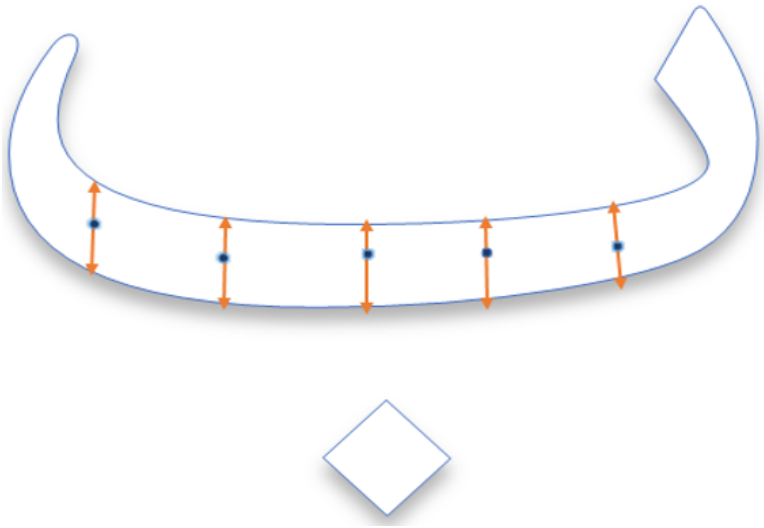
شکل ۱۵-۳: نحوه استخراج اندازه پهناهای پیکسل (j,k)



شکل ۱۶-۳: نحوه نمایش به‌دست آوردن پهناهای CC برای هر پیکسل

همان‌طور که گفته شد، پس از بدست آوردن فواصل بین لبه‌ها به مرکزیت یک پیکسل خاص، کمینه این فواصل به عنوان پهنا CC در آن پیکسل در نظر گرفته می‌شود که به اختصار آن را پهنای پیکسل می‌نامیم. برای درک بهتر دلیل در نظر گرفتن کمینه فواصل به عنوان پهنا پیکسل، شکل ۱۶-۳ را در نظر بگیرید. همانطور که مشاهده می‌شود، برای ۵ پیکسل فواصل دو لبه استخراج شده‌اند. همان‌طور که مشاهده می‌شود،

کمترین فاصله بین دو لبه، در بین بقیه فواصل بیانگر دقیق‌تری از پهنای پیکسل است. شکل ۱۷-۳ پهناهای مناسب را نشان داده است.



شکل ۱۷-۳: نمایش پهناهای مناسب

پس از انجام این مرحله، آن پهنای پیکسلی که بیشترین تعداد تکرار را در $CC_i$ داشته، به عنوان پهنای $CC_i$ در نظر گرفته خواهد شد. در نهایت مقدار پهنای تمامی پیکسل‌های درون $CC_i$، برابر با اندازه پهنای $CC_i$ می‌شوند. به عنوان مثال CC درون شکل ۱۸-۳ را در نظر بگیرید که پهنای تمامی پیکسل‌های درون آن استخراج شده‌اند. همانطور که در جدول ۴-۳ نشان داده شده است، پهنای اکثر پیکسل‌ها برابر با ۲ است. لذا پهنای قالب این CC برابر با دو است. به همین دلیل، مابقی پیکسل‌هایی که پهنایی به جز ۲ دارند نیز، برایشان پهنای ۲ در نظر گرفته می‌شود. در نهایت در جدول ۵-۳ تعداد پیکسل‌ها و پهنای قالب این CC نشان داده شده است. این عملیات را برای تمامی CCهای درون تصویر تکرار می‌کنیم و پس از استخراج پهنای هر یک از آنها، به مرحله بعد می‌رویم.

شکل ۱۸-۳: نمونه‌ای از یک CC

جدول ۴-۳: خروجی بخش پهنای پیکسل

| تعداد پیکسل‌ها | پهنای پیکسل |
|---|---|
| ۲۷۵ | ۲ |
| ۱۳ | ۲.۸ |
| ۴۵ | ۳ |

جدول ۵-۳: اطلاعات نهایی CC

| تعداد پیکسل‌ها | پهنای پیکسل |
|---|---|
| ۳۳۳ | ۲ |

### ۳-٤-۳-۲ اندازه نهایی قلم

در این مرحله، الگوریتم پیشنهادی تمامی پهناهای موجود در تصویر به همراه تعداد پیکسل‌های مربوط به آنها را دریافت می‌کند. سپس پهنایی که بیشترین تعداد تکرار را داشته باشد، به عنوان اندازه قلم اولیه تصویر در نظر می‌گیرد. در انتخاب این اندازه به عنوان اندازه قلم اولیه ممکن است نقاط، اعراب و حرکات کلمات موثر باشند، لذا نیاز به حذف آنها است. چرا که نقاط ریز برای ما از اهمیت بالایی برخوردار نیستند و در صورت عدم حذف آنها، الگوریتم در تشخیص درست موقعیت خطوط، دچار خطا شود. همانطور که در الگوریتم ۶ نشان داده شده، اگر تعداد پیکسل‌های یک CC از ده برابر اندازه قلم اولیه تصویر کوچک‌تر باشد، آن CC حذف می‌شود و در محاسبات بعدی تاثیری نخواهد داشت. همچنین این عدد ده، پس از سعی و خطاهای بسیار بر روی انواع مختلفی از تصاویر بدست آمده است تا از کوچک‌ترین خطاها نیز جلوگیری شود. بعلاوه، درصورت عدم حذف بعضی از نقاط به روال کلی الگوریتم ضربه‌ای وارد نمی‌شود و حذف نقاط کوچک فقط باعث بهبود عملکرد الگوریتم می‌شود.

پس از حذف این CCها و به‌تبع آن کم شدن تعداد تکرار پهناهای مختلف یا حتی حذف آنها، الگوریتم پیشنهادی پهناهای باقیمانده را بر اساس اندازه مرتب کرده و پهنایی با بیشترین تعداد تکرار را به عنوان اندازه قلم ثانویه تصویر در نظر می‌گیرد.

الگوریتم ۶: شبه کد مربوط به مراحل محاسبه اندازه نهایی قلم تصویر

```
Function final-font-sizes(UDiameters,UDiametersNum,SecondFontSize):
    BodyFontSize = SecondFontSize;
    for i=1:Number of UDiameters do
        if UDiameter_i > (2 × SecondFontSize) then
            *The image has a big title*;
            TitleFontSize = The most number of repetition of the
             remaining UDiameter
        end
    end
    return BodyFontSize, TitleFontSize;
Function Main():
    FirstFontSize ⇐ Choosing the diameter with the most number of
     repetition;
    for i=1:Number of CCs do
        if Number of CC_i's pixels < (10 × FirstFontSize) then
            removing CC_i's diameters from diameter's table
        end
    end
    UD = Sorting unique diameters;
    UDNum = Calculating the number of repetitions of unique diameters;
    SecondFontSize ⇐ choosing the diameter with the most number of
     repetition;
    BodyFontSize, TitleFontSize =
     final-font-sizes(UD,UDNum,SecondFontSize);
    Title Image = [ ];
    Body Image = [ ];
    for i=1:Number of CCs do
        if CC_i's diameter < (2 × BodyFontSize) then
            Body Image ⇐ CC_i
        else
            Title Image ⇐ CC_i
        end
    end
```

در بیشتر مواقع، این مرحله در همین بخش به پایان می‌رسد و اندازه قلم ثانویه تصویر به عنوان اندازه قلم نهایی تصویر در خروجی قرار می‌گیرد. اما در بعضی از مواقع این اندازه قلم میزان خطای بالایی خواهد

داشت. وجود تیتر بزرگ در تصویر یکی از این موارد است. همانطور که در شکل ۱۹-۳ نشان داده شده است، پهنای CCهای تیتر بسیار بزرگ‌تر از پهنای CCهای مربوط به بدنه تصویر هستند و در صورت انتخاب یکی از پهناهای CCهای تیتر به عنوان اندازه نهایی قلم تصویر، ما در مراحل بعدی با مشکلات عمده‌ای مواجه خواهیم شد. لذا بهترین کار، جداسازی تیتر از بدنه تصویر است. به همین منظور، نیاز به تعریف اندازه قلم برای تیتر و بدنه به صورت جداگانه است. همانطور که در شبه کد **Error! Reference source not found.** نشان داده شده، در صورتی که پهنایی بزرگ‌تر از دو برابر اندازه قلم ثانویه تصویر وجود داشته باشد، الگوریتم متوجه وجود تیتر بزرگ درون تصویر می‌شود. در صورت وجود تیتر بزرگ درون تصویر، اندازه قلم ثانویه تصویر به عنوان اندازه قلم بدنه تصویر لحاظ می‌شود و در مرحله بعد، در بین مابقی پهناهای درون تصویر که بزرگتر از دو برابر اندازه قلم ثانویه تصویر هستند، پهنایی که بیشترین تکرار (تعداد پیکسل) را به خود اختصاص داده داست، به عنوان اندازه قلم تیتر در نظر گرفته می‌شود. در نهایت CCهای مربوط به هر بخش در یک تصویر جداگانه ریخته می‌شوند. شکل ۱۹-۳ نمونه‌ای از اجرای این مرحله را به نمایش گذاشته است.


الف: تصویر ورودی

ب: تصویر عنوان

ج: تصویر بدنه

شکل ۱۹-۳: نمونه‌ای از تصویر با تیتر بزرگ

## ۳-٤-٤ تشخیص و تقسیم‌بندی خطوط

در این مرحله موقعیت تقریبی خطوط در تصویر استخراج می‌شود. برای این منظور، الگوریتم پیشنهادی CCهایی که به صورت افقی در یک راستا هستند را پیدا کرده و به یکدیگر متصل می‌نماید. برای پیدا کردن CCهای همسایه CCیی مانند i ($CC_i$) نیاز به یک جستجو با یک شعاع خاص هست. اندازه این شعاع در تشخیص درست خطوط بسیار موثر است. به‌طور خاص، زمانی که تصویر بیشتر از یک ستون متنی داشته باشد، با بکارگیری یک شعاع جستجوی نسبتا بزرگ، خطوط دو ستون مجاور به یکدیگر متصل می‌شوند. برای نمونه شکل ۲۰-۳ را در نظر بگیرید. همان‌طور که مشاهده می‌شود، بکارگیری شعاع جستجوی نامناسب باعث افزایش خطا سیستم می‌شود. در صورت کم بودن این شعاع، CCهای درون یک خط نمی‌توانند یکدیگر را پیدا کنند و خط بدرستی تشخیص داده نخواهد شد. همچنین در صورت بزرگ بودن این شعاع، امکان اتصال خطوط ستون‌های مجاور وجود دارد. لذا انتخاب یک شعاع جستجوی دقیق بسیار حائز اهمیت است.


الف: تصویر دو ستونه

ب: استفاده از شعاع جستجوی کوچک

ج: استفاده از شعاع جستجوی بزرگ

د: استفاده از شعاع جستجوی مناسب

شکل ۲۰-۳: نمونه‌ای از چالش پیدا کردن شعاع جستجو

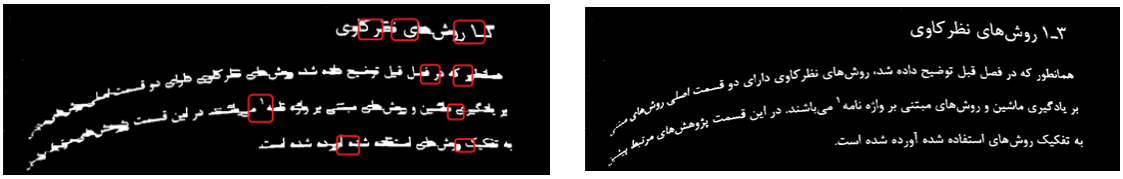
برای حل مشکل پیدا کردن بهترین شعاع جستجو، الگوریتم از اندازه نهایی قلم استخراج شده از بخش قبل کمک می‌گیرد. با توجه به گستردگی متون و انواع مختلف تصویر، پس از سعی و خطاهای فراوان با بکارگیری یک ضریب مناسب از اندازه نهایی قلم، توانستیم بدرستی CCهای درون یک خط را به یکدیگر متصل و از اتصال دو خط در دو ستون مجاور جلوگیری کنیم. همانطور که در الگوریتم ۷ مشاهده می‌شود، بر اساس اندازه نهایی قلم تصویر، این ضریب تنظیم شده است. با بزرگ شدن اندازه قلم نهایی تصویر، این ضریب کوچک‌تر می‌شود تا از اتصال‌های نادرست جلوگیری کند. با بکارگیری این شعاع جستجو در هر CC، می‌توان CCهای همسایه در راستای افق را تشخیص و به یکدیگر متصل ساخت.

الگوریتم ۷: شبکه کد استخراج شعاع جستجو

**if** $final - font - size < 50$ **then**
  $A = \frac{60 - final - font - size}{10}$
**else**
  $A = 1$
**end**
searching-length = A $\times$ final-font-size

### ۳-٤-٤-۱ خطوط دارای انحنا

همان‌طور که در فصل اول نیز گفته شد، یکی از چالش‌های اساسی پیش‌رو این تحقیق خطوط دارای انحنا است. شعاع جستجو در تشخیص خطوط دارای انحنا از اهمیت بیشتری برخوردار است. همان‌طور که در شکل ۲۱-۳ نشان داده شده است، حتی در صورت یک ستونه بودن تصویر نیز امکان اتصال خطوط وجود دارد. در صورت بزرگ بودن این شعاع جستجو، در انتهای خطوط که انحنا وجود دارد، ممکن است خطوط به‌هم متصل شوند. لذا باز هم پیدا کردن شعاع جستجوی مناسب از اهمیت بالایی برخوردار است. همان‌طور که در بخش قبل نیز گفته شد، شعاع جستجویی که در این تحقیق از آن استفاده شده است، پس از آزمون و خطاهای فراوان بر روی انواع مختلف تصویر حاصل شده است. لذا بر روی خطوط دارا انحنا نیز جواب بسیار خوبی می‌دهد.




الف: تصویر دارای انحنا

ب: استفاده از شعاع جستجوی کوچک

ج: استفاده از شعاع جستجوی بزرگ

د: استفاده از شعاع جستجوی مناسب

شکل ۲۱-۳: نمونه‌ای از چالش شعاع جستجو در خطوط دارای انحنا

## ۳-٤-٥ استخراج ناحیه هر خط

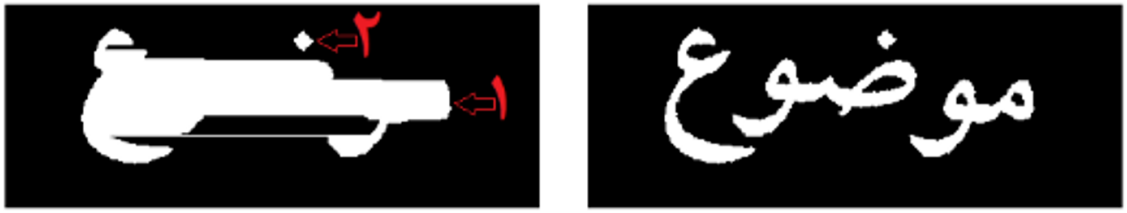
پس از اتصال CCهای درون هر خط به یکدیگر، برای استخراج کامل و صحیح خطوط نیاز است که بعضی از CCهایی که به بدنه اصلی نچسبیده‌اند را نیز شناسایی کنیم. در بعضی از مواقع، اگر CCیی در نزدیکی یک CC نباشد، آن CC به صورت جدا قرار می‌گیرد و به هیچ خطی متصل نمی‌شود. همان‌طور که در شکل ۳-۲۲ نشان داده شده است، پس از اتمام مرحله اتصال CCها، چندین CC هستند که به بدنه اصلی خط نچسبیده‌اند. این CCها معمولا نقاط یا اعراب مربوط به بعضی از کلمات هستند. همان‌طور که گفته شد، دلیل این اتفاق نوع جستجویی است که بر روی CCها اعمال می‌شود. الگوریتم فقط در راستای افق به دنبال پیکسل‌های سفید می‌گردد که دلیل استفاده نکردن از جستجو در راستای عمود، چسبیده شدن خطوط در تصاویری با فاصله عرضی بسیار کم، است. ما برای حل مشکل نچسبیدن بعضی از اعراب و نقاط به بدنه اصلی خط، راه‌حلی را پیشنهاد کرده‌ایم.


الف: تصویر ورودی

ب: تصویر حاصل از اتصال CCها

شکل ۲۲-۳: عدم اتصال بعضی از نقاط به بدنه اصلی خط

نقطه "ض" در کلمه "موضوع" شکل ۲۲-۳، یکی از نقاطی بود که الگوریتم به دلیل پیدا نکردن هیچ همسایه‌ای در شعاع جستجوی تعریف شده، نتوانسته بود آن را به بدنه اصلی خط بچسباند. همانطور که در شکل ۲۳-۳ مشاهده می‌شود، در پایان بخش قبل، این نقطه به عنوان یک CC جدا در نظر گرفته شد.




شکل ۲۳-۳: عدم اتصال نقطه "ض"

برای حل این مشکل و حداقل ساختن خطا، همانطور که در شکل ۲۴-۳ مشاهده می‌شود، الگوریتم حول بدنه هر خط یک کادر محصورکننده می‌سازد و هر CCیی که در این کادر قرار گیرد به عنوان بخشی از CC اصلی در نظر گرفته می‌شود. در نهایت هر یک از این خطوط در خروجی قرار می‌گیرند.

روش پیشنهادی در این مرحله برای ساختن کادر محصورکننده حول هرخط مینیمم x و y و همچنین ماکسیمم x و y را در آن خط بدست می‌آورد و بر مبنای آن کادری را حول آن خط می‌کشد.


شکل ۲۴-۳: نمونه‌ای از ساخت کادر محصورکننده حول هر خط

## ۳-۵ جمع‌بندی

در این فصل به بیان کلیت روش پیشنهادی خود پرداختیم و برخی از مفاهیم پایه‌ی بیان‌شده در فصل دوم را با جزئیات بیشتری توضیح دادیم. کلیت فرآیندهای استخراج مناطق متنی و آشکارسازی خطوط طبق

روش پیشنهادی را مرحله به مرحله بیان نمودیم. آنچه که به عنوان گام اصلی فرآیند استخراج مناطق متنی در روش پیشنهادی مطرح بود، استفاده از سیستم رای‌گیری بود. همچنین آنچه که به عنوان گام اصلی فرآیند آشکارسازی در روش پیشنهادی مطرح بود، مرحله مربوط به تعیین و برچسب‌گذاری نهایی خطوط بود که با استفاده از مفهوم اندازه قلم و عملیات اجرای آن انجام شد.

# فصل ۴ نتایج و آزمایش‌ها

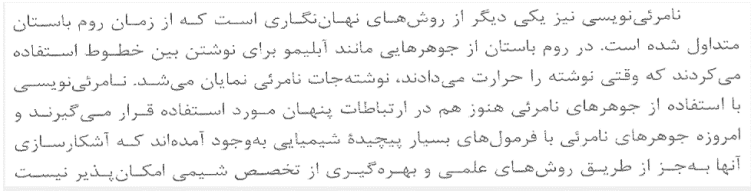
## ٤-١ مقدمه

در فصل قبل روشی برای پیاده‌سازی سیستم‌های DLA ارائه شد که در آن ابتدا مناطق حاوی متن از درون تصویر استخراج می‌شوند و سپس خطوط درون این مناطق بدست می‌آیند. در این فصل ابتدا عملکرد روش پیشنهادی را بررسی می‌کنیم و سپس به انجام آزمایش‌های مختلف بر روی مدل پیشنهادی خواهیم پرداخت. تصویر ورودی سیستم DLA پیشنهادی، توسط اسنکر یا دوربین دریافت می‌شود که ما از یک اسکنر برای تهیه تصاویر کمک گرفتیم. مجموعه تصاویر استفاده شده برای آزمایش مدل پیشنهادی از منابع مختلفی مانند مجله، روزنامه و کتاب‌های گوناگون جمع آوری شده‌اند. همان‌طور که قبلا نیز گفته شد، روش پیشنهادی از دو مرحله تشکیل شده است. به دلیل پیاده‌سازی‌های مدل‌های یادگیری عمیق به زبان پایتون مرحله اول با زبان برنامه‌نویسی پایتون در محیط گوگل کولب[1] نوشته شده است. اما به دلیل وجود توابع آماده در متلب که کار را برای پیاده‌سازی سیستم استخراج خطوط راحت‌تر می‌سازد، مرحله دوم و در اصل بخش اساسی‌تر روش پشنهادی در فضای این نرم‌افزار و نسخه 2018 a شبیه‌سازی شده‌اند.

## ٤-٢ مجموعه دادگان

برای ارزیابی عملکرد روش پیشنهادی، نیاز به یک مجموعه دادگان مناسب است. اما در زبان فارسی ما با کمبود مجموعه دادگان مناسب روبرو هستیم. به همین دلیل، یک مجموعه داده استاندارد ساختیم که برای هر دو مرحله کارایی داشته باشد. این مجموعه داده بالغ بر ۲۰۰۰ تصویر دارد. این تصاویر توسط یک اسکنر مدل HP Scanjet 4890 با تنظیمات پیش‌فرض تهیه شده است. درجه تفکیک در برخی منابع 200 dpi و در برخی دیگر از منابع 300 dpi است که نمونه‌ای از تصاویر پایگاه داده با درجه تفکیک 300 dpi در شکل ۴-۱ مشاهده می‌شود.

[1] Google Colab


نامرئی‌نویسی نیز یکی دیگر از روش‌های نهان‌نگاری است که از زمان روم باستان متداول شده است. در روم باستان از جوهرهایی مانند آبلیمو برای نوشتن بین خطوط استفاده می‌کردند که وقتی نوشته را حرارت می‌دادند، نوشته‌جات نامرئی نمایان می‌شد. نامرئی‌نویسی با استفاده از جوهرهای نامرئی هنوز هم در ارتباطات پنهان مورد استفاده قرار می‌گیرند و امروزه جوهرهای نامرئی با فرمول‌های بسیار پیچیدهٔ شیمیایی به‌وجود آمده‌اند که آشکارسازی آنها به‌جز از طریق روش‌های علمی و بهره‌گیری از تخصص شیمی امکان‌پذیر نیست

شکل ۱-۴: نمونه‌ای از تصاویر مجموعه دادگان بخش متنی

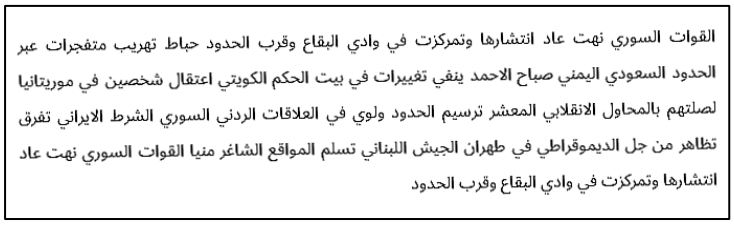
متون این مجموعه دادگان از اندازه‌ها[1]، سبک‌ها[2] و فونت‌ها[3] گوناگونی تشکیل شده‌اند. ما از شش نوع فونت مختلف به نام‌های "Arial"، "Nazanin"، "Ziba"، "IranNastaliq"، "Tahoma" و "XB-Niloofar"، در سه نوع سبک مختلف "Bold"، "Italic" و "Regular"، در پنج اندازه ۸، ۱۰، ۱۴، ۱۸ و ۲۲ در تصاویر خود استفاده کرده‌ایم که در مجموع شامل حدود ۲۴۰۰۰ خط حاوی متن می‌شود. جدول ۱-۴ خلاصه‌ای از اطلاعات بخش متنی مجموعه دادگان را نشان می‌دهد.

جدول ۱-۴: خلاصه‌ای از اطلاعات بخش متنی مجموعه دادگان

| سبک فونت | نوع فونت | تعداد خطوط |
|---|---|---|
| Bold | Arial | ۱۴۲۱ |
| | Nazanin | ۱۴۴۱ |
| | Ziba | ۱۲۳۱ |
| | IranNastaliq | ۷۹۸ |
| | Tahoma | ۱۸۷۳ |
| | XB-Niloofar | ۱۴۶۰ |
| Italic | Arial | ۱۴۰۲ |
| | Nazanin | ۱۴۴۹ |
| | Ziba | ۱۲۲۷ |
| | IranNastaliq | ۷۹۱ |
| | Tahoma | ۱۶۸۵ |
| | XB-Niloofar | ۱۴۳۴ |
| Regular | Arial | ۱۴۰۲ |
| | Nazanin | ۱۴۴۹ |
| | Ziba | ۱۲۲۷ |
| | IranNastaliq | ۷۹۱ |
| | Tahoma | ۱۶۸۵ |
| | XB-Niloofar | ۱۴۳۴ |

[1] Sizes
[2] Styles
[3] Fonts

برای ارزیابی بهتر روش پیشنهادی از یک مجموعه دادگان دیگر به زبان عربی نیز استفاده شده است. این مجموعه دادگان با نام Arabic OCR dataset در مرجع [۸۱] در دسترس است. تصاویر این مجموعه دادگان اندازه‌های مختلفی دارند. تصاویر این مجموعه دادگان تک ستونه و دارای اندکی زاویه هستند. در شکل ۲-۴ نمونه‌ای از تصاویر این مجموعه دادگان نشان داده شده است.


القوات السوري نهت عاد انتشارها وتمركزت في وادي البقاع وقرب الحدود حباط تهريب متفجرات عبر
الحدود السعودي اليمني صباح الاحمد ينفي تغييرات في بيت الحكم الكويتي اعتقال شخصين في موريتانيا
لصلتهم بالمحاول الانقلابي المعشر ترسيم الحدود ولوي في العلاقات الردني السوري الشرط الايراني تفرق
تظاهر من جل الديموقراطي في طهران الجيش اللبناني تسلم المواقع الشاغر منيا القوات السوري نهت عاد
انتشارها وتمركزت في وادي البقاع وقرب الحدود

شکل ۲-۴: نمونه‌ای از تصاویر مجموعه دادگان عربی

## ٤-٣ ارزیابی عملکرد روش پیشنهادی

برای ارزیابی عملکرد مرحله اول الگوریتم پیشنهادی، ما علاوه بر آزمون مرحله اول روش پیشنهادی، ۴ مدل از قبل آموزش دیده Faster R-CNN، YOLOv3، SSD، Layout Parser و OCRopus را نیز بر روی مجموعه دادگان خود آزمودیم. همان‌طور که در جدول ۲-۴ نشان داده شده، روش پیشنهادی بهترین جواب را دارد. دلیل پایین بودن دقت مدل Layout Parser، همپوشانی نسبتا زیاد مناطق غیر هم‌نوع بوده است که باعث کاهش دقت عملکرد الگوریتم شده است. برای محاسبه دقت هر روش از رابطه (۱-۴) بهره گرفته می‌شود که در آن تعداد پیکسل‌های متنی درست تشخیص داده شده بر تعداد کل پیکسل‌های متنی تقسیم می‌شوند.

جدول ۲-۴: دقت تشخیص صحیح مناطق متنی توسط روش پیشنهادی و چهار مدل دیگر

| نام مدل | دقت |
|---|---|
| Layout Parser | ٪۷۸.۳۱ |
| SSD | ٪۸۷.۵۱ |
| YOLOv3 | ٪۹۳.۳۳ |
| Faster R-CNN | ٪۹۵.۶۷ |
| OCRopus | ٪۹۷.۵۴ |
| **روش پیشنهادی تلفیقی در مرحله اول** | **٪۹۸.۰۴** |

$$accuracy = \frac{\sum_{j=1}^{\#of\ pages} \#of\ pixels\ correctly\ detect\ as\ text\ region\ in\ page_j}{\sum_{j=1}^{\#of\ pages} \#of\ text\ region\ pixels\ in\ page_j} \qquad (۴-۱)$$

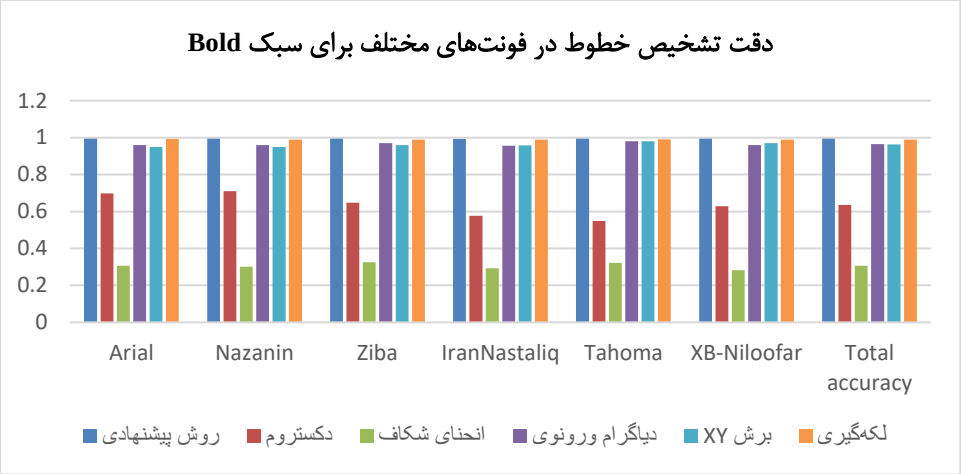
همان‌طور که در جدول ۲-۴ نشان داده شده، تنها دقت نتایج حاصل بر روی مناطق متنی قرار داده شده است. دلیل این امر، اهمیت داشتن این مناطق برای سیستم OCR است. درست است که ما مناطق غیرمتنی را نیز جدا می‌سازیم ولی کار یک سیستم OCR بر روی مناطق متنی صورت می‌گیرد و در مرحله بعدی نیاز است که خطوط درون این مناطق استخراج شوند.

پس از اتمام رای‌گیری میان چهار مدل مبتنی بر یادگیری عمیق، ۵۲۰ تصویر دارای پیکسل‌هایی هستند که سیستم رای‌گیری نمی‌تواند در مورد آنها به قطعیت اظهار نظر کرد. لذا باید درباره آنها مجددا تصمیم‌گیری شود. در جدول ۳-۴ دقت و سرعت مدل OCRopus در مقایسه با روش پنجره ۵*۵ است آورده شده است. همان‌طور که مشاهده می‌شود، با وجود کند بودن مدل OCRopus نسبت به استفاده از تکنیک رای‌گیری روی پنجره لغزان، چندان تفاوت دقتی وجود ندارند. به همین منظور، تنها در تصاویری که تعداد پیکسل‌های مورد مناقشه زیاد باشد (بیشتر از ۵٪ پیکسل‌های تعیین تکلیف شده)، از مدل OCRopus استفاده می‌کنیم. لذا از میان کل ۵۲۰ صفحه پیکسل‌های دارای مناقشه دارند، تنها ۲۳ تصویر یعنی کمتر از ۱٪ تعداد کل تصاویر مجموعه دادگان، از OCRopus استفاده می‌کنند. که بیان‌گر عملکرد خوب روش پیشنهادی است.

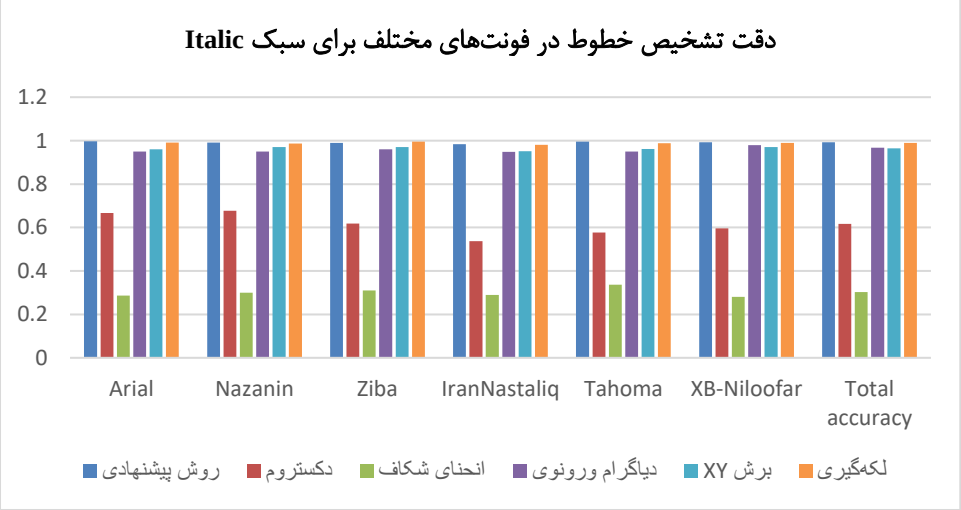
جدول ۳-۴: مقایسه دقت و سرعت مدل OCRopus و پنجره ۵*۵

| نام مدل | دقت | میانگین زمان اجرا برای هر تصویر (s) |
|---|---|---|
| پنجره ۵*۵ | ٪۹۷.۳۷ | ۲.۳۴ |
| OCRopus | ٪۹۷.۵۴ | ۱۸.۰۵۱ |

برای ارزیابی عملکرد مرحله دوم الگوریتم پیشنهادی، ما نیاز به پیاده‌سازی چندین روش مختلف و آزمایش آنها بر روی مجموعه دادگان خود داشتیم. لذا از پنج روش پایه لکه‌گیری[1]، برش (XY)[2]، دیاگرام ورونوی[3]، انحنای شکاف[4] و دکستروم[5] برای مقایسه با روش خود استفاده کردیم. همانطور که مشاهده می‌شود، جدول ۴-۴، جدول ۴-۵ و جدول ۴-۶ دقت الگوریتم‌ها را در هر یک از انواع فونت‌ها بر مبنای نوع سبک و در شکل ۴-۳، شکل ۴-۴ و شکل ۴-۵ خلاصه‌ای از این نتایج نشان داده شده است.

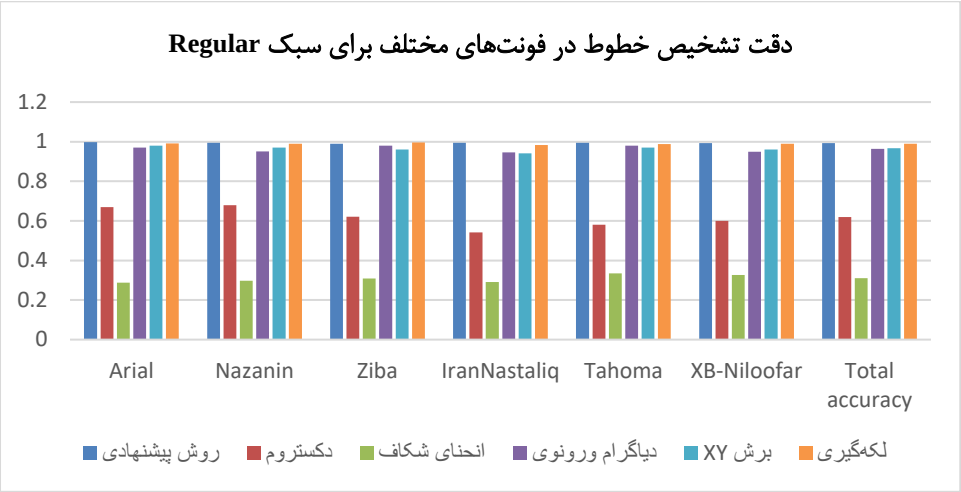



شکل ۴-۳: نمودار دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در فونت‌های مختلف برای سبک Bold




[1] Smearing
[2] XY cut
[3] Voronoi diagram
[4] Seam curving
[5] Docstrum

شکل ۴-۴: نمودار دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در فونت‌های مختلف برای سبک Italic




شکل ۵-۴: نمودار دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در فونت‌های مختلف برای سبک Regular

جدول ۴-۴: دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در فونت‌های مختلف برای سبک Bold

| نوع فونت | لکه‌گیری | برش XY | دیاگرام ورونوی | انحنای شکاف | دکستروم | روش پیشنهادی |
|---|---|---|---|---|---|---|
| Arial | ٪۹۹.۲۳ | ٪۹۵ | ٪۹۵.۹۹ | ٪۳۰.۶۱ | ٪۶۹.۷۴ | ٪۹۹.۵۱ |
| Nazanin | ٪۹۸.۸۹ | ٪۹۵ | ٪۹۵.۹۸ | ٪۳۰.۰۵ | ٪۷۱.۰۶ | ٪۹۹.۴۴ |
| Ziba | ٪۹۸.۹۴ | ٪۹۶.۰۲ | ٪۹۶.۹۹ | ٪۳۲.۵۸ | ٪۶۴.۸۳ | ٪۹۹.۳۵ |
| IranNastaliq | ٪۹۸.۸۷ | ٪۹۵.۸۴ | ٪۹۵.۵۹ | ٪۲۹.۲۶ | ٪۵۷.۷۶ | ٪۹۹.۲۴ |
| Tahoma | ٪۹۹.۰۴ | ٪۹۸.۰۲ | ٪۹۸.۰۲ | ٪۳۲.۱۴ | ٪۵۴.۹۹ | ٪۹۹.۴۱ |
| XB-Niloofar | ٪۹۸.۸۴ | ٪۹۶.۹۹ | ٪۹۶.۰۳ | ٪۲۸.۲۲ | ٪۶۲.۸۱ | ٪۹۹.۴۵ |
| Total accuracy | ٪۹۸.۹۸ | ٪۹۶.۲۸ | ٪۹۶.۵۷ | ٪۳۰.۶ | ٪۶۳.۴۹ | ٪۹۹.۴۲ |

جدول ۵-۴: دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در فونت‌های مختلف برای سبک Italic

| نوع فونت | لکه‌گیری | برش XY | دیاگرام ورونوی | انحنای شکاف | دکستروم | روش پیشنهادی |
|---|---|---|---|---|---|---|
| Arial | ٪۹۹.۰۷ | ٪۹۶.۰۱ | ٪۹۵.۰۱ | ٪۲۸.۶۷ | ٪۶۶.۶۹ | ٪۹۹.۷۱ |
| Nazanin | ٪۹۸.۶۹ | ٪۹۷.۰۳ | ٪۹۵.۰۳ | ٪۲۹.۹۵ | ٪۶۷.۷ | ٪۹۹.۱ |
| Ziba | ٪۹۹.۵۱ | ٪۹۶.۹۸ | ٪۹۶.۰۱ | ٪۳۱.۰۵ | ٪۶۱.۷۸ | ٪۹۸.۹۴ |
| IranNastaliq | ٪۹۸.۱ | ٪۹۵.۰۷ | ٪۹۴.۸۲ | ٪۲۸.۹۵ | ٪۵۳.۷۳ | ٪۹۸.۳۶ |

| | | | | | | |
|---|---|---|---|---|---|---|
| Tahoma | ٪۹۸.۷۵ | ٪۹۶.۱۴ | ٪۹۵.۰۱ | ٪۳۳.۷۱ | ٪۵۷.۷۴ | ٪۹۹.۴۷ |
| XB-Niloofar | ٪۹۸.۸۸ | ٪۹۷ | ٪۹۷.۹۸ | ٪۲۸.۱ | ٪۵۹.۶۲ | ٪۹۹.۳ |
| Total accuracy | ٪۹۸.۸۹ | ٪۹۶.۴۶ | ٪۹۶.۶۸ | ٪۳۰.۲۶ | ٪۶۱.۶۸ | ٪۹۹.۲۲ |

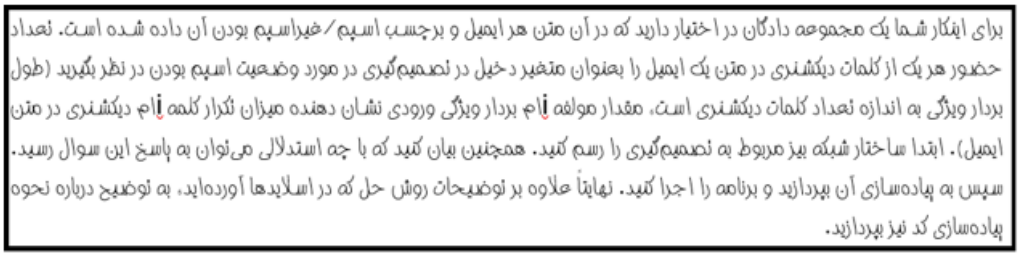
جدول ۶-۴: دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در فونت‌های مختلف برای سبک Regular

| نوع فونت | لکه‌گیری | برش XY | دیاگرام ورونوی | انحنای شکاف | دکستروم | روش پیشنهادی |
|---|---|---|---|---|---|---|
| Arial | ٪۹۹.۱۴ | ٪۹۸ | ٪۹۷ | ٪۲۸.۷۴ | ٪۶۶.۹۸ | ٪۹۹.۷۱ |
| Nazanin | ٪۹۹.۰۳ | ٪۹۷.۰۳ | ٪۹۵.۰۳ | ٪۲۹.۸۱ | ٪۶۷.۹۸ | ٪۹۹.۳۸ |
| Ziba | ٪۹۹.۶۷ | ٪۹۶.۰۱ | ٪۹۷.۹۶ | ٪۳۰.۸۱ | ٪۶۲.۰۲ | ٪۹۸.۹۴ |
| IranNastaliq | ٪۹۸.۲۳ | ٪۹۴.۰۴ | ٪۹۴.۵۶ | ٪۲۹.۰۸ | ٪۵۴.۱۱ | ٪۹۹.۳۶ |
| Tahoma | ٪۹۸.۸۱ | ٪۹۶.۹۷ | ٪۹۷.۹۸ | ٪۳۳.۴۷ | ٪۵۷.۹۸ | ٪۹۹.۴۱ |
| XB-Niloofar | ٪۹۹.۰۲ | ٪۹۶.۰۳ | ٪۹۴.۹۸ | ٪۳۲.۶۴ | ٪۵۹.۹۷ | ٪۹۹.۳ |
| Total accuracy | ٪۹۹.۰۲ | ٪۹۶.۶۴ | ٪۹۶.۳۹ | ٪۳۰.۹۸ | ٪۶۱.۹۷ | ٪۹۹.۲۶ |

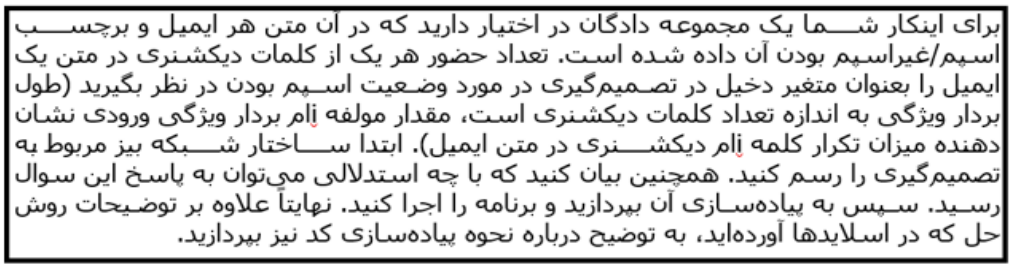
ما برای محاسبه دقت هر یک از این روش‌ها از رابطه (۲-۴) استفاده کردیم که در آن، تعداد خطوطی که بدرستی تشخیص داده شده‌اند را بر تعداد کل خطوط تقسیم می‌کند.

$$accuracy_{i,j} = \frac{\#of\ lines\ corretly\ segmented}{total\ \#of\ lines}, i \in font\ style, j \in font\ types \qquad (۴-۲)$$

همان‌طور که در جدول ۵-۴ و جدول ۶-۴ نشان داده شده است، بجز در یک مورد که دقت روش پیشنهادی در مواجهه با فونت Ziba کمتر از دقت الگوریتم لکه‌گیری است، در مابقی حالات روش پیشنهادی بهترین دقت را گرفته است. دلیل این کاهش دقت در فونت Ziba، کوچک بودن پهنا یا قطر حروف در این فونت است. به عنوان نمونه در شکل ۶-۴ دو نوع فونت Ziba و Tahoma دارای اندازه فونت یکسان ۱۴ هستند و پهنای کلمات در فونت Tahoma بیشتر از پهنای کلمات در فونت Ziba است. این کوچک بودن پهنا باعث کاهش شعاع جستجو و در نهایت پایین آمدن دقت تشخیص صحیح خطوط در فونت Ziba توسط روش پیشنهادی می‌شود. همچنین همان‌طور که در جدول ۴-۴ نشان داده شده است، با بکارگیری سبک Bold، به دلیل افزایش پهنای کلمات، این مشکل تا حدود زیادی مرتفع می‌شود.


برای اینکار شما یک مجموعه دادگان در اختیار دارید که در آن متن هر ایمیل و برچسب اسپم/غیراسپم بودن آن داده شده است. تعداد حضور هر یک از کلمات دیکشنری در متن یک ایمیل را بعنوان متغیر دخیل در تصمیم‌گیری در مورد وضعیت اسپم بودن در نظر بگیرید (طول بردار ویژگی به اندازه تعداد کلمات دیکشنری است، مقدار مولفه iام بردار ویژگی ورودی نشان دهنده میزان تکرار کلمه iام دیکشنری در متن ایمیل). ابتدا ساختار شبکه نیز مربوط به تصمیم‌گیری را رسم کنید. همچنین بیان کنید که با چه استدلالی می‌توان به پاسخ این سوال رسید. سپس به پیاده‌سازی آن بپردازید و برنامه را اجرا کنید. نهایتاً علاوه بر توضیحات روش حل که در اسلایدها آورده‌اید، به توضیح درباره نحوه پیاده‌سازی کد نیز بپردازید.

الف: فونت Ziba با اندازه ۱۴


برای اینکار شما یک مجموعه دادگان در اختیار دارید که در آن متن هر ایمیل و برچسب اسپم/غیراسپم بودن آن داده شده است. تعداد حضور هر یک از کلمات دیکشنری در متن یک ایمیل را بعنوان متغیر دخیل در تصمیم‌گیری در مورد وضعیت اسپم بودن در نظر بگیرید (طول بردار ویژگی به اندازه تعداد کلمات دیکشنری است، مقدار مولفه iام بردار ویژگی ورودی نشان دهنده میزان تکرار کلمه iام دیکشنری در متن ایمیل). ابتدا ساختار شبکه نیز مربوط به تصمیم‌گیری را رسم کنید. همچنین بیان کنید که با چه استدلالی می‌توان به پاسخ این سوال رسید. سپس به پیاده‌سازی آن بپردازید و برنامه را اجرا کنید. نهایتاً علاوه بر توضیحات روش حل که در اسلایدها آورده‌اید، به توضیح درباره نحوه پیاده‌سازی کد نیز بپردازید.

ب: فونت Tahoma با اندازه ۱۴

شکل ۶-۴: تفاوت ساختاری فونت Ziba با دیگر فونت‌ها

در جدول ۷-۴، جدول ۸-۴ و جدول ۱۰-۴ دقت الگوریتم‌ها در هر یک از اندازه‌های داده شده بر مبنای نوع سبک نشان داده است. ما برای محاسبه دقت هر یک از این روش‌ها از رابطه (۳-۴) استفاده کردیم که در آن، تعداد خطوطی که بدرستی تشخیص داده شده‌اند را بر تعداد کل خطوط تقسیم می‌کند. همان‌طور که در جدول ۷-۴، جدول ۸-۴ و جدول ۱۰-۴ مشاهده می‌شود، دقت روش پیشنهادی از تمامی ۵ الگوریتم پایه آورده شده بالاتر است که نشان از ظرفیت بالای این روش است.

$$accuracy_{i,j} = \frac{\#of\ lines\ corretly\ segmented}{total\ \#of\ lines}, i \in font\ style, j \in font\ sizes \qquad (۴-۳)$$

جدول ۷-۴: دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در اندازه فونت‌های مختلف برای سبک Bold

| روش پیشنهادی | دکستروم | انحنای شکاف | دیاگرام ورونوی | برش XY | لکه‌گیری | اندازه فونت |
|---|---|---|---|---|---|---|
| ٪۹۷.۳ | ٪۵۴.۹۹ | ٪۲۱.۳۴ | ٪۹۳.۴۸ | ٪۹۳.۴۸ | ٪۹۵.۸۱ | ۸ |
| ٪۹۹.۰۳ | ٪۵۷.۶۸ | ٪۲۶.۸۷ | ٪۹۵.۲۵ | ٪۹۴.۸۵ | ٪۹۷.۴۳ | ۱۰ |
| ٪۹۹.۷۱ | ٪۶۰.۸۳ | ٪۲۸.۵۶ | ٪۹۶.۸۶ | ٪۹۶.۵۵ | ٪۹۹.۷ | ۱۴ |
| ٪۹۹.۹۲ | ٪۶۵.۷۲ | ٪۳۳.۴۲ | ٪۹۷.۰۵ | ٪۹۶.۶ | ٪۹۹.۹ | ۱۸ |
| ٪۱۰۰ | ٪۷۰.۴ | ٪۳۵.۸۴ | ٪۹۸.۰۸ | ٪۹۷.۸۶ | ٪۹۹.۹۶ | ۲۲ |
| ٪۹۹.۴۲ | ٪۶۳.۴۹ | ٪۳۰.۶ | ٪۹۶.۵۷ | ٪۹۶.۲۶ | ٪۹۸.۹۸ | Total accuracy |

جدول ۸-۴: دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در اندازه فونت‌های مختلف برای سبک Italic

| روش پیشنهادی | دکستروم | انحنای شکاف | دیاگرام ورونوی | برش XY | لکه‌گیری | اندازه فونت |
|---|---|---|---|---|---|---|
| ٪۹۶.۳۷ | ٪۵۱.۲۴ | ٪۱۹.۵۶ | ٪۹۳.۸ | ٪۹۳.۴۲ | ٪۹۵.۶۱ | ۸ |
| ٪۹۸.۶۸ | ٪۵۴.۹ | ٪۲۵.۶۸ | ٪۹۳.۹۹ | ٪۹۵.۴۷ | ٪۹۷.۹۴ | ۱۰ |
| ٪۹۹.۶۱ | ٪۵۹.۰۳ | ٪۲۸.۴۵ | ٪۹۴.۹ | ٪۹۵.۹۴ | ٪۹۹.۲۹ | ۱۴ |
| ٪۹۹.۹۵ | ٪۶۴.۶۳ | ٪۳۳.۴۹ | ٪۹۶.۳۹ | ٪۹۹.۹۷ | ٪۹۹.۷۹ | ۱۸ |
| ٪۹۹.۹۶ | ٪۶۹.۴۸ | ٪۳۶.۱۸ | ٪۹۷.۳۹ | ٪۹۸.۳۲ | ٪۹۹.۸۷ | ۲۲ |
| ٪۹۹.۲۲ | ٪۶۱.۶۸ | ٪۳۰.۲۶ | ٪۹۶.۶۸ | ٪۹۶.۴۶ | ٪۹۸.۸۹ | Total accuracy |

جدول ۹-۴: دقت تشخیص خطوط روش پیشنهادی و سه روش دیگر به همراه میانگین زمان اجرا

| روش‌ها | دقت | میانگین زمان اجرا به ازای هر تصویر (ثانیه) |
|---|---|---|
| Kraken [82] | ٪۹۸.۷۵ | ۲۳.۲۲ |
| OCRopus [83] | ٪۹۹.۱ | ۱۸.۲۴ |
| لکه‌گیری | ٪۹۵.۱ | ۱۴.۴۵ |
| روش پیشنهادی | ٪۹۹.۲۴ | ۸.۳۳ |

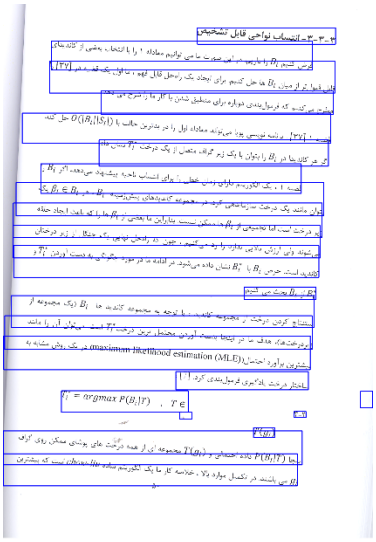
جدول ۱۰-۴: دقت تشخیص خطوط روش پیشنهادی و پنج الگوریتم پایه در اندازه فونت‌های مختلف برای سبک Regular

| روش پیشنهادی | دکستروم | انحنای شکاف | دیاگرام ورونوی | برش XY | لکه‌گیری | اندازه فونت |
|---|---|---|---|---|---|---|
| ٪۹۶.۳۷ | ٪۵۱.۵۳ | ٪۲۱.۰۹ | ٪۹۴.۵۶ | ٪۹۳.۷ | ٪۹۶.۱۸ | ۸ |
| ٪۹۸.۷۷ | ٪۵۵.۱۴ | ٪۲۶.۵ | ٪۹۴.۹ | ٪۹۵.۷۲ | ٪۹۸.۰۲ | ۱۰ |
| ٪۹۹.۶۸ | ٪۵۹.۳۷ | ٪۲۹.۰۳ | ٪۹۵.۷۴ | ٪۹۶.۱۶ | ٪۹۹.۴۲ | ۱۴ |
| ٪۹۹.۹۵ | ٪۶۴.۹۴ | ٪۳۴.۰۱ | ٪۹۶.۸۷ | ٪۹۷.۰۷ | ٪۹۹.۸۴ | ۱۸ |
| ٪۱۰۰ | ٪۶۹.۷۵ | ٪۳۶.۷۵ | ٪۹۸.۱ | ٪۹۸.۵ | ٪۹۹.۹۱ | ۲۲ |
| ٪۹۹.۲۶ | ٪۶۱.۹۷ | ٪۳۰.۹۸ | ٪۹۶.۳۹ | ٪۹۶.۶۴ | ٪۹۹.۰۲ | Total accuracy |

برای ارزیابی بهتر روش پیشنهادی، از دادگانی دیگر با زبان عربی نیز استفاده کرده‌ایم. علاوه بر روش پیشنهادی، ما روش‌های لکه‌گیری، Kraken و OCRopus را نیز بر روی این مجموعه دادگان آزمودیم که نتایج آن در جدول ۹-۴ آورده شده است. همان‌طور که در جدول ۹-۴ نشان داده شده، روش پیشنهادی بهترین دقت را بر روی دادگان عربی داشته است. دلیل بالاتر بودن دقت روش پیشنهادی بر روی این مجموعه دادگان، وجود برخی مشکلات در روش‌های دیگر است که مانع افزایش دقت آنها می‌شود.

یکی از مشکلات اساسی روش OCRopus، بالا بودن زمان اجراست که ممکن است برای بعضی از تصاویر بیشتر از ۳۰ ثانیه نیز به طول بیانجامد. اما مشکل این الگوریتم به همینجا ختم نمی‌شود. یکی از مشکلات اساسی این روش حذف نقاط کوچک درون تصویر است. در زبان‌های فارسی و عربی که تعداد زیادی از حروف دارای نقطه هستند، این مسئله بسیار حائز اهمیت خواهد شد.

همانند روش OCRopus، روش Kraken نیز زمان اجرای نسبتا طولانی دارد. به‌علاوه، روش Kraken در تصاویری با خطوط کج یا دارای انحنا، دقت پایینی دارد. همان‌طور که در شکل ۷-۴ نشان داده شده، در صورت کج بودن تصویر، روش Kraken قادر به استخراج صحیح خطوط متنی نیست.



شکل ۷-۴: خروجی روش Kraken بر روی تصویر کج

## ٤-٤ نتیجه‌گیری

در این فصل، خروجی‌های حاصل از اجرای روش پیشنهادی را در مراحل مختلف نشان دادیم و برخی از نتایج حاصل از بکارگیری الگوریتم‌های مورد استفاده در روش پیشنهادی را بیان نمودیم. همچنین به مقایسه دقت حاصل از روش پیشنهادی با برخی از کارهای مرتبط پرداخته شد و مشاهده گردید که روش پیشنهادی از دقت بسیار خوبی هم در استخراج مناطق حاوی متن و هم در آشکارسازی خطوط در متون تایپی برخورداراست. در مرحله اول، روش پیشنهادی با چندین مدل یادگیری عمیق که همگی در حوزه شناسایی و استخراج مناطق متنی دقت‌های خوبی داشته است مقایسه گردید. در مرحله دوم نیز سعی گردید علاوه بر روش‌های سنتی، روش پیشنهادی با روش‌های جدیدتر که مبتنی بر یادگیری عمیق هستند مقایسه شود تا نتیجه نهایی قابل استنادتر شود.

# فصل ۵ جمع‌بندی و سوی کارهای آتی

## ۵-۱ جمع‌بندی

در مرحله اول این پایان‌نامه، به عملیات استخراج مناطق حاوی متن از درون تصویر اصلی با بکارگیری یک سیستم رای‌گیری بین چندین روش مختلف، پرداخته شد. در مرحله دوم این پایان‌نامه نیز به عملیات تعیین و آشکارسازی خطوط تایپی فارسی با استفاده از الگوریتم‌ها و مفاهیم موجود در پردازش تصویر پرداخته شد. با توجه به وجود چالش‌های موجود سعی در ارائه‌ی روشی بود که بتوانیم بهترین مناطق متنی را استخراج و فرایند آشکارسازی خط را با دقت بالایی انجام دهیم. یکی از چالش‌های موجود، وجود خطوط مورب در تصاویر متنی و همچنین تغییرات زاویه انحراف در امتداد خطوط متنی بود. این چالش با توجه به اینکه باعث درهم‌فرورفتگی و فاصله‌ی نابرابر در خطوط متون نسبت به هم می‌شود، گاهاً باعث انجام فرآیند آشکارسازی به شکل کم‌دقت و حتی اشتباه می‌شود. بدین منظور باید از روشی استفاده می‌شد، که بتوان با استفاده از آن، خروجی فرایند را بهبود بخشید. ما در روش خود از مفهوم اندازه قلم برای جداسازی دقیق خطوط استفاده نمودیم که به نوبه‌ی خود روشی جدید در زبان فارسی محسوب می‌شود. در ادامه به بیان چالش‌ها و پیشنهادهایی برای ادامه این موضوع خواهیم پرداخت.

## ۵-۲ چالش‌ها، مزایا و معایب

با توجه به آنچه که به‌عنوان روش پیشنهادی ارائه گردید، دو مشکل اصلی در روش ارائه‌شده مطرح می‌گردد. اولین مشکلی که می‌توان از لحاظ کارآیی به الگوریتم وارد دانست، مشکل در شناسایی دقیق خطوط برای متونی است که در آن‌ها میزان انحراف خطوط زیاد باشد. نتیجه‌ی اصلی حاصل از وجود احتمالی انحراف خطوط، شناسایی برخی از اجزای مربوط به کاراکترها در کلمات موجود در یک خط (همچون تشدید مربوط به کاراکتر مشخص در یک کلمه)، در خطی غیر از خط اصلی کلمه خواهد بود. ما سعی کردیم تا حد امکان این مشکل را برطرف نماییم که تا حدود زیادی نیز موفق شدیم. بنابراین با توجه به آنچه بیان گردید، خصوصیات روش پیشنهادی را می‌توان به صورت زیر دانست:

- **معایب**
  - ایجاد مشکل اندک در شناسایی دقیق خطوط به دلیل انحراف خطوط در متون.
- **مزایا**
  - دارای دقت بالا.
  - دارای سرعت اجرای بالا.
  - تشخیص خطوط از درون متون چند ستونه.
  - جداسازی عنوان از بدنه اصلی متن.

## ۳-۵ پیشنهادات برای ادامه فعالیت

با توجه به آنچه به‌عنوان چالش‌های روش پیشنهادی مطرح گردیده‌شد، می‌توان موارد زیر را به‌عنوان پیشنهاداتی برای ادامه روند پژوهشی انجام‌شده مطرح دانست.

- ارائه رویکردهایی برای بکارگیری روش پیشنهادی برای متون با زبان‌های دیگر.
- ارائه یک روش کاملا جدید و بدون استفاده از مدل‌های از قبل آموزش دیده شده برای جداسازی مناطق متنی و غیر متنی.
- ارائه یک روش مبتنی بر یادگیری عمیق برای استخراج خطوط.

# فهرست واژگان

| عنوان فارسی | عنوان انگلیسی |
|---|---|
| ابزار وکا | Weka tool |
| اجزای متصل | Connected Component |
| اشکال | Shape |
| الگوریتم‌های غیر تکراری | Non-Iterative Algorithms |
| الگوریتم‌های مبتنی بر الگوهای دودویی محلی | Local Binary Pattern |
| الگوریتم‌های مبتنی بر تکرار | Iterative Thinning Algorithms |
| الگوریتم‌هایی استفاده کننده رویکرد موازی | Fully Parallel Approaches |
| انباشتگر | Accumulator |
| انتقال | Transition |
| اندازه | Size |
| انعکاس | Reflection |
| آنالیز قالب‌بندی سند | Document Layout Analysis |
| بالا به پایین | Top-Down |
| برچسب | Label |
| بلادرنگ | Real-time |
| پایین به بالا | Bottom-Up |
| تبدیل هاف | Hough Transform |
| تصاویر صحنه طبیعی | Natural scene images |
| تقسیم‌بندی | Segmentation |
| تقسیم‌بندی مبتنی بر پایه آستانه‌گذاری | Region based segmentation |
| تقسیم‌بندی مبتنی بر خوشه‌بندی | Thresholding based segmentation |
| تقسیم‌بندی مبتنی بر لبه‌های | Edge based segmentation |
| تقسیم‌بندی مبتنی بر ناحیه | Edge based segmentation |
| جستجوی انتخابی | Selective Search |
| خط پایه | baseline |
| ساختارها | Structure |
| ساختارهای هندسی | Geometrical structure |
| سایش | Erosion |

| | |
|---|---|
| Style | سبک |
| Page Reading | صفحه‌خوان |
| gap | فضای خالی |
| Font | فونت |
| Bounding Box | کادر محصورکننده |
| Character | کاراکتر |
| Dilation | گسترش |
| Clustering based segmentation | لبه‌یاب پریویت |
| Prewitt edge detector | لبه‌یاب سوبل |
| Sobel edge detector | لبه‌یاب کنی |
| Smearing | لکه‌گیری |
| Pre-trained model | مدل از قبل آموزش دیده |
| Automatic Sorting | مرتب‌سازی خودکار |
| Feature map | نقشه ویژگی |
| Project Profile | نمایه تصویر |
| Voronoi diagram | نمودار ورونوی |
| Optimal Character Recognition | نویسه‌خوان نوری |
| Transfer Learning | یادگیری انتقال |
| Deep learning | یادگیری عمیق |

# مراجع


1. Ashiquzzaman, A., et al., *An efficient recognition method for handwritten arabic numerals using cnn with data augmentation and dropout*, in *Data Management, Analytics and Innovation*. 2019, Springer. p. 299-309.
2. Soomro, W.J., et al., *Optical Character Recognition System for Sindhi Text: A Survey*. 2018. **2**(2): p. 81-87.
3. Alkhateeb, F., et al., *Arabic optical character recognition software: A review*. 2017. **27**(4): p. 763-776.
4. Chaudhuri, A., et al., *Optical character recognition systems*, in *Optical Character Recognition Systems for Different Languages with Soft Computing*. 2017, Springer. p. 9-41.
5. Hesham, A.M., et al., *Arabic document layout analysis*. 2017. **20**(4): p. 1275-1287.
6. Amer, I.M., S. Hamdy, and M.G. Mostafa. *Deep Arabic document layout analysis*. in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*. 2017. IEEE.
7. Singh, S., T. Patnaik, and S. Choudhary, *Document Layout Analysis for Hindi Newspapers*.

۸. مظاهری, م. و ب. برکتین, بازشناسی کاراکترهای شماره پلاک خودروی ایرانی مبتنی بر شبکه عصبی پیچشی, در کنفرانس بین المللی مهندسی و علوم کامپیوتر. ۱۳۹۵.

۹. صمدی بهرامی, ه. و ع. جبرئیلی, تشخیص و دنبال کردن پلاک خودرو در یک تصویر ویدئویی بااستفاده از ویژگی رنگی *HSV* و*ocr* آن با شبکه عصبی هاپفیلد؛به صورت بلادرنگ, در سومین کنفرانس بین المللی پژوهشهای کاربردی در مهندسی کامپیوتر و فن آوری اطلاعات. ۱۳۹۴.

10. Roy, A. and D.P. Ghoshal. *Number Plate Recognition for use in different countries using an improved segmentation*. in *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*. 2011. IEEE.
11. Wen, Y., et al., *An algorithm for license plate recognition applied to intelligent transportation system.* 2011. **12**(3): p. 830-845.

۱۲. صابری, ر. و س. طوسی زاده, شناسایی موقعیت و استخراج اعداد کنتور گاز از تصاویر واقعی با استفاده از الگوریتم *MSER*، در دومین همایش چشم انداز تکنولوژی کامپیوتر و شبکه در ۲۰۳۰. ۱۳۹۵.

۱۳. صیادی, ا., کاربرد هوش مصنوعی و سیستم های *OCR* در خواندن متون برای نابینایان, در کنفرانس بین المللی پژوهش های کاربردی در فناوری اطلاعات، کامپیوتر ومخابرات. ۱۳۹۴.

14. Srivastava, S., et al., *Optical character recognition on bank cheques using 2D convolution neural network*, in *Applications of Artificial Intelligence Techniques in Engineering*. 2019, Springer. p. 589-596.

۱۵. برومندنیا, ع., ه. یوسفی رامندی, و ک. خزایی, شناسایی الگوی کدپستی از روی تصویر پاکت نامه و تشخیص ارقام کدپستی از روس نامه های دستنویس فارسی, در همایش ملی کاربرد سیستم های هوشمند ( محاسبات نرم ) در علوم و صنایع. ۱۳۹۲.

16. Guo, Y., et al. *Text line detection based on cost optimized local text line direction estimation*. in *Color Imaging XX: Displaying, Processing, Hardcopy, and Applications*. 2015. International Society for Optics and Photonics.
17. Bukhari ,S.S., et al., *Coupled snakelets for curled text-line segmentation from warped document images.* 2013. **16**(1): p. 33-53.
18. Bukhari, S.S., F. Shafait, and T.M. Breuel. *Ridges based curled textline region detection from grayscale camera-captured document images*. in *International Conference on Computer Analysis of Images and Patterns*. 2009. Springer.
19. Ayesh, M., et al., *A Robust Line Segmentation Algorithm for Arabic Printed Text with Diacritics.* 2017. **2017**(13): p. 42-47.
20. Gatos, B., I. Pratikakis, and K. Ntirogiannis. *Segmentation based recovery of arbitrarily warped document images*. in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. 2007. IEEE.
21. Alaei, A., et al., *Piece-wise painting technique for line segmentation of unconstrained handwritten text: a specific study with Persian text documents.* 2011. **14**(4): p. 381-394.
22. Lauren, B. and L.J.U.P.A. Lee, *Perceptual information processing system. Paravue Inc.* 2003. **10** (543-618).
23. Jebari, I. and D.J.P.E. Filliat, *Color and depth-based superpixels for background and object segmentation.* 2012. **41**: p. 1307-1315.
24. Kaur, M. and E.N.J.I. Singh, *Image segmentation techniques: An overview*. 2002. **2**(y2).
25. Song, Y. and H. Yan. *Image Segmentation Techniques Overview*. in *2017 Asia Modelling Symposium (AMS)*. 2017. IEEE.
26. Kumar, M.J., et al., *Review on image segmentation techniques.* 2014: p. 2278-0882.
27. Liew, A.-C., H. Yan, and N.-F.J.I.T.o.F.S. Law, *Image segmentation based on adaptive cluster prototype estimation.* 20. 444-453.
28. Zaharescu, M., I.C.J.J.o.I.S. Petrescu, and O. Management, *Edge detection in document analysis.* 2013. **7**(1): p. 156-165.
29. Threshold, H.N.N.J.E.J.o.B., *A Survey on relevant Edge Detection Technique For medical image processing* .2016 317-327.
30. Pirzada, S.J.H. and A. Siddiqui. *Analysis of edge detection algorithms for feature extraction in satellite images*. in *2013 IEEE International Conference on Space Science and Communication (IconSpace)*. 2013. IEEE.
31. Phueakjeen, W., et al. *A study of the edge detection for road lane*. in *The 8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand-Conference 2011*. 2011. IEEE.

32. Gao, W., et al. *Based on soft-threshold wavelet de-noising combining with Prewitt operator edge detection algorithm*. in *2010 2nd International Conference on Education Technology and Computer*. 2010. IEEE.
33. Gao, W., et al. *An improved Sobel edge detection*. in *2010 3rd International conference on computer science and information technology*. 2010. IEEE.
34. Rong, W., et al. *An improved CANNY edge detection algorithm*. in *2014 IEEE International Conference on Mechatronics and Automation*. 2014. IEEE.
۳۵. حسن‌پور و اسدی، مفاهیم جامع پردازش تصویر دیجیتال به همراه پیاده سازی الگوریتم ها با *Matlab* انتشارات دانشگاه صنعتی شاهرود, ۱۳۹۴.
36. Dhanachandra, N., K. Manglem, and Y.J.J.P.C.S. Chanu, *Image segmentation using K-means clustering algorithm and subtractive clustering algorithm.* 2015. **54**: p. 764-771.
37. Bloomberg, D.S. *Multiresolution morphological approach to document image analysis*. in *Proc. of the International Conference on Document Analysis and Recognition, Saint-Malo, France*. 1991.
38. Bukhari, S.S., F. Shafait, and T.M. Breuel .*Improved document image segmentation algorithm using multiresolution morphology*. in *Document recognition and retrieval XVIII*. 2011. International Society for Optics and Photonics.
39. Abhishek, L.K.J.I.J.o.L.T.i.E. and Technology, *Thinning approach in digital image processing.* 2017.
40. Ghosh, M., et al., *Coalition game based feature selection for text non-text separation in handwritten documents using LBP based features.* 2020: p. 1-21.
41. Ghosh, S., et al., *Text/non-text separation from handwritten document images using LBP based features: An empirical study.* 2018. **4**(4): p. 57.
42. Harwood, D., et al., *Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions.* 1995. **16**(1): p. 1-10.
43. Jin, H., et al. *Face detection using improved LBP under Bayesian framework*. in *Third International Conference on Image and Graphics (ICIG'04)*. 2004. IEEE.
44. Ojala, T., et al., *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.* 2002. **24**(7): p. 971-987.
45. Pan, S.J., Q.J.I.T.o.k. Yang, and d. engineering, *A survey on transfer learning.* 2009. **22**(10): p. 1345-1359.
46. Redmon, J., et al. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
47. Redmon, J. and A.J.a.p.a. Farhadi, *Yolov3: An incremental improvement.* 2018.
48. Ren, S., et al., *Faster R-CNN: towards real-time object detection with region proposal networks.* 2016. **39**(6): p .1137-1149.
49. Stott, L. *Smart Acc.: Web application based on Faster-RCNN architecture making it easier to find accommodations in London.* 2019; Available from:

https://techcommunity.microsoft.com/t5/educator-developer-blog/smart-acc-web-application-based-on-faster-rcnn-architecture/ba-p/381235.

50. Liu, W., et al. *Ssd: Single shot multibox detector*. in *European conference on computer vision*. 2016. Springer.
51. Dell, Z.S.a.R.Z.a.M. *LayoutParser*. 2020; Available from: https://github.com/Layout-Parser/layout-parser.
52. Nagy, G., S. Seth, and M.J.C. Viswanathan, *A prototype document image analysis system for technical journals.* 1992. **25**(7): p. 10-22.
53. Soni, R., et al., *Optimal feature and classifier selection for text region classification in natural scene images using Weka tool.* 2019. **78**(22): p. 31757-31791.
54. Hall, M., et al., *The WEKA data mining software: an update.* 2009. **11**(1): p. 10-18.
55. Karatzas, D., et al. *ICDAR 2013 robust reading competition*. in *2013 12th International Conference on Document Analysis and Recognition*. 2013. IEEE.
56. Mahmood, A. and A. Srivastava. *A novel segmentation technique for Urdu type-written text*. in *2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS)*. 2018. IEEE.
57. Koo, H.I.J.I.T.o.I.P., *Text-line detection in camera-captured document images using the state estimation of connected components.* 2016. **25**(11): p. 5358-5368.
58. Matas, J., et al., *Robust wide-baseline stereo from maximally stable extremal regions.* 2004. **22**(10): p. 761-7.٦٧
59. Kise, K., et al., *Segmentation of page images using the area Voronoi diagram.* 1998. **70**(3): p. 370-382.
60. Ahmad, I., et al., *Line and ligature segmentation of urdu nastaleeq text.* 2017. **5**: p. 10924-10940.
61. Cheung, A., M. Bennamoun, and N.W.J.P.r. Bergmann, *An Arabic optical character recognition system using recognition-based segmentation.* 2001. **34**(2): p. 215-233.
62. Zeki, A.M., M.S. Zakaria, and C.-Y. Liong, *Segmentation of Arabic characters: A comprehensive survey*, in *Technology Diffusion and Adoption: Global Complexity, Global Innovation*. 2013, IGI Global. p. 251-288.
63. Soujanya, P., et al., *Comparative study of text line segmentation algorithms on low quality documents.* 2010.
64. Banumathi, K. and A.J. Chandra. *Line and word segmentation of Kannada handwritten text documents using projection profile technique*. in *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*. 2016. IEEE.
65. Garg, R. and N.K. Garg, *A New Approach for line segmentation in Punjabi Language using Strip based Projection profile method.* 2014.
66. Malakar, S., et al. *Text line extraction from handwritten document pages using spiral run length smearing algorithm*. in *2012 International Conference on Communications, Devices and Intelligent Systems (CODIS)*. 2012. IEEE.
67. Louloudis, G., et al. *A block-based Hough transform mapping for text line detection in handwritten documents*. 2006.

68. Lyu, B., et al. *The Early Japanese Books Text Line Segmentation base on Image Processing and Deep Learning*. in *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*. 2019. IEEE.
69. Grüning, T., et al., *A two-stage method for text line detection in historical documents.* 2019. **22**(3): p. 285-302.
70. El Bahi, H., A.J.M.t. Zatni, and applications, *Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network.* 2019. **78**(18): p. 26453-26481.
71. Shen, Z., R. Zhang, and M. Dell, *layoutparser*. 2020.
72. Imaging, T., *Monk Object Detection - A low code wrapper over state-of-the-art deep learning algorithms.* 2019.
73. Wu, Y., et al., *Detectron2*. 2019.
74. Mustafa, W.A. and M.M.M.A. Kader. *Binarization of document images: A comprehensive review*. in *Journal of Physics: Conference Series*. 2018. IOP Publishing.
75. Sauvola, J. and M.J.P.r. Pietikäinen, *Adaptive document image binarization.* 2000. **33**(2): p. 225-236.
76. Gomez, G. *Local Smoothness in terms of Variance: the Adaptive Gaussian Filter*. in *BMVC*. 2000 .Citeseer.
77. Bhateja, V., et al. *A non-iterative adaptive median filter for image denoising*. in *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*. 2014. IEEE.
78. Thanh, D.N.H. and S.J.I.A. Engínoğlu, *An iterative mean filter for image denoising.* 2019. **7**: p. 167847-167859.
79. Jin, F., et al. *Adaptive Wiener filtering of noisy images and image sequences*. in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*. 2003. IEEE.
80. Guo, D., et al ,.*Salt and pepper noise removal with noise detection and a patch-based sparse representation.* 2014. **2014**.
81. H. Youssef, Arabic OCR dataset, "https://drive.google.com/drive/folders/1--wsm4NIZB8Reu70jg-wBO56Pq89N6fs" 2021.
82. last release of Kraken https://github.com/mittagessen/kraken 2021.
83. last release of OCRopus https://github.com/ocropus/ocropy 2017.

# Abstract

**Document layout analysis is one of the critical steps in converting a document image to its text. Separating textual and non-textual areas within an image and extracting lines containing text from textual regions is the most effective preprocessing possible in optical character recognition systems. Failure to correctly identify areas containing text and, consequently, incorrect recognition of line coordinates will disrupt all subsequent parts of an optical character recognition system. Problems such as curvature of lines, skewness of the image, presence of diacritics and many dots in Persian and Arabic, the proximity of lines within the image, and pictures with more than one column prevented previous works from achieving high accuracy in document formatting analysis systems. In this research, a new method for analyzing Persian document layout is presented. The proposed method uses several different ways and a voting system among them, extracts the textual areas of the image, and then uses the font size estimate to more accurately identify the lines, which has not been used in previous works. The proposed method is evaluated on a database that contains more than 2000 images. In two sections of detecting areas containing text and extracting lines, the accuracy has reached 98.04% and 99.42%.**

**Keywords: Document Layout Analysis, Line Segmentation, Text Recognition, Image Segmentation, Font Size, Voting**

**Shahrood University of Technology**

Faculty of Computer Engineering and Information Technology

M.Sc. Thesis in Artificial Intelligence Engineering

# Persian Document Layout Analysis

By: Amirreza Fateh

Supervisor:
Dr. Mohsen Rezvani

Advisor:
Dr. Alireza Tajary

August 2021