

## Online Handwritten Persian Isolated Letter Recognition by Using Discrete Markov Models and Language-Based Features

Mohammad Ashurpour<sup>1</sup> and Majid Ziaratban<sup>2\*</sup>

1- Engineering faculty, Golestan university, Gorgan, Iran.

2\*- Engineering faculty, Golestan university, Gorgan, Iran.

<sup>1</sup> ashurpour2@gmail.com, <sup>2\*</sup> m.ziaratban@gu.ac.ir

Corresponding author address: Majid Ziaratban, Engineering faculty of Golestan University, Gorgan, Iran, Post Code : 49188 - 88369

**Abstract-** In this paper, an online Persian isolated letter recognition by using discrete Markov models and language-based features is presented. In the proposed method, the preprocessing includes the extraction of dots, the removal of ending-bends, the normalization of the main body dimension, and the main body smoothing. The direction of the pen for the main body, and also the location and the number of secondary movements and dots of letters are used as the observable features in order to train and test models. No alphabet grouping has been used in the proposed method. In order to produce appropriate discrete observation sequences for the hidden Markov models, the proposed codebook is based on the characteristics of the Persian language. To improve the recognition accuracy, the proposed post-processing checks the recognized character in terms of the number and location of the secondary movements and dots. The final recognition rate is obtained more than 98% for the test dataset and outperforms other methods applied to this dataset.

**Keywords-** Online Recognition, Handwritten Persian Isolated Letter, Discrete Markov Models, Code book, Pre-processing, Post-processing.

## بازشناسی برخط حروف مجزای دست نویس فارسی با استفاده از مدل مخفی مارکوف گسسته و ویژگی‌های مبتنی بر زبان فارسی

محمد عاشورپور<sup>۱</sup>، مجید زیارت‌بان<sup>۲\*</sup>

۱- دانشکده فنی و مهندسی گرگان، دانشگاه گلستان، گرگان، ایران.

۲\* - دانشکده فنی و مهندسی گرگان، دانشگاه گلستان، گرگان، ایران.

<sup>1</sup> ashurpour2@gmail.com, <sup>2\*</sup> m.ziaratban@gu.ac.ir

\* نشانی نویسنده مسئول: مجید زیارت‌بان، گرگان، دانشگاه گلستان، دانشکده فنی و مهندسی گرگان، کد پستی: ۴۹۱۸۸-۸۸۳۶۹

چکیده- در این مقاله، باز شناسی برخط حروف مجزای فارسی با استفاده از مدل مخفی مارکوف گسسته و ویژگی‌های مبتنی بر زبان فارسی از پایه می‌شود. در روش پیشنهادی، از استخراج نقطه‌های حرف، حذف قلاب‌ها، یکسان سازی ابعاد بدنه اصلی و هموار سازی آن بعنوان بخشی از فرایند پیش پردازش استفاده می‌شود. جهت حرکت قلم برای بدنه اصلی و محل و تعداد ریز حرکات و نقطه‌های حروف به عنوان ویژگی‌های قابل مشاهده به منظور استفاده در فرایند آموزش و آزمایش مدل بکار می‌رود. در روش پیشنهادی، از هیچ گونه گروه بندی حروف، استفاده نشده است. به منظور تولید دنباله مشاهدات گسسته برای مدل مخفی مارکوف، از کتاب کد پیشنهادی که مبتنی بر خصوصیات نوشتاری زبان فارسی است استفاده می‌شود. برای افزایش دقت باز شناسی، فرایند پس پردازش پیشنهادی حرف باز شناسی شده را به لحاظ تعداد و محل ریز حرکات و نقطه‌ها بررسی کرده و خروجی نهایی را برمی‌گرداند. نرخ باز شناسی با استفاده از روش پیشنهادی بیش از ۹۸ درصد برای مجموعه داده‌های آزمایشی است که نسبت به سایر روش‌های اعمال شده بر روی این مجموعه داده، دقت بهتری را ارائه می‌دهد.

واژه‌های کلیدی: باز شناسی برخط، حروف مجزای فارسی، مدل مخفی مارکوف، کتاب کد، پیش پردازش، پس پردازش.

### ۱- مقدمه

که در باز شناسی بر خط، اطلاعات نوشتار از قبیل مختصات پیکسل‌ها، ترتیب ثبت آنها، تعداد اجزای حروف و در برخی از موارد، میزان فشار وارده بر صفحه لمسی، همزمان با نوشتن گردآوری می‌شود. تلاش‌های متعددی در زمینه باز شناسی برخط نوشتار در زبان‌های مختلف انجام گرفته است. در زبان فارسی نیز در سالهای اخیر تحقیقاتی صورت گرفته است. از آنجا که بررسی میزان کارایی هر یک از روش‌ها نیاز به پایگاه داده واحد دارد، این پایگاه داده توسط دانشگاه تربیت مدرس به نام Online-TMU [۱-۲] تهیه شده است که در سالهای اخیر بعنوان پایگاه داده مرجع در بسیاری از کارها بوده است. در ادامه به برخی از کارهای انجام گرفته بر روی

امروزه بدلیل استفاده از وسایل ارتباطی که تبادل اطلاعات در آنها بین کاربر و سیستم مبتنی بر صفحات لمسی است، در بسیاری از سیستم‌ها، اطلاعات مورد نیاز از کاربر به صورت برخط<sup>۱</sup> دریافت می‌شود. از طرفی اهمیت دقت داده‌های دریافتی، بویژه داده‌های نوشتاری در اینگونه سیستم‌ها، لزوم استفاده از روش باز شناسی برخط حروف مجزا را نمایان می‌کند.

باز شناسی نوشتار با توجه به نحوه دریافت اطلاعات ورودی به دو روش برون خط<sup>۲</sup> و برخط تقسیم می‌شود. در باز شناسی برون خط، اطلاعات دست نوشته به صورت تصویر موجود می‌باشند. در حالی

این پایگاه داده اشاره می‌شود.

در [۳] از رویکرد تطابق مدل و از یک روش متامورفیزم شکل دو بعدی استفاده شده است. حروف فارسی به چهار گروه تقسیم شده‌اند و بدنه اصلی پس از پس‌پردازش به وسیله متامورفیزم با الگوهای پایه مقایسه شده و بعد از شناسایی اجزای کوچک نتیجه نهایی تعیین شده است. در این روش ضمن مقایسه روش متامورفیزم با  $DTW^3$ ، ترکیب این دو روش هم بکار گرفته شده است. نرخ بازشناسی نهایی با آموزش ۷۰ نفر اول از ۱۲۴ نفر برای مجموعه آزمایش و با روش متامورفیزم ۹۰/۸ و ترکیب دو روش ۹۵/۱ عنوان شده است.

در [۴] یک روش مبتنی بر گروه بندی با استفاده از مدل مخفی مارکوف پیوسته ارائه شده است. ابتدا حروف بر اساس تشابه بدنه به ۱۸ گروه تقسیم شده است. برای هر گروه، آموزش مدل مخفی مارکوف صورت گرفته است. در مرحله بازشناسی، ابتدا گروهی که حرف به آن تعلق دارد بازشناسی شده است. سپس با شناسایی ویژگی‌های خاص آن شامل نقطه و علائم ثانویه، تشخیص نهایی حرف انجام گرفته است. بیشترین نرخ بازشناسی نهایی گزارش شده با آموزش ۷۰ نویسنده اول، برای گروه‌ها ۹۰/۸۲ و برای حروف ۸۸/۸۷ گزارش شده است.

در [۵] از دانش مربوط به بدنه اصلی و ریزحرکات به طور همزمان و به منظور تعیین کلاس خروجی استفاده شده است و حروف مجزای فارسی براساس تشابه بدنه اصلی در ۱۸ گروه و براساس تشابه ریزحرکات در ۱۱ گروه، گروه‌بندی شده‌اند. برای تشخیص نمونه‌های ناشناخته ورودی، بدنه اصلی و ریزحرکات شناسایی شده‌اند. اگر گروه شناسایی شده از لحاظ بدنه اصلی و ریزحرکات همخوانی داشته باشد، نمونه ورودی بازشناسی شده است در غیر این صورت ناهمخوانی پیش آمده با استفاده از الگوریتم تصحیح خطا، تا حد امکان تصحیح شده است. طبقه بند بکار رفته، ماشین بردار پشتیبان (SVM) با رویکرد یک در مقابل یک بوده است. میانگین نرخ بازشناسی برای ده بار اجرای تصادفی و نسبت داده‌های آموزشی و آزمایشی به ترتیب ۰/۷ و ۰/۳ به کل داده‌ها، ۹۷/۹۷ و ۹۷/۴۸ و ۹۷/۲۹ به ترتیب برای بدنه اصلی و ریزحرکات و بازشناسی نهایی گزارش شده است.

در [۶] حروف براساس شکل و ساختار نوشتاری بدنه اصلی به ۹ گروه تقسیم شده‌اند. پس از استخراج ویژگی‌ها، گروه بندی با استفاده از درخت تصمیم انجام شده است. بازشناسی نهایی حروف با توجه به ساختار اجزای کوچک آنها در هر گروه انجام گرفته است. از ۱۱۷ نمونه برخط از هر حرف، ۶۰ نمونه بعنوان داده آموزشی و

۵۷ نمونه بعنوان داده آزمایشی مورد استفاده قرار گرفته است. نتایج بازشناسی گروه بندی و بازشناسی نهایی حروف به ترتیب با دقت ۹۴/۲ و ۹۲/۳ گزارش شده است.

در [۷] برای بازشناسی برخط حروف مجزای فارسی از دسته بندی کننده ماشین بردار پشتیبان استفاده شده است. این عملیات در دو مرحله صورت گرفته است. در مرحله اول حروف ورودی با استفاده از دسته بندی کننده، در قالب یکی از ۱۸ گروه بدنه اصلی حروف، طبقه بندی شده و سپس در مرحله دوم، موقعیت، تعداد و شکل سایر حرکات‌ها مانند نقطه و سرکش (ریزحرکت‌ها)، حرف نهایی را تعیین کرده است. در این روش هر حرکتی غیر از حرکت بدنه اصلی ریزحرکت نامیده شده است. به عنوان نمونه برای تشخیص حرف «ت» ابتدا گروه بدنه «ب، پ، ت، ث» تشخیص داده شده و سپس وجود ریزحرکت «دو نقطه» در بالای آن منجر به انتخاب «ت» از این گروه شده است. متوسط نرخ بازشناسی بدنه اصلی ۹۴/۳ و با در نظر گرفتن پس‌پردازش‌ها بر اساس ریزحرکت‌ها حدود ۹۸/۰ عنوان شده است. در [۸] مانند روش [۷] عمل شده است، به جز اینکه دسته بندی کننده مدل مخفی مارکوف بکار گرفته شده است. متوسط نرخ بازشناسی بدنه اصلی ۹۳/۰ و با در نظر گرفتن پس‌پردازش‌ها بر اساس ریزحرکت‌ها این نرخ به حدود ۹۶/۰ رسیده است.

در [۹] ابتدا حروف با توجه به شکل و ساختار بدنه اصلی به ۱۸ گروه تقسیم شده‌اند، سپس به کمک مدل مخفی مارکوف پیوسته، بدنه اصلی تشخیص داده شده است. در مرحله بعد، تشخیص نهایی بر اساس ریزحرکات و مدل مخفی آنها صورت گرفته است. نرخ بازشناسی برای گروه و حروف به ترتیب ۹۵/۹ و ۹۴/۲ گزارش شده است.

در [۱۰] حروف مجزای فارسی بر اساس نقاط و علائم بالا یا پایین بدنه اصلی به ۱۲ گروه تقسیم شده‌اند. نخست نقاط و علائم بالا یا پایین هر حرف با یک شبکه عصبی تشخیص شده، سپس با توجه به آن، گروهی که حرف ورودی در آن قرار دارد، تعیین شده است. در صورتی که در گروه مربوطه فقط یک حرف وجود داشته باشد، آن حرف به عنوان خروجی در نظر گرفته شده است. در غیر این صورت بدنه حرف ورودی با بدنه حروفی از پایگاه داده که با حرف ورودی هم گروه هستند مقایسه شده و با طبقه‌بندی کمترین فاصله، بازشناسی انجام گرفته است. نرخ بازشناسی گزارش شده ۹۳/۳ می‌باشد.

در این مقاله برای بازشناسی حروف مجزای فارسی از مدل مخفی مارکوف (HMM) در حالت گسسته استفاده شده است و نتایج بر

### ۱-۱-۲ حذف پیکسل‌های دور از مرکز ثقل پیکسل‌های حرف

در دستگاه‌های مبتنی بر قلم دیجیتال، ممکن است که به علت کم دقتی نویسنده پیکسل‌های ناخواسته‌ای در اثر برخورد قلم با صفحه، ایجاد شوند. این پیکسل‌های ناخواسته، برای سیستم‌های بازشناسی که از اطلاعات مربوط به علائم و نقاط حروف نیز استفاده می‌کنند، سبب کاهش نرخ بازشناسی خواهند شد. نمونه‌ای از این پیکسل‌ها در شکل ۱ مشاهده می‌شود. به منظور شناسایی این پیکسل‌ها، ابتدا میانگین مختصات پیکسل‌های موجود در حرف نوشته شده، شامل بدنه اصلی و ریزحرکات محاسبه شده و فاصله پیکسل‌ها تا میانگین پیکسل‌ها به دست می‌آید. پیکسل‌هایی که در فاصله دورتری از یک حد آستانه نسبت به میانگین پیکسل‌ها قرار دارند، حذف خواهند شد. در روش پیشنهادی حد آستانه ۵ برابر متوسط فاصله‌ها تا میانگین پیکسل‌ها در نظر گرفته شده است.

### ۲-۱-۲ نرمال سازی اندازه

هدف از نرمال سازی اندازه، حذف تغییرات ناشی از نحوه نوشتار در اندازه و نیز ساده سازی شکل حرف است. معمولاً این عمل شامل دو فرایند نرمال سازی اندازه و انتقال پیکسل‌ها می‌باشد [۱۳]. در روش پیشنهادی به منظور نرمال سازی اندازه از روابط ۱ و ۲ استفاده می‌شود.

$$x' = \frac{(x-\bar{x})}{\sigma} * ScaleFactor + O_x \quad (1)$$

$$y' = \frac{(y-\bar{y})}{\sigma} * ScaleFactor + O_y \quad (2)$$

در روابط ۱ و ۲،  $\bar{x}$  و  $\bar{y}$  مختصات پیکسل جدید،  $O_x$  و  $O_y$  مبدا مختصات جدید،  $\bar{x}$  و  $\bar{y}$  میانگین پیکسل‌های موجود و  $ScaleFactor$  ضریب مقیاس دهی است و مقدار  $\sigma$  از رابطه ۳ بدست می‌آید.

$$\sigma = \sqrt{\sigma_x^2 + \sigma_y^2} + d \quad (3)$$

که در آن  $\sigma_x^2$  و  $\sigma_y^2$  به ترتیب واریانس پیکسل‌ها در جهات  $x$  و  $y$  می‌باشد. کمیت  $d$  مقدار ثابتی است که بهترین مقدار برای آن به صورت تجربی برابر با یک بدست آمده است. میزان تاثیر مقادیر مختلف پارامتر  $d$  بر روی نرخ بازشناسی نهایی داده‌های آزمایشی در بخش نتایج نشان داده شده است. مقدار  $ScaleFactor$  برابر با ۵۵ در نظر گرفته شده است. نمونه‌ای از نرمال سازی اندازه، در شکل ۲ قابل مشاهده است.

روی نسخه دوم پایگاه داده Online-TMU [۱-۲] و بخش حروف مجزای آن مورد آزمایش قرار گرفته است. ساختار ادامه مقاله به این صورت است: در بخش ۲، مراحل مختلف روش پیشنهادی از جمله پیش پردازش، نحوه استخراج دنباله مشاهدات، ساختار HMM و همچنین پس پردازشها تشریح می‌شود. نتایج آزمایشها و جمع‌بندی به ترتیب در بخشهای ۳ و ۴ ارائه خواهد شد.

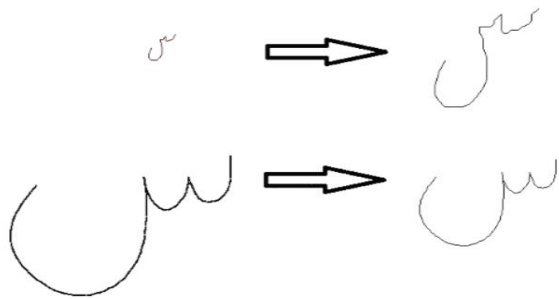
### ۲- روش پیشنهادی

در روش پیشنهادی برای بازشناسی حروف مجزای فارسی از مدل مخفی مارکوف از نوع چپ به راست (Left to Right) استفاده شده است. در اکثر کارهای انجام شده، ابتدا حروف به گروه‌هایی بر اساس شباهت در بدنه اصلی و یا براساس ریزحرکات، تقسیم شده‌اند. سپس فرایند بازشناسی بر روی گروه‌ها انجام گرفته و در نهایت بازشناسی نهایی حروف به کمک ریزحرکات تکمیل شده است [۱۲-۲].

در روش پیشنهادی هیچ گونه گروه بندی انجام نمی‌شود و بدنه اصلی و ریزحرکات آنها همزمان در فرایند استخراج ویژگی و آموزش مدل و بازشناسی نهایی بکار گرفته می‌شوند. روش کار بدین گونه است که ابتدا پیش پردازش‌هایی روی داده‌ها انجام می‌گیرد و سپس ویژگی‌ها استخراج شده و به سیستم، آموزش داده می‌شوند. داده‌های آزمایش به منظور بازشناسی به سیستم اعمال می‌شوند و در نهایت، خطاهای احتمالی به کمک پس‌پردازش تصحیح می‌گردند.

### ۲-۱ پیش پردازش

هدف از پیش پردازش کنار گذاشتن اطلاعات غیر مفیدی است که می‌تواند در فرایند بازشناسی تاثیر منفی داشته باشند. عموماً این اطلاعات غیر مفید می‌تواند شامل نقاط تکراری، قلاب‌های ابتدا و انتهای حرف، نویز و غیره باشد [۱۳]. در روش پیشنهادی، در مرحله پیش پردازش، نقاطی که از میانگین نقاط حروف در فاصله دورتری قرار دارند حذف می‌شوند. نرمال سازی اندازه حروف انجام می‌گیرد تا تمایز بهتری بین حروف بوجود آید. همچنین نقطه‌های مربوط به هر حرف با استفاده از یک پنجره آستانه تعیین می‌شوند. در مرحله بعد قلاب‌های ممکن موجود حذف می‌شوند. سپس هموار سازی روی داده‌های ثبت شده انجام می‌گیرد تا لغزش‌های نوشتاری که ممکن است ناشی از لغزش دست شخص نویسنده و یا به سبب دستگاه باشد را کاهش دهد. پس از آن ویژگی‌های مربوط به هر حرف استخراج شده و در نهایت فرایند آموزش سیستم و آزمایش آن انجام می‌گیرد. در ادامه نحوه انجام هر مورد تشریح می‌گردد.



شکل ۲- نرمال سازی اندازه- شکل های سمت چپ نوشتار اصلی و شکل سمت راست تغییر مقیاس داده شده به کمک روابط ۱ و ۲ و ۳



شکل ۱- تشخیص پیکسل های پرت

### ۲-۱-۳ شناسایی نقطه

با توجه به اینکه نقاط حروف، نقش تعیین کننده‌ای در تشخیص حروف فارسی دارند، بنابراین مشخص نمودن آنها در بازشناسی حروف، اهمیت تعیین کننده‌ای خواهند داشت. از طرفی در نوشتار حروف فارسی، نقطه از تنوع بسیاری برخوردار می باشد که گاهی نقاط با تعداد مختلف در نوشتار بسیار به هم شبیه می‌باشند. نمونه‌ای از تنوع نوشتاری نقاط در شکل‌های ۳ و ۴ برای حرف «ب» و «پ» قابل مشاهده است.

در روش پیشنهادی برای شناسایی تک نقطه از یک پنجره آستانه استفاده شده است. بدین صورت که اگر پس از نرمال سازی اندازه حرف، طول و عرض یک حرکت، از ابعاد این پنجره آستانه کوچکتر باشد، آن حرکت بعنوان تک نقطه در نظر گرفته خواهد شد. در غیر این صورت جهت آموزش به سیستم سپرده می‌شود. حد آستانه مناسب در آزمایشات مقدار ۱۸ پیکسل در جهت افقی و ۱۷ پیکسل در جهت عمودی به دست آمده است.

### ۲-۱-۴ حذف قلاب‌ها

اصطلاح قلاب به حرکاتی از قلم گفته می‌شود که در ابتدا و یا در انتهای حرف ظاهر می‌شوند و معمولاً طول آن نسبت به طول کل بدنه حرف خیلی کمتر بوده و دارای تغییر ناگهانی در حرکت قلم می‌باشند [۱۳-۱۴]. در این مقاله حداکثر طول مجاز قلاب، ۳ درصد طول بدنه و حداکثر زاویه مجاز آن ۹۰ درجه در نظر گرفته شده است. حذف قلاب‌ها در نوشتار حروف فارسی باید با احتیاط باشد.

زیرا برخی از حروف فارسی مانند «ف» و «ق» و «و» در بدنه خود در ابتدا، دارای این دو ویژگی هستند که آنها را از دیگر حروف متمایز می‌کند. در این مقاله از حذف قلاب در انتهای حرف به دلیل عدم تاثیر در نرخ بازشناسی در آزمایشها، صرف نظر شده است. در شکل ۵ نمونه‌ای از حذف قلاب در ابتدای حرف «ر» قابل مشاهده است.

### ۲-۱-۵ هموار سازی

نویز موجود در نوشته‌های برخط می‌تواند ناشی از محدودیت دقت صفحه لمسی، فرایند رقمی کردن داده‌ها، لغزش دست هنگام نوشتن و کم دقتی در قرار دادن یا برداشتن قلم باشد [۱۵]. روشهای مختلفی برای حذف نویز و هموارسازی داده‌ها وجود دارد. این لرزش‌ها خصوصاً هنگام بزرگنمایی در حالت مقیاس دهی، اثر نامطلوبی خواهند داشت. در این مقاله به منظور هموارسازی، از میانگین مختصات مکانی سه پیکسل مجاور، طبق روابط ۴ و ۵ استفاده شده است.

$$\hat{x}_i = \frac{x_{i-1} + x_i + x_{i+1}}{3} \quad i \neq 0, l-1 \quad (4)$$

$$\hat{y}_i = \frac{y_{i-1} + y_i + y_{i+1}}{3} \quad i \neq 0, l-1 \quad (5)$$



شکل ۳- نمونه حرف «ب» با تنوع نوشتاری تک نقطه



شکل ۴- نمونه حرف «پ» با تنوع نوشتاری در سه نقطه

شکل کلی حرف نگهداری می‌شوند. از خط ۸ تا ۱۴ با استفاده از درونیایی خطی، مختصات پیکسل‌های جدید محاسبه می‌شوند. مقدار مناسب برای  $T$  در این مقاله با توجه به مقیاس دهی گفته شده در مراحل قبلی، برابر با ۲۶ بدست آمده است. میزان تاثیر مقادیر مختلف  $T$  بر روی نرخ بازشناسی نهایی داده‌های آزمایشی در بخش نتایج نشان داده شده است. نتایج یکسان سازی تعداد و فاصله پیکسل‌ها برای دو حرف نمونه در شکل ۷ قابل مشاهده است.

- 1:  $In: S = \{(x_0, y_0), \dots, (x_{n-1}, y_{n-1})\}, T$
- 2:  $Out: \hat{S} = \{(\hat{x}_0, \hat{y}_0), \dots, (\hat{x}_{T-1}, \hat{y}_{T-1})\}$
- 3:  $L_0 = 0$  // accumulated stroke length
- 4: for  $i=1$  to  $n-1$
- 5:  $L_i = L_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$
- 6:  $a = L_{n-1}/T$  // distance between points
- 7:  $(\hat{x}_0, \hat{y}_0) = (x_0, y_0)$  // keep the first point
- 8:  $j=1$
- 9: for  $p=1$  to  $T-2$
- 10: while  $L_j < p * a$
- 11:  $j++$
- 12:  $C = (p * a - L_{j-1}) / (L_j - L_{j-1})$
- 13:  $\hat{x}_p = x_{j-1} + (x_j - x_{j-1}) * C$
- 14:  $\hat{y}_p = y_{j-1} + (y_j - y_{j-1}) * C$
- 15:  $(\hat{x}_{T-1}, \hat{y}_{T-1}) = (x_{n-1}, y_{n-1})$  // keep the last point



شکل ۶- هموارسازی با استفاده از میانگین مختصات مکانی سه پیکسل مجاور



شکل ۷- یکسان سازی فاصله و تعداد پیکسل‌ها با  $T=26$  سمت چپ، نمونه اصلی و سمت راست، نمونه حرف پس از یکسان سازی تعداد و فاصله پیکسل‌ها

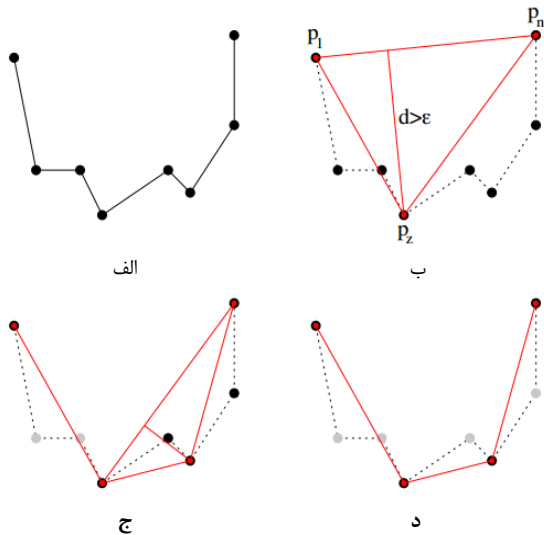


شکل ۸- حذف قلاب از ابتدای حرف «ر»- شکل سمت چپ نوشتار اصلی و شکل سمت راست نوشتار پس از حذف قلاب ابتدایی حرف

که در آن  $\hat{x}_i$  و  $\hat{y}_i$  مختصات مکانی پیکسل‌های جدید می‌باشند. مختصات مکانی پیکسل‌های ابتدایی و انتهایی بدون تغییر حفظ شده‌اند. شکل ۹ سمت چپ حرف «س» را پس از نرمال سازی اندازه و شکل سمت راست، هموار شده آن را به کمک روابط ۴ و ۵ نمایش می‌دهد.

#### ۶-۱-۲ یکسان سازی تعداد نمونه‌ها و فاصله بین آنها

صفحه رقومی کننده<sup>۴</sup>، در فواصل زمانی ثابتی از حرکت قلم نمونه برداری می‌کند، در حالیکه سرعت نوشتن افراد مختلف، حتی برای یک نویسنده بسته به شرایط مختلف، ممکن است برای حرف‌های یکسان هم متفاوت باشد [۱۶]. اگر سرعت نوشتن کم باشد، تعداد پیکسل‌ها زیاد شده و فاصله بین آنها کم می‌شود. در حالیکه این پیکسل‌های با فاصله کم، ممکن است اطلاعات مفید جدیدی ارائه نکنند. از طرفی با افزایش سرعت نوشتن، فاصله پیکسل‌ها زیاد شده و در نتیجه تعداد آنها کاهش می‌یابد. در روش پیشنهادی به منظور یکسان سازی تعداد نمونه‌ها و فاصله بین آنها از الگوریتم ارائه شده در [۱۶] استفاده شده است. با توجه به نتایج آزمایش‌ها، داشتن تعداد پیکسل‌های یکسان برای بازشناسی حروف مشابه، تاثیر بیشتری در بازشناسی داشته است. بنابراین الگوریتم ارائه شده در [۱۶] که  $a$  را بعنوان ورودی برای فاصله بین پیکسل‌ها دریافت می‌کرد و در نتیجه تعداد پیکسل‌های متفاوتی بدست می‌داد، جهت بدست آوردن تعداد پیکسل‌های یکسان، کمی تغییر داده شده است. در الگوریتم تغییر یافته،  $T$  یعنی تعداد جدید پیکسل‌ها، بعنوان ورودی در نظر گرفته شده است. در نتیجه تعداد پیکسل‌ها برای بدنه اصلی حروف یکسان خواهد بود. در ادامه، این الگوریتم پس از اعمال تغییرات لازم آورده شده است. در این الگوریتم،  $S$  مجموعه پیکسل‌های موجود در حرف و  $T$  تعداد مطلوب پیکسل‌ها بعنوان ورودی هستند و  $\hat{S}$  مجموعه مختصات پیکسل‌های جدید می‌باشند. آرایه  $L$  فاصله پیکسل‌ها از پیکسل ابتدایی را در خود نگه می‌دارد. در خط ۷ و ۱۵ از شبه کد بالا، پیکسل‌های شروع و انتها جهت حفظ



شکل ۸ - «الف» مجموعه نقاط منحنی اصلی. «ب» دورترین نقطه نسبت به خط واصل  $P_1$  و  $P_n$  حفظ می‌شود چون  $d > \epsilon$ . «ج» الگوریتم به صورت بازگشتی قسمت سمت راست و چپ را بررسی می‌کند. در نیمه چپ، نقاط خاکستری حذف می‌شوند، چون فاصله آنها نسبت به خط واصل  $P_1$  و  $P_2$  کمتر از  $\epsilon$  است. در نیمه راست، دورترین نقطه حفظ می‌شود. در «د» نقاط خاکستری به علت نزدیکی بیش از حد حذف می‌شوند [۱۸].

### ۲-۲-۱ ویژگی جهت حرکت قلم

برای این ویژگی از زنجیره کد ۸ تایی استفاده شده است. برای این کار، زاویه بین خط واصل بین دو پیکسل نسبت به محور X ها محاسبه شده، سپس با توجه به زاویه بدست آمده، کدی بین ۰-۷ به آن نسبت داده می‌شود. در روش پیشنهادی، زاویه بین پیکسل انتها و ابتدا نیز در نظر گرفته شده است. شکل‌های ۱۰ و ۱۱ این فرایند را نشان می‌دهند در شکل ۱۰ مقدار  $\theta_i$  از رابطه زیر به دست می‌آید:

$$\theta_i = \tan^{-1} \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \quad (6)$$

به عنوان مثال اگر  $67/5 < \theta_i \leq 22/5$  باشد، کد ۱ برای زاویه حرکت نسبت داده می‌شود.

### ۲-۲-۲ ویژگی گذاشتن و برداشتن قلم

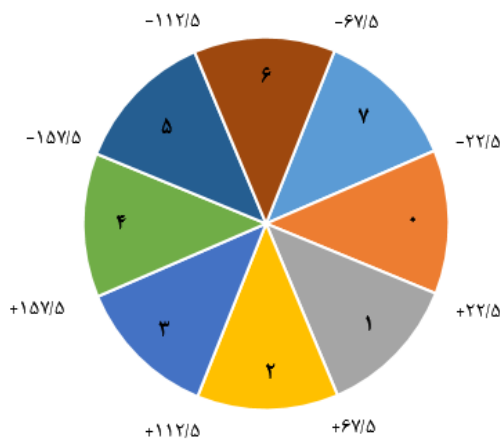
برخی حروف فارسی از نظر تعداد حرکات پیوسته دست، متفاوت می‌باشند. بنابراین تعداد گذاشتن و برداشتن قلم از روی صفحه بعنوان ویژگی متمایز کننده بکار رفته است. در روش پیشنهادی کد ۸ برای برداشتن قلم و کد ۹ برای گذاشتن قلم در نظر گرفته شده است. جهت تاکید بیشتر، هر کدام از آنها در صورت وقوع، دو بار تولید (یک بار تکرار) می‌شوند.

### ۷-۱-۲ کاهش تعدا پیکسل‌ها با الگوریتم داگلاس پوکر

هدف از بکارگیری الگوریتم داگلاس پوکر [۱۷] حل مسئله ساده‌سازی منحنی می‌باشد. در واقع منحنی با  $n$  رأس با میزان خطای مشخصی تخمین زده می‌شود. در این الگوریتم منحنی با  $n$  نقطه،  $P = p_1, \dots, p_n$  و میزان خطای مجاز،  $\epsilon$ ، بعنوان ورودی و مجموعه نقاط  $Q = q_1, \dots, q_m$  با  $m$  نقطه بعنوان خروجی الگوریتم است که  $m \leq n$  است و هر نقطه از  $Q$  در  $P$  قرار دارد. الگوریتم داگلاس-پوکر به صورت بازگشتی، چند ضلعی ورودی را قطعه بندی کرده و هر قطعه را به طور مجزا مدیریت می‌کند. این الگوریتم ابتدا قطعه خطی را بین نقاط ابتدایی و انتهایی ( $p_1$  و  $p_n$ ) در نظر می‌گیرد. سپس  $p_z$  یعنی نقطه‌ای از مجموعه  $P$  که در دورترین فاصله از این قطعه خط قرار دارد را می‌یابد. اگر فاصله  $p_z$  و  $d$ ، از  $\epsilon$  کمتر باشد تمام نقاط بین نقاط ابتدایی و انتهایی منحنی حذف می‌شوند. در غیر این صورت  $p_z$ ، در مجموعه نقاط تخمین  $Q$  قرار داده می‌شود و الگوریتم به صورت بازگشتی تکرار می‌شود (شکل‌های ۸ الف تا د) [۱۸]. در روش پیشنهادی از الگوریتم داگلاس-پوکر جهت کاهش تعداد پیکسل‌های ریز حرکت استفاده شده است.

### ۲-۲-۲ استخراج ویژگی به منظور ایجاد دنباله مشاهدات

هدف از استخراج ویژگی، استخراج اطلاعاتی از داده‌های خام است که برای طبقه بندی داده‌ها مناسب باشند. یعنی تغییرات الگویی درون کلاسی را کاهش و بین کلاسی را افزایش دهد [۱۹]. در این تحقیق با توجه به اینکه از مدل مخفی مارکوف<sup>۵</sup> جهت طبقه بندی و بازشناسی استفاده شده است، بنابراین ویژگی‌های استخراج شده بعنوان رویدادهای قابل مشاهده برای HMM محسوب می‌شوند. رویدادهای قابل مشاهده می‌توانند به دو صورت پیوسته و گسسته مدل شوند [۲۰-۲۱]. در این مقاله مدل گسسته بکار گرفته شده است. روش معمول برای حالت گسسته این است که ابتدا ویژگی‌های مشاهده شده در قالب بردار ویژگی، توسط توابع چگالی احتمال پیوسته مدل شده و سپس بوسیله چندی سازی برداری<sup>۶</sup>، به کمیت گسسته تبدیل شوند. برای هر کمیت گسسته بدست آمده، کدی از میان کدهای تعریف شده در کتاب کد<sup>۷</sup>، نسبت داده می‌شود تا دنباله مشاهدات گسسته برای حرف مورد نظر بدست آید [۲۲]. این فرایند در شکل ۹ نشان داده شده است. ویژگی‌های استخراج شده شامل بخش‌های مختلفی است که در ادامه تشریح خواهند شد.



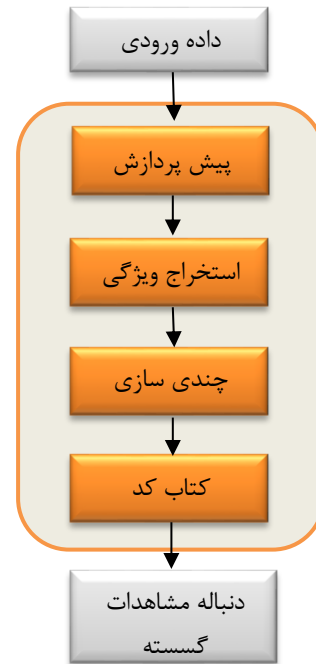
شکل ۱۱ - نحوه اختصاص کد ۸ تایی به زاویه

### ۳-۲-۲ ویژگی موقعیت ریزحرکات

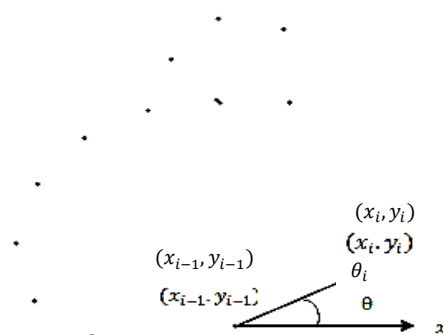
از جمله ویژگی‌های متمایز کننده حروف از یکدیگر، موقعیت قرار گرفتن ریزحرکت نقطه یا سرکش نسبت به بدنه اصلی حرف می‌باشد. ریزحرکت می‌تواند بالا یا پایین بدنه اصلی باشد. ویژگی دیگری که می‌توان جهت تمایز حروفی مثل «ک» و «گ» از دیگر حروف در نظر گرفت، قرار گرفتن ریزحرکت در سمت راست بدنه می‌باشد. در روش پیشنهادی، اگر پیکسل مورد نظر از ریزحرکت در پایین بدنه اصلی قرار داشته باشد، کد ۱۰ و اگر بالای بدنه اصلی قرار داشته باشد، کد ۱۱ به موقعیت مکانی پایین/بالا بودن پیکسل ریزحرکت نسبت داده می‌شود.

به منظور تعیین موقعیت بالا/پایین بودن نسبت به بدنه اصلی از الگوریتم ارایه شده در [۱۰] استفاده شده است. به این صورت که اگر  $y_m$  مربوط به مرکز چارچوب ریزحرکت، بزرگتر از  $y_{max}$  یا کوچکتر از  $y_{min}$  چارچوب بدنه اصلی باشد، موقعیت آن به ترتیب پایین یا بالا در نظر گرفته می‌شود. در غیر این صورت، بالا یا پایین بودن اجزای ریزحرکت نسبت به موقعیت دو پیکسل  $p_1$  و  $p_2$  از بدنه اصلی تعیین می‌شود. بنابراین، ابتدا باید موقعیت پیکسل‌های  $p_1$  و  $p_2$  مشخص شود. برای اینکار روی پیکسل‌های بدنه اصلی حرکت می‌کنیم و جفت پیکسل‌های مجاور را در نظر می‌گیریم. فرض کنیم مرکز چارچوب ریزحرکت، پیکسل با موقعیت مکانی  $p(x_m, y_m)$  باشد. در مسیر حرکت بر روی بدنه اصلی، اولین جفت پیکسل مجاور که برای آنها شرط  $x_m > \min(x_1, x_2)$  برقرار باشد، به عنوان پیکسل‌های  $p_1(x_1, y_1)$  و  $p_2(x_2, y_2)$  انتخاب می‌شوند. پس از تعیین  $p_1$  و  $p_2$ ، اگر  $y_m < \min(y_1, y_2)$  باشد، موقعیت مکانی پایین به اجزای ریزحرکت نسبت داده می‌شود. در شکل ۱۲ موقعیت

در روش پیشنهادی، ویژگی‌هایی از ریزحرکات استخراج شده‌اند مانند گذاشتن و برداشتن قلم، موقعیت ریزحرکت، داشتن نقطه و همچنین موقعیت نقطه‌ها نسبت به بدنه اصلی. زاویه حرکت دست از یک پیکسل به پیکسل بعدی، نسبت به محور افقی، مانند قبل با کد زنجیره‌ای ۸ تایی مدل می‌شوند و کدهای ۰ تا ۷ برای آن در نظر گرفته می‌شوند. در ادامه در مورد سایر ویژگی‌های مربوط به ریزحرکات توضیح داده می‌شود.



شکل ۹- روال کلی استخراج دنباله مشاهدات در HMM گسسته



شکل ۱۰- زاویه هر نقطه نسبت به محور افقی



۶-۲-۲ - کتاب کد

با توجه به مطالب گفته شده، کتاب کدی که برای ایجاد دنباله مشاهدات گسسته از ویژگی‌های استخراج شده در نظر گرفته شده است به صورت جدول ۱ خواهد بود.

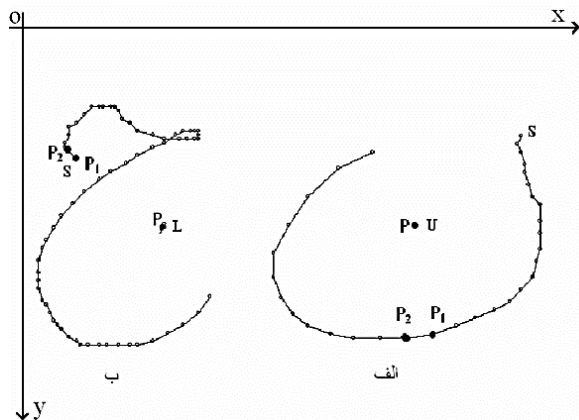
۷-۲-۲ - نحوه استخراج ویژگی‌های حرف

در این مقاله فرض بر این است که بدنه اصلی حرف در ابتدا و با یک حرکت قلم نوشته شده باشد. هرچند در پایگاه داده، نوشتار به روش غیرمعمول نیز وجود دارد. در شکل ۱۳ نمونه‌هایی از نوشتار معمول و غیر معمول مشاهده می‌شود. اعداد نشان داده شده در کنار اجزای حروف، محل شروع و ترتیب نوشتن را نشان می‌دهد. در آزمایش‌ها، هیچ یک از نوشتارهای معمول و غیر معمول از مجموعه داده حذف نشده‌اند و همه داده‌ها در فرایند آموزش و بازشناسی شرکت داده شده‌اند.

۸-۲-۲ - ترتیب پردازش‌های لازم برای استخراج ویژگی‌ها

پردازش‌های لازم به منظور استخراج ویژگی‌ها به ترتیب زیر انجام می‌شود:

- ۱- حذف پیکسل‌های بسیار دور از مرکز ثقل پیکسل‌های کلی حرف
- ۲- نرمال سازی کل پیکسل‌های موجود در حرف (شامل بدنه اصلی و ریزحرکات در صورت وجود)



شکل ۱۲- تعیین موقعیت ریزحرکت [۱۰]

پیدا شده برای پیکسل‌های  $p_1$  و  $p_2$  در دو مثال نشان داده شده است. در شکل ۱۲-الف، موقعیت پیکسل‌های ریزحرکت، بالا و در شکل ۱۲-ب، پایین به دست می‌آیند.

برای ویژگی سمت راست یا چپ قرار داشتن پیکسل ریزحرکت نسبت به بدنه اصلی، اگر مقدار  $x_i$  ریزحرکت بزرگتر از  $x_{max}$  بدنه اصلی (ویژگی سمت راست بودن) باشد کد ۱۲ و در غیر این صورت (ویژگی سمت راست نبودن) کد ۱۳ برای آن پیکسل ریزحرکت در نظر گرفته می‌شود.

۴-۲-۲ - ویژگی نقطه‌های حروف

در حروف فارسی، برخی حروف با توجه به داشتن نقطه، تعداد آنها و موقعیت آنها از یکدیگر متمایز می‌شوند. در روش پیشنهادی، اگر تعداد نقطه‌ها بیشتر از یکی باشد، در هم ادغام شده و به عنوان یک ریزحرکت پیوسته در نظر گرفته می‌شوند و ویژگی‌های بیان شده در بخش ۱-۲-۲ تا ۳-۲-۲ برای آن استخراج خواهند شد. برای وضعیت تک نقطه سه حالت در نظر گرفته شده است: بدون نقطه، بالای بدنه اصلی و پایین بدنه اصلی.

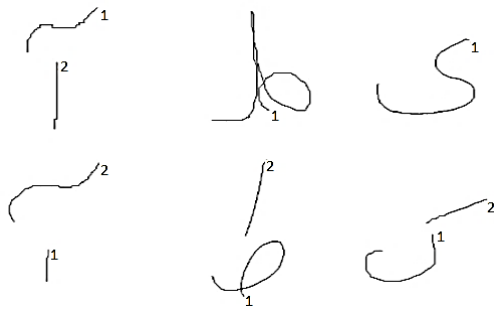
کدهای ۱۴ و ۱۵ و ۱۶ به ترتیب برای حالت بدون نقطه، بالا و پایین بدنه اصلی اختصاص یافته‌اند. جهت تعیین موقعیت نقطه نسبت به بدنه اصلی از الگوریتم ارایه شده در بخش ۳-۲-۲ استفاده شده است. در اینجا نیز جهت تاکید بر اهمیت، هر کد دو بار تولید می‌شوند. میزان تاثیر تکرار کدها در بخش نتایج به ازای حالت‌های مختلف آورده شده است.

۵-۲-۲ - ویژگی رسیدن به انتهای مشاهدات

در روش پیشنهادی از طبقه بندی کننده HMM از نوع چپ به راست با یک حالت پایانی استفاده شده است که آخرین حالت، تنها قادر به مشاهده ویژگی رسیدن به انتهای مشاهدات (PTO) می‌باشد. در بخش ۲-۳ در مورد ساختار HMM توضیح بیشتری بیان خواهد شد. در این مقاله برای PTO کد ۱۷ در نظر گرفته شده است.

جدول ۱ - کتاب کد ویژگی‌ها

کد	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷
توضیح کد	جهت حرکت قلم	گذاشتن قلم	برداشتن قلم	ریزحرکت پایین بدنه	ریزحرکت بالای بدنه	ریزحرکت سمت راست بدنه نباشد	ریزحرکت سمت راست بدنه نباشد	انتهای مشاهدات
تعداد تکرار کد در دنباله مشاهدات	بدون تکرار	یک بار تکرار	بدون تکرار	بدون تکرار	بدون تکرار	یک بار تکرار	یک بار تکرار	بدون تکرار



شکل ۱۳- نوشتار معمول و غیر معمول حروف. سطر اول، نوشتار غیر معمول حروف «آ» و «ط» و «ک» و سطر دوم نوشتار معمول

رشته کد "2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-6" برای حرف «آ» در شکل ۱۴- الف (کدها از سمت چپ خوانده شوند) کدهای مربوط به زاویه جفت پیکسل‌های متوالی نسبت به محور افقی می‌باشند. در این رشته کد، کد "6"، کد زاویه پیکسل انتهایی نسبت به پیکسل ابتدا را نشان می‌دهد. رشته کد "11-11-11-11-11-11" مربوط به بالای بدنه اصلی بودن پیکسل‌های ریز حرکت هستند.

کدهای "12-12-12-13-13-13" نشان می‌دهند که سه پیکسل اول ریز حرکت در سمت راست و سه پیکسل بعدی در سمت چپ بدنه اصلی قرار دارند. رشته کد "3-4-4-3-2-0" کدهای زاویه حرکت قلم برای پیکسل‌های ریز حرکت می‌باشند. کدهای "14-14" نشان می‌دهند که این حرف بدون نقطه می‌باشد. "17" کد رسیدن به انتهای دنباله مشاهدات را مشخص می‌کند.

در شکل ۱۴- ب کدهای "16-16" در انتهای دنباله مشاهدات نشان می‌دهد که نقطه حرف با توجه به الگوریتم ارائه شده در بخش ۳-۲ در پایین بدنه اصلی قرار دارد. این دنباله کدهای استخراج شده، بعنوان دنباله مشاهدات گسسته برای HMM بکار گرفته خواهند شد. در ادامه نحوه اعمال این ویژگی‌ها به HMM ارائه خواهد شد.

۲-۳- مدل مخفی مارکوف (HMM)

در روش پیشنهادی از مدل مخفی مارکوف گسسته از نوع چپ به راست برای طبقه بندی دست نوشته‌های برخط استفاده شده است. فرض کنید  $S = \{S_0, S_1, \dots, S_{N-1}\}$  مجموعه حالت‌های مخفی در مدل  $V = \{v_0, v_1, \dots, v_{M-1}\}$  مجموعه نشانه‌های (ویژگی‌های) قابل مشاهده برای مدل باشند.

دنباله حالت‌های مخفی  $Q = q_0, q_1, \dots, q_{T-1}$  تولید شده که عناصر آن از مجموعه  $S$  و دنباله  $O = o_0, o_1, \dots, o_{T-1}$  ویژگی‌های مشاهده شده در هر حالت باشند.

۳- تشخیص نقطه‌ها در صورت وجود

۴- حذف قلاب‌های بدنه اصلی

۵- محاسبه  $x_{max}$  و  $y_{max}$  و  $x_{min}$  و  $y_{min}$  بدنه اصلی

۶- نرمال سازی اندازه بدنه اصلی

۷- هموار سازی بدنه اصلی

۸- یکسان سازی تعداد و فاصله بین پیکسل‌های بدنه اصلی

۹- تولید کد گذاشتن قلم برای بدنه اصلی

۱۰- استخراج کدهای زاویه حرکت قلم نسبت به محور  $x$  برای همه پیکسل‌های بدنه اصلی

۱۱- تولید کد برداشتن قلم

۱۲- ادغام نقطه‌ها در صورت وجود و تبدیل آنها به یک ریز حرکت.

در صورتی که حرف شامل بیش از یک حرکت (غیر از نقطه) باشد ترتیب زیر برای استخراج کدهای مربوط به ریز حرکت دنبال می‌شود:

۱۳- هموار سازی ریز حرکت

۱۴- کاهش تعداد پیکسل‌های ریز حرکت با الگوریتم داگلاس-پوکر

۱۵- تولید کد گذاشتن قلم بر روی صفحه

۱۶- تعیین بالا/پایین بودن برای همه پیکسل‌های ریز حرکت نسبت به بدنه اصلی

۱۷- تعیین کد سمت راست بودن/نبودن برای همه پیکسل‌های ریز حرکت نسبت به بدنه اصلی

۱۸- تعیین کد زاویه حرکت قلم نسبت به محور  $x$ ها برای همه پیکسل‌های ریز حرکت

۱۹- تولید کد برداشتن قلم از صفحه

۲۰- پس از استخراج کدهای مربوط به بدنه و ریز حرکت (غیر از تک نقطه) ویژگی‌های مربوط به تک نقطه (بدون نقطه، بالا و پایین بودن) و نیز کد انتهای مشاهدات (PTO) به دنباله کدهای تولید شده اضافه می‌شوند.

در شکل‌های ۱۴- الف و ۱۴- ب کدهای استخراج شده برای دو حرف نمونه قابل مشاهده است. در این شکل‌ها برای وضوح بیشتر، تعداد و فاصله پیکسل‌های بدنه اصلی با  $T=15$  تنظیم شده‌اند. شکل ۱۴- الف نمونه کد استخراج شده را برای حرف «آ» نشان می‌دهد.

در این شکل، با توجه به تنظیم پارامتر میزان خطای مجاز،  $\epsilon = 3$  در الگوریتم داگلاس-پوکر، تعداد پیکسل‌های ریز حرکت به ۶ پیکسل کاهش یافته است. در کدهای استخراج شده، کدهای "9-9" کد گذاشتن قلم و "8-8" کد برداشتن قلم هستند.

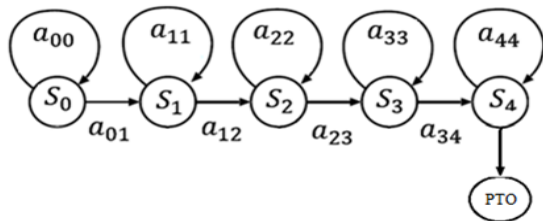
### ۲-۳-۲ آموزش مدل

در روش پیشنهادی جهت آموزش مدل و در نهایت محاسبه پارامترهای A, B (در مدل چپ به راست، پارامتر  $\pi$  نیاز به محاسبه ندارد) از الگوریتم بام-ولش که روابط آن در [۲۰] آمده است، استفاده شده است.

یکی از مسائلی که در تخمین پارامترهای HMM با استفاده از الگوریتم بام-ولش وجود دارد، تعیین مقادیر اولیه  $a_{ij}$  و  $b_j(k)$  جهت دستیابی به ماکزیمم کلی می باشد. در نظر گرفتن مقادیر تصادفی و یا یکنواخت برای  $a_{ij}$  ها می تواند تقریباً در همه موارد تخمین مناسبی از پارامتر را بدست دهد. تجربه نشان داده است که تخمین اولیه مناسب از  $b_j(k)$  می تواند در مدل مخفی مارکوف گسسته، مفید و در حالت پیوسته، حیاتی باشد. روشهای مختلفی برای تخمین مقادیر اولیه عناصر ماتریس B وجود دارد [۲۰]. مقادیر اولیه در نظر گرفته شده در روش پیشنهادی به صورت روابط ۷ و ۸ می باشند.

$$\pi_i = \begin{cases} 1 & i = 0 \\ 0 & i \neq 0 \end{cases} \quad (7)$$

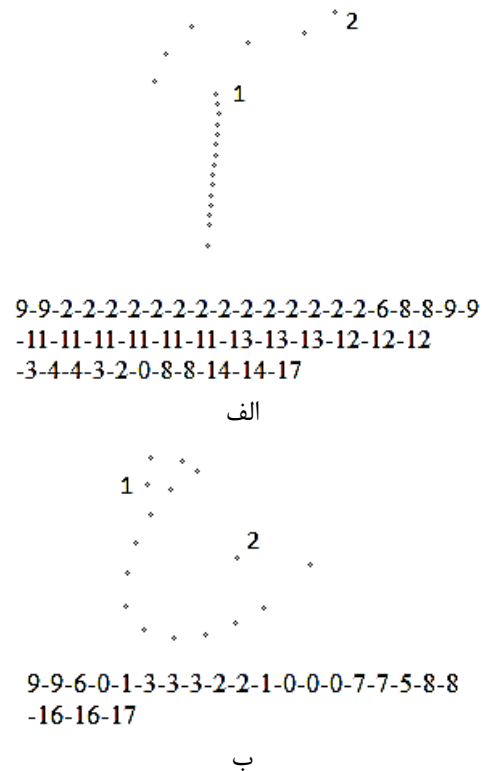
$$a_{ij} = \begin{cases} \frac{1}{2} & (j = i \text{ or } j = i + 1) \text{ and } i, j \neq N - 1 \\ 0 & j \neq i \text{ and } j \neq i + 1 \text{ and } i, j \neq N - 1 \\ 1 & i = j = N - 1 \end{cases} \quad (8)$$



شکل ۱۵- مدل چپ به راست با ۴ حالت و یک حالت پایانی که تنها قادر به مشاهده PTO است.

$$b_j(k) = \begin{cases} \frac{1}{M-1} & j \neq N - 1, k \neq M - 1 \\ \epsilon & j = N - 1, k \neq M - 1 \\ \epsilon & k = M - 1, j \neq N - 1 \\ 1 & j = N - 1, k = M - 1 \end{cases} \quad (9)$$

که M، تعداد ویژگی‌های قابل مشاهده، N، تعداد حالت‌ها و  $\epsilon$  احتمال مشاهده PTO توسط هر حالت است. به نظر می‌رسد که مقدار  $\epsilon$  می‌بایست برای حالت‌ها به جز حالت پایانی برابر صفر باشد. اما در آزمایش‌ها مشاهده شد که نرخ بازشناسی به ازای  $\epsilon \neq 0$  و برای مقادیر کوچک آن اندکی سبب بهبود در نرخ بازشناسی



شکل ۱۴- نمونه دنباله مشاهدات گسسته (کد استخراج شده) برای حروف «آ» و «ج» با T=۱۵

HMM به صورت  $\lambda = (A, B, \pi)$  ارائه می‌شود که در آن  $A = \{a_{ij} | 0 \leq i, j \leq N - 1\}$  ماتریس توزیع گذر حالات و  $B = \{b_j(k) | 0 \leq j \leq N - 1, 0 \leq k \leq M - 1\}$  ماتریس توزیع انتشار نشانه‌ها و  $\pi = \{\pi_i | 0 \leq i \leq N - 1\}$  ماتریس توزیع آغازین حالات می‌باشند. بدلیل خاصیت احتمالی  $a_{ij}$  و  $b_j(k)$  شرایط  $\sum_{k=0}^{M-1} b_j(k) = 1$  و  $\sum_{j=0}^{N-1} a_{ij} = 1$  در ادامه ساختار HMM استفاده شده در روش پیشنهادی بیان می‌شود.

### ۲-۳-۱ ساختار HMM

ساختار HMM می‌تواند در دو نوع ارگودیک<sup>۱</sup> و غیرارگودیک<sup>۲</sup> باشد. در ساختار ارگودیک می‌توان از هر حالت با طی کردن دفعات محدودی به حالت دیگری رسید. در ساختار غیرارگودیک به برخی حالات نمی‌توان از حالات دیگر رسید. در این مقاله از HMM نوع چپ به راست که حالت خاصی از ساختار غیرارگودیک است و در آن، حالت‌ها اجازه انتقال به حالت‌های قبل از خود را ندارند [۲۰]. با در نظر گرفتن یک حالت بعنوان حالت پایانی، استفاده شده است. حالت پایانی، فقط قادر به مشاهده ویژگی رسیدن به انتهای مشاهدات (PTO) می‌باشد. در شکل ۱۵، شمای ساختار HMM بکار گرفته شده در روش پیشنهادی مشاهده می‌شود.

حال اگر  $Z = \{O^0, O^1, \dots, O^k\}$  باشد یعنی کلاس  $Z$  (مثلا حرف «آ») دارای  $K$  نمونه مختلف و مستقل که هر کدام دارای دنباله ویژگی  $O^k$  است، احتمال هر کدام از نمونه‌ها در مدل  $\lambda$  از رابطه ۱۵ بدست می‌آید که در رابطه ۱۶ بازنویسی شده است. هدف از بازتخمین پارامترهای  $\lambda$  حداکثر کردن مقدار رابطه ۱۷ است [۲۰].

$$P_k = P(O^k | \lambda) \quad (16)$$

$$P(Z|\lambda) = \prod_{k=0}^{K-1} P(O^k|\lambda) \equiv \prod_{k=0}^{K-1} P_k \quad (17)$$

در این صورت بازتخمین دارایی‌های ماتریس  $A$  و  $B$  به صورت روابط ۱۸ و ۱۹ خواهد بود.

$$\bar{a}_{ij} = \quad (18)$$

$$\frac{\sum_{k=0}^{K-1} \frac{1}{P_k} \sum_{t=0}^{T_k-2} \alpha^k_t(i) \alpha_{ij} b_j(O^k_{t+1}) \beta^k_{t+1}(j)}{\sum_{j=0}^{N-1} \sum_{k=0}^{K-1} \frac{1}{P_k} \sum_{t=0}^{T_k-2} \alpha^k_t(i) \alpha_{ij} b_j(O^k_{t+1}) \beta^k_{t+1}(j)}$$

$$\bar{b}_j(l) = \frac{\sum_{k=0}^{K-1} \frac{1}{P_k} \sum_{t=0, O_t=v_l}^{T_k-1} \alpha^k_t(j) \beta^k_t(j)}{\sum_{k=0}^{K-1} \frac{1}{P_k} \sum_{t=0}^{T_k-1} \alpha^k_t(j) \beta^k_t(j)} \quad (19)$$

بطور ایده‌آل برای بازتخمین کامل همه پارامترهای مدل به بینهایت داده نیاز است. ولی در عمل تعداد داده آموزشی محدود است. بنابراین اگر نشانه  $v_l$  امکان مشاهده آن توسط حالت  $S_j$  باشد، ولی در داده آموزشی جاری وجود موجود نباشد، بعلت این که  $b_j(k) = 0$ ، احتمال مدل، هنگام وجود این نشانه در ویژگی‌ها صفر خواهد شد. بنابراین حداقل احتمال مشاهده نشانه‌ها لازم می‌شود. راه حل ساده برای این کار، پس پردازش فرآیند آموزش است. طوریکه حداقل احتمال مشاهده هر نشانه،  $\Delta$ ، پس از هر بار بازتخمین به کلیه عناصر ماتریس  $B$  افزوده می‌شود [۲۳]. بنابراین خواهیم داشت:

$$b^*_j(k) = b_j(k) + \Delta \quad (20)$$

بعلت داشتن خاصیت احتمالی، عناصر ماتریس  $B$  به صورت زیر بروز رسانی خواهند شد.

$$b_j(k) = \frac{b^*_j(k)}{\sum_{k=0}^{M-1} b^*_j(k)}, \quad (21)$$

در روابط ۲۰ و ۲۱ شرایط  $0 \leq k \leq M-1$  و  $0 \leq j \leq N-1$  برقرار است. اثر مقادیر مختلف پارامتر  $\Delta$  در نرخ بازشناسی در بخش نتایج نشان داده شده است.

اگر پارامترهای مدل بازتخمین شده با  $\bar{\lambda}$  نمایش داده شود، فرآیند بازتخمین می‌تواند تا حصول شرایط خاصی ادامه یابد. در این مقاله از دو شرط زیر به صورت همزمان استفاده شده است.

$$|\log P(Z|\bar{\lambda}) - \log P(Z|\lambda)| > \delta \quad (22)$$

$$Iteration\ Times < \tau \quad (23)$$

می‌شود. فرآیند بکار رفته به این صورت بوده است که در ابتدا به عنوان مقدار اولیه و نیز پس از بازتخمین ماتریس  $B$  مقدار  $\epsilon$  برای درایه‌های متناظر آن در رابطه ۹، مقداری کوچک و مخالف صفر در نظر گرفته شده است. همچنین همواره  $b_{N-1}(M-1) = 1$  در نظر گرفته می‌شود. میزان تاثیر مقادیر مختلف  $\epsilon$  بر روی نرخ بازشناسی در بخش نتایج آورده شده است. چون باید  $\sum_{k=0}^{M-1} b_j(k) = 1$  بنابراین ماتریس  $B$  به صورت زیر اصلاح می‌شود.

$$b^*_j(k) \frac{b_j(k)}{\sum_{k=0}^{M-1} b_j(k)} \quad (10)$$

از این به بعد  $b_j(k)$  همان  $b^*_j(k)$  در نظر گرفته شده است.

#### • متغیرهای پس‌رو-پیش‌رو

طبق تعریف متغیر پیش‌رو  $\alpha_t(j)$  به صورت زیر محاسبه می‌شود.

۱- مقدار دهی اولیه

$$\alpha_0(j) = \pi_j b_j(O_0), 0 \leq j \leq N-1 \quad (11)$$

۲- بازگشتی

$$\alpha_{t+1}(j) = \sum_{i=0}^{N-1} \alpha_t(i) a_{ij} * b_j(O_{t+1}) \quad (12)$$

در رابطه ۱۲ شرایط  $0 \leq j \leq N-1$  و  $0 \leq t \leq T-2$  برقرار است.

و متغیر پس‌رو  $\beta_t(t)$  به صورت زیر محاسبه می‌شود.

۱- مقدار دهی اولیه

$$\beta_T(i) = 1, 0 \leq i \leq N-1 \quad (13)$$

۲- بازگشتی

$$\beta_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad (14)$$

در رابطه ۱۴ شرایط  $0 \leq j \leq N-1$  و  $0 \leq t \leq T-2$  برقرار است.

### ۳-۲-۳ بازتخمین پارامترهای مدل $\lambda$

اگر  $O$  یک مجموعه ویژگی مشاهده شده باشد، احتمال آن در مدل  $\lambda = (A, B, \pi)$  طبق رابطه ۱۵ می‌تواند بدست آید [۲۰].

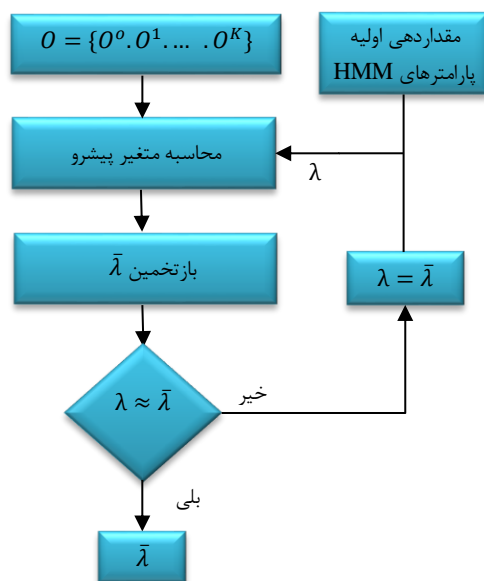
$$P(O|\lambda) = \sum_{i=0}^{N-1} \alpha_T(i) \quad (15)$$

ویژگی جهت حرکت قلم برای بدنه اصلی در هر دو جهت از ابتدا به انتها و بر عکس استخراج شده و به همراه سایر ویژگی‌ها، دو دنباله مشاهدات برای آن حرف ایجاد خواهد شد. کلاسی که دارای احتمال بیشتری از بین این دو حالت باشد، به عنوان حرف خروجی در نظر گرفته می‌شود.

به عنوان مثال در شکل ۱۸- الف، نویسنده حرف «ه» را در جهت مشخص شده نوشته است و سیستم حرف «ع» را با احتمال ۲۳/۱۴۷۴۹- تشخیص داده است. در شکل ۱۸- ب، سیستم با فرض جهت حرکت قلم از انتها به ابتدا حرف «ه» را با احتمال ۱۹/۴۵۵۹- تشخیص داده است. بنابراین حرف دارای احتمال بزرگتر (حرف «ه») به عنوان خروجی نهایی برگردانده می‌شود.

#### ۲-۵-۲ اصلاح میان «ک» و «گ»

طبق بررسی‌های به عمل آمده از نتایج فرآیند بازشناسی مشخص شد که سیستم در تشخیص حرف‌های «ک» و «گ» دچار بیشترین خطا بوده است.



شکل ۱۶- تخمین پارامترهای مدل مخفی مارکوف

اگر دو شرط بالا همزمان برقرار باشند، یعنی اگر اختلاف احتمال لگاریتمی مجموعه دنباله مشاهده شده برای کلاس  $Z$  با پارامترهای  $\lambda$  و  $\bar{\lambda}$  بزرگتر از  $\delta$  باشد و تعداد دفعات تکرار عملیات باز تخمین کمتر از  $\tau$  باشد، فرآیند باز تخمین طبق روابط ۱۶ تا ۲۱ با قرار دادن  $\lambda = \bar{\lambda}$  ادامه خواهد یافت در غیر این صورت  $\bar{\lambda} \approx \lambda$  بوده و فرآیند باز تخمین پایان خواهد یافت.

در روش پیشنهادی  $\delta = 0.05$  و  $\tau = 50$  انتخاب شده است. شکل ۱۶ روال کلی استفاده شده به منظور تخمین پارامترهای HMM را نشان می‌دهد. به این ترتیب برای تمام کلاس‌ها، مدل باز تخمین شده محاسبه می‌شود. در بخش بعد نحوه آزمایش داده‌ها بیان می‌شوند.

#### ۲-۴ بازشناسی

با توجه به اینکه در بخش قبل پارامترهای مدل هر کلاس تعیین شد، اگر تعداد کلاس‌ها  $V$  باشد، در حال حاضر  $V$  مدل در دسترس است، یعنی  $\Lambda = \{\lambda^0, \lambda^1, \dots, \lambda^V\}$ . تعیین کلاس داده آزمایش بدین گونه خواهد بود که ابتدا به کمک مراحل بیان شده در بخش‌های قبل، ویژگی یا دنباله مشاهدات برای داده آزمایش استخراج می‌شود ( $O$ )، سپس به کمک رابطه (۱۵) احتمال  $O$  برای هر کلاس با توجه به مدل آن،  $\lambda^v$  محاسبه می‌شود. در نهایت هر کدام از مدل‌ها که احتمال بیشتری ایجاد نماید، به عنوان کلاس خروجی داده آزمایشی در نظر گرفته خواهد شد. شکل ۱۷ نحوه انجام کار را نشان می‌دهد.

#### ۲-۵ پس پردازش

با توجه به شباهت‌های بسیار زیاد برخی حروف با هم و تنوع نوشتاری بسیار زیاد، بهتر است پس از باز شناسی اولیه، بر اساس برخی خصوصیات حروف مشابه و همچنین برخی خصوصیات زبان نوشتاری فارسی در رسم الخط‌های مختلف، یک سری عملیات پس پردازش به منظور تصمیم‌گیری نهایی انجام شود. در ادامه هر یک از پس پردازش‌های انجام شده در روش پیشنهادی توضیح داده می‌شوند.

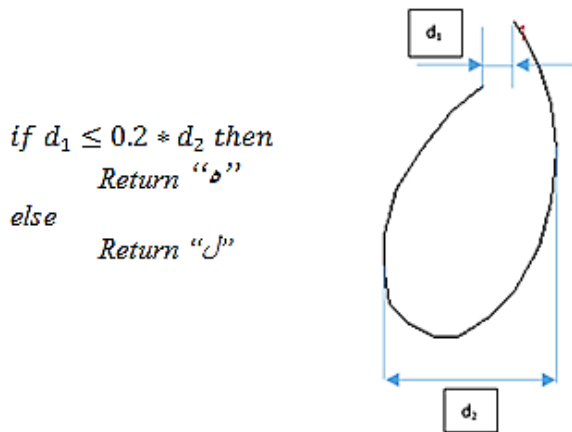
#### ۲-۵-۱ اصلاح جهت حرکت قلم

با توجه به اینکه ویژگی عمده استخراج شده در روش پیشنهادی جهت حرکت قلم می‌باشد، حال اگر اکثر نمونه‌های نوشتار در داده‌های آموزشی برای یک حرف در جهت خاصی باشد، اگر نویسنده، حرف را در خلاف جهت معمول آن بنویسد، احتمال اشتباه در تشخیص آن حرف برای سیستم وجود دارد. برای اصلاح این اشتباه در مرحله بازشناسی برای هر حرف از مجموعه آزمایش،

در غیر این صورت، حرف «ک» به عنوان حرف تشخیص داده شده برگردانده می‌شود. همچنین اگر حرف دارای سه حرکت قلم باشد و سیستم حرف را «ک» تشخیص داده باشد، فرایند پس‌پردازش، حرف «گ» را برمی‌گرداند.

### ۳-۵-۲ تشخیص میان «ه» و «ل»

این دو حرف در برخی نوشتار بسیار به هم شبیه نوشته می‌شوند. جهت تمایز، اگر فاصله مختصات افقی ابتدا و انتهای حرف،  $d_1$  کمتر یا مساوی ۲۰ درصد فاصله حداکثر و حداقل مختصات افقی بدنه،  $d_2$  باشد، حرف «ه» و در غیر این صورت حرف «ل» برگردانده می‌شود. شکل ۱۹ این موضوع را نشان می‌دهد. این فرایند فقط برای حرف‌هایی که در بازشناسی اولیه، حرف «ل» تشخیص داده شده‌اند به کار می‌رود.



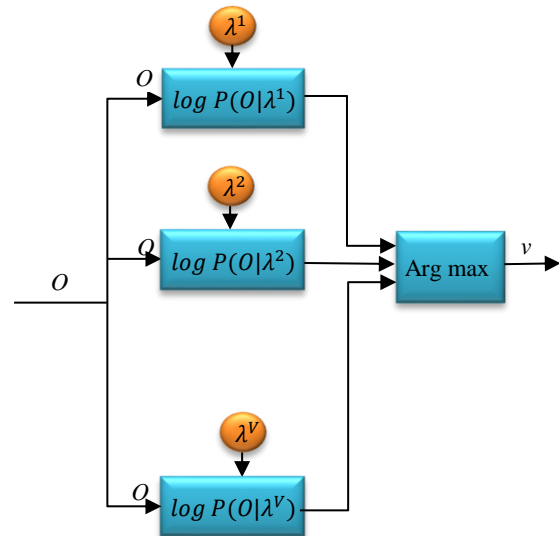
شکل ۱۹ - نحوه تمایز میان «ه» و «ل»

### ۴-۵-۲ تشخیص میان حروف «ت»، «ث» و «ک»، «گ»

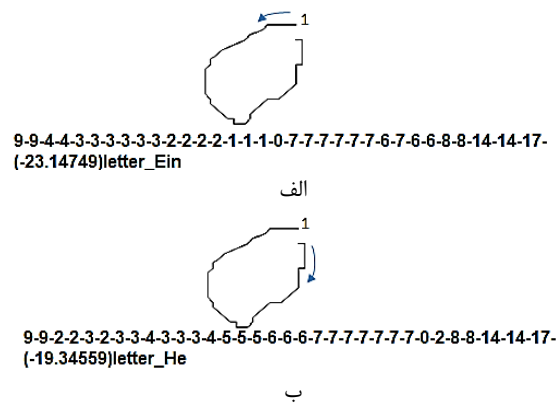
سیستم بازشناسی در برخی از حالت‌هایی که دو نقطه حرف «ت» و «ث» شبیه سرکش حرف «ک» و «گ» بوده، دچار خطا شده است. جهت افزایش نرخ بازشناسی میان این حروف، میانگین مختصات افقی آنها،  $x_m$ ، برای ریز حرکت با حداکثر مختصات افقی،  $x_{max}$  بدنه اصلی مقایسه می‌شود. اگر  $x_m > x_{max}$  و خروجی سیستم حرف «ت» باشد، پس از پس‌پردازش، حرف «ک» و اگر خروجی اولیه، حرف «ث» باشد، خروجی نهایی، حرف «گ» خواهد بود.

### ۵-۵-۲ اصلاح خطاهای ناشی از نقاط حروف

بدلیل اینکه در روش پیشنهادی نقطه‌ها توسط پنجره آستانه تعیین شده‌اند و از طرفی تنوع در نوشتن نقطه زیاد است، سیستم بازشناسی در برخی حالات دچار اشتباه شده است. به عنوان مثال برای حرف «ث» زمانی که سه نقطه آن با یک حرکت نوشته شود و کوچکتر از حد آستانه باشد، سیستم حرف «ب» را برگردانده است



شکل ۱۷ - نحوه تعیین کلاس دنباله مشاهده شده O



شکل ۱۸ - بررسی جهت نوشتن حرف «ه» - در الف با شروع از محل ۱ حرف «ع» تشخیص داده شده در ب با شروع در جهت عکس الف حرف «ه» تشخیص داده شده است.

علت آن هم، تنوع در نوشتن سرکش حرف «گ» است. برای کاهش خطا و انتخاب میان دو حرف، بدین شیوه عمل می‌شود که ابتدا قلاب‌های موجود در ابتدا و انتهای سرکش طبق فرایند بیان شده در بخش ۲-۱-۴ حذف می‌شوند. به جز یک تغییر کوچک در فرایند حذف قلاب‌ها از این قرار که طول پیکسل‌های ابتدایی و انتهایی تا تغییر زاویه مجاز، کمی بیشتر از مقدار آن در حذف قلاب از بدنه اصلی در نظر گرفته شده است. در اینجا حداکثر طول مجاز قلاب ابتدایی ۱۰٪ و قلاب انتهایی ۲۰٪ از طول کل سرکش در نظر گرفته شده است. سپس به کمک الگوریتم داگلاس-پوکر تعداد پیکسل‌های سرکش کاهش داده می‌شوند. اگر تعداد پیکسل‌های بدست آمده بیشتر از دو تا باشد و پیکسل‌های سرکش در حین حرکت دچار تغییر زاویه شدیدی (حداکثر ۹۰ درجه) شود، حرف «گ» برگردانده می‌شود.

جدول ۳ نتایج به ازای  $\epsilon = 1 \times 10^{-5}$  و  $\Delta = 1 \times 10^{-4}$  برای داده‌های آزمایش نشان داده شده است. در این جدول، به منظور مقایسه بهتر با سایر روشها، نتایج تشخیص گروه‌های حروف نیز برای روش پیشنهادی بر طبق نحوه گروه‌بندی جدول ۲ آورده شده است. طبق جدول ۳ مشاهده می‌شود افت شدیدی به ازای  $N > 33$  رخ داده است. دلیل این افت شدید این است که در HMM با ساختار چپ به راست، تعداد حالت‌ها نباید بیش از تعداد مشاهدات باشد و حداکثر می‌تواند برابر با تعداد مشاهدات باشد. بنابراین به ازای  $N = 34$  نرخ تشخیص برای حروف متشکل از یک جزء که طول دنباله ویژگی‌های مشاهده شده در آنها ۳۳ است (بعنوان مثال برای حرف «ا») صفر در نظر گرفته شده است. به همین ترتیب به ازای  $N \geq k$  برای حروفی که طول دنباله ویژگی‌های مشاهده شده در آنها کوچکتر از  $k$  باشد، نرخ تشخیص صفر خواهد بود.

جدول ۲- گروه بندی حروف بر اساس مراجع [۵،۷،۸،۹،۱۱]

گروه ۱	ا، آ	گروه ۷	ص، ض	گروه ۱۳	ل
گروه ۲	ب، پ، ت، ث	گروه ۸	ط، ظ	گروه ۱۴	م
گروه ۳	ج، چ، ح، خ	گروه ۹	ع، غ	گروه ۱۵	ن
گروه ۴	د، ذ	گروه ۱۰	ف	گروه ۱۶	و
گروه ۵	ر، ز، ژ	گروه ۱۱	ق	گروه ۱۷	ه
گروه ۶	س، ش	گروه ۱۲	ک، گ	گروه ۱۸	ی

و یا برای حرف «خ» زمانی که نقطه آن از پنجره آستانه بزرگتر بوده سیستم حرف «چ» را برگردانده است. این تشخیص اشتباه از طریق تطبیق دادن نقاط حرف شناسایی شده با محل و تعداد نقاط در حالت صحیح حرف، اصلاح شده است.

### ۶-۵-۲ تشخیص «ی» از «س»

همانطور که بیان شد در این مقاله، بدنه اصلی حروف به ۲۶ پیکسل هم فاصله تقسیم شده است. از طرفی گاهی در نوشتن «س» نویسنده دندان‌های آن را بسیار ریز می‌نویسد و سیستم آن را «ی» تشخیص می‌دهد. برای اطمینان بیشتر از درستی نتیجه بازشناسی، هنگامی که سیستم حرف «ی» را تشخیص دهد، بازشناسی دیگری با تعداد پیکسل‌های نمونه برداری شده بیشتر یعنی ۳۵ پیکسل انجام می‌شود و نتیجه آن به عنوان خروجی نهایی در نظر گرفته می‌شود.

### ۳- نتایج آزمایش‌ها

به منظور ارزیابی روش پیشنهادی، مجموعه داده TMU-Online [۱-۲] در آزمایش‌ها مورد استفاده قرار گرفته است که ۷۰ درصد آن بعنوان مجموعه آموزش و ۳۰ درصد باقی مانده از آن جهت آزمایش بکار رفته است. در جدول ۲ نحوه گروه‌بندی حروف بر اساس بدنه که در مراجع [۵،۷،۸،۹،۱۱] انجام شده است، آمده است. در

جدول ۳- نرخ بازشناسی با پس‌پردازش برای داده آزمایشی و به ازای  $\epsilon = 1 \times 10^{-5}$  و  $\Delta = 1 \times 10^{-4}$

حالت‌ها	نرخ تشخیص گروه (%)	نرخ تشخیص حرف (%)	حالت‌ها	نرخ تشخیص گروه (%)	نرخ تشخیص حرف (%)
۳	۹۲،۷۸۹	۹۱،۱۵۱	۲۱	۹۸،۵۷۶	۹۸،۰۷۵
۴	۹۳،۵۲۶	۹۱،۹۱۱	۲۲	۹۷،۷۳۵	۹۷،۳۲۴
۵	۹۳،۸۰۹	۹۲،۲۴۶	۲۳	۹۷،۸۵۶	۹۷،۳۲۶
۶	۹۵،۳۶۳	۹۲،۷۳۳	۲۴	۹۷،۶۶۴	۹۶،۶۵۸
۷	۹۴،۷۰۳	۹۳،۱۴۴	۲۵	۹۷،۰۹۹	۹۶،۱۵۹
۸	۹۴،۹۳۹	۹۳،۱۶۰	۲۶	۹۶،۸۱۸	۹۵،۸۲۹
۹	۹۵،۵۷۸	۹۴،۴۱۲	۲۷	۹۶،۹۰۸	۹۶،۷۴۶
۱۰	۹۵،۵۹۸	۹۳،۹۸۹	۲۸	۹۷،۰۶۱	۹۶،۸۳۳
۱۱	۹۶،۲۰۲	۹۴،۲۳۴	۲۹	۹۶،۶۱۱	۹۶،۳۳۲
۱۲	۹۵،۹۹۸	۹۴،۶۶۷	۳۰	۹۶،۱۶۴	۹۵،۹۲۰
۱۳	۹۶،۳۷۰	۹۵،۰۸۱	۳۱	۹۷،۷۷۷	۹۷،۱۶۰
۱۴	۹۷،۳۰۱	۹۵،۹۸۴	۳۲	۹۷،۲۵۲	۹۶،۴۹۶
۱۵	۹۷،۱۳۴	۹۲،۹۵۴	۳۳	۹۷،۸۵۳	۹۶،۹۹۲
۱۶	۹۷،۳۸۱	۹۵،۸۲۰	۳۴	۶۵،۴۸۱	۵۸،۶۶۹
۱۷	۹۷،۲۵۳	۹۵،۶۵۴	۳۵	۶۵،۳۸۲	۵۸،۰۹۶
۱۸	۹۷،۸۴۱	۹۷،۱۶۰	۳۶	۶۵،۴۸۱	۵۸،۱۸۰
۱۹	۹۸،۱۸۲	۹۷،۲۳۵	۳۷	۶۵،۵۳۱	۵۷،۹۳۷
۲۰	۹۸،۳۶۱	۹۷،۷۴۵	۳۸	۶۵،۵۳۰	۵۷،۹۳۷

جدول ۴ - مقایسه روش پیشنهادی با سایر روشها در مراجع بیان شده

مرجع	روش	دقت بازشناسی گروه (%)	دقت بازشناسی حروف (%)	شرایط آموزش و آزمون پایگاه داده
[۳]	ترکیب متامورفیزم و DTW	---	۹۵/۱	آموزش ۷۰ نفر اول و بقیه برای آزمون از ۱۲۴ نفر
[۴]	مدل مخفی مارکوف پیوسته	۹۲/۸۲	۸۸/۸۷	آموزش ۷۰ نفر اول و بقیه برای آزمون از ۱۲۴ نفر
[۵]	SVM	۹۷/۹۷	۹۷/۲۹	میانگین ۱۰ بار اجرای تصادفی با آموزش ۷۰ درصد و ۳۰ درصد برای آزمون از ۱۲۴ نفر
[۶]	درخت تصمیم	۹۴/۲	۹۲/۳	۶۰ نمونه مجموعه و ۵۷ نمونه برای آزمایش از ۱۱۷ نفر
[۷]	SVM	۹۴/۳	۹۸	آموزش ۷۰ درصد و ۳۰ درصد برای آزمون از ۱۲۴ نفر
[۸]	مدل مخفی مارکوف	۹۳	۹۶	آموزش ۷۰ درصد و ۳۰ درصد برای آزمون از ۱۲۴ نفر
[۹]	مدل مخفی مارکوف پیوسته	۹۵/۹	۹۴/۲	۶۰ نمونه (۵۱٪) مجموعه و ۵۷ نمونه برای آزمایش از ۱۱۷ نفر
[۱۰]	شبکه عصبی و طبقه بندی کمترین فاصله	---	۹۳/۳	۴۰ نمونه برای آموزش اجزای کوچک به کمک شبکه عصبی، همه نمونهها برای آموزش و آزمایش بدنه اصلی
روش پیشنهادی	مدل مخفی مارکوف گسسته و روش پیشنهادی برای استخراج دنباله مشاهدات	۹۸/۵۸	۹۸/۰۸	آموزش ۷۰ درصد و ۳۰ درصد بقیه برای آزمون

آموزش و تخمین مدلها بخصوص در  $N=21$  بهتر از سایر مقادیر  $\Delta$  انجام می شود.

نتایج نمودار ۳ نشان می دهد نرمال سازی حروف به ازای  $d=1$  (رابطه ۳) بهتر انجام شده است. افزایش بیش از حد مقدار  $d$  بدلیل نامطلوب شدن فرایند نرمال سازی، باعث کاهش دقت بازشناسی حتی کمتر از حالت  $d=0$  شده است.

در نمودار ۴ مشاهده می شود که تکرار کدهای ۸، ۹، ۱۴، ۱۵ و ۱۶ در دنباله مشاهدات توانسته است نتایج بهتری را نسبت به حالت بدون تکرار ارائه نماید. در  $N=21$  بیشترین دقت به ازای وضعیتی رخ داده است که کدهای مورد نظر در دنباله مشاهدات یک بار تکرار شده اند.

به طور کلی، کوتاه ترین دنباله مشاهدات متعلق به حروفی است که نقطه یا ریز حرکت نداشته باشند. بنابراین برای این حروف به ازای وضعیت بدون تکرار کدها، دنباله مشاهدات شامل ۳۰ کد خواهد شد. اگر یک بار تکرار را برای کدها در نظر بگیریم، برای حروف ساده که نقطه یا ریز حرکت ندارند، تنها کدهای ۸، ۹ و ۱۴ تکرار خواهند شد. در نتیجه حداقل طول دنباله مشاهدات برابر با ۳۳ می شود. به همین ترتیب برای دو بار تکرار کدها در دنباله مشاهدات، حداقل طول ۳۶ کد را خواهیم داشت. به همین دلیل در نمودار ۴ مشاهده می شود که افت ناگهانی در دقت بازشناسی به ازای سه وضعیت بدون تکرار کد، یک بار تکرار و دو بار تکرار در دنباله مشاهدات به ترتیب در  $N=31$ ،  $N=34$  و  $N=37$  رخ داده است.

با توجه به جدول ۳، مشاهده می شود بیشترین نرخ بازشناسی برای داده های آزمایشی در حالت  $N=21$  بیش از ۹۸٪ و برابر با ۹۸/۰۷۵ بدست آمده است که نسبت به سایر روشها دقت بالاتری ارائه کرده است. به منظور مقایسه بهتر، خلاصه نتایج بدست آمده در سایر روشها که در بخش مقدمه بیان شده است به همراه نتیجه روش پیشنهادی در جدول ۴ قابل مشاهده است. در جدول ۵ حداکثر نرخ بازشناسی مربوط به  $Top-1$ ،  $Top-2$ ،  $Top-3$  و  $Top-5$  برای بازشناسی داده های آزمایشی بدون انجام پس پردازش آمده است. مشاهده می شود که انجام اصلاحات ذکر شده در فرایند پس پردازش توانسته است درصد بازشناسی را از ۹۷/۰۵ به ۹۸/۰۷۵ افزایش دهد. میزان تاثیر پارامترهای  $\epsilon$ ،  $\Delta$  و  $d$  و همچنین تاثیر تکرار کدها در دنباله مشاهدات در نرخ بازشناسی (بدون پس پردازش) به ترتیب در نمودارهای ۱ تا ۴ آورده شده است. در نمودار ۱ مشاهده می شود نتایج به ازای  $\epsilon = 1e-5$  و  $\epsilon = 0.1$  بطور کلی از نتایج بدست آمده به ازای  $\epsilon = 0$  بهتر بوده است. در تعداد حالتها برابر با ۲۱، بیشترین دقت بازشناسی به ازای هر سه مقدار  $\epsilon$  به دست آمده است که بیشترین دقت مربوط به  $\epsilon = 1e-5$  بوده است. بنابراین بهتر است در ساختارهای مشابه برای HMM گسسته با حالت پایانی PTO، مقدار  $\epsilon$  یک مقدار بزرگتر از صفر و نزدیک به صفر انتخاب شود.

تاثیر مقادیر مختلف  $\Delta$  (حداقل احتمال مشاهده هر حرف در رابطه ۲) در نمودار ۲ نشان می دهد به ازای  $\Delta = 1e-4$ ، فرایند

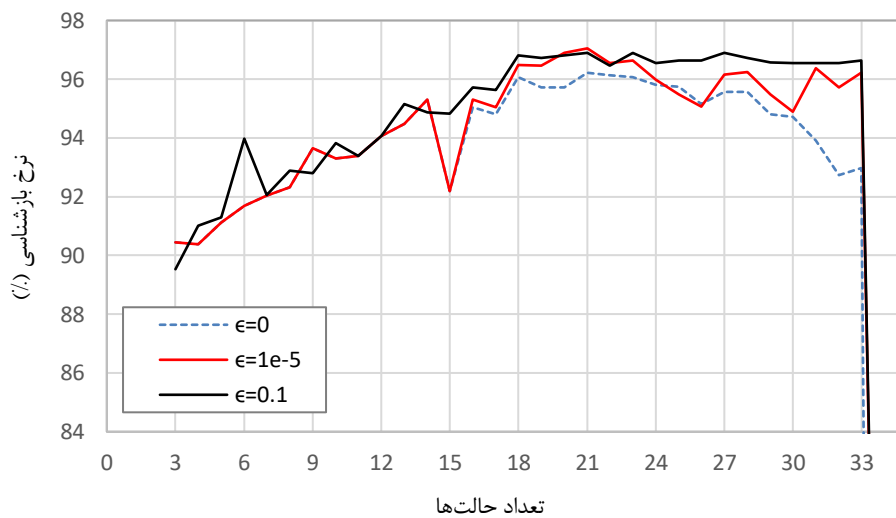


پیکسل‌ها در مرحله یکسان سازی تعداد و فاصله پیکسل‌ها برابر با ۲۶ به دست آمد.

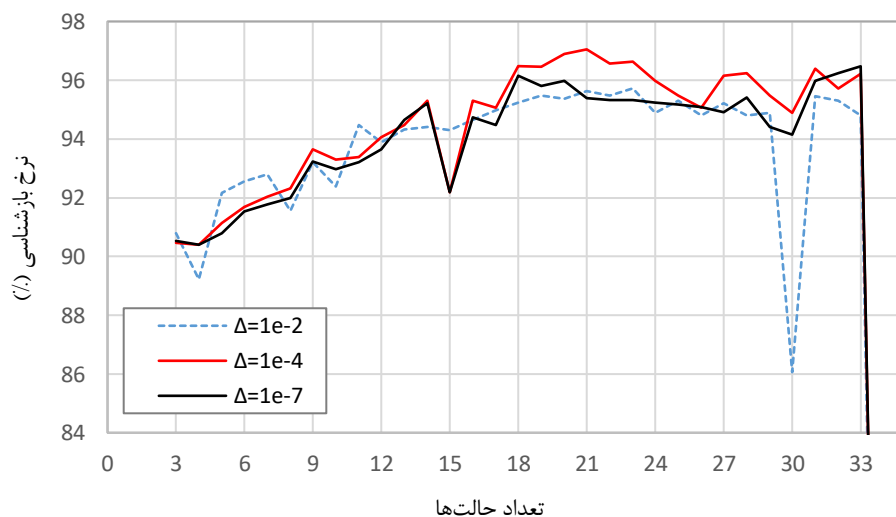
جدول ۵- حداکثر نرخ بازشناسی سطح‌های مختلف برای مجموعه آزمایش به ازای  $\epsilon = 1 \times 10^{-5}$  و  $\Delta = 1 \times 10^{-4}$

سطح Top	Top-۵	Top-۳	Top-۲	Top-۱
حالت (N)	۳۲	۳۱	۲۱	۲۱
حداکثر نرخ تشخیص حرف (%)	۹۹/۴۹	۹۹/۰۸	۹۸/۴۱	۹۷/۰۵

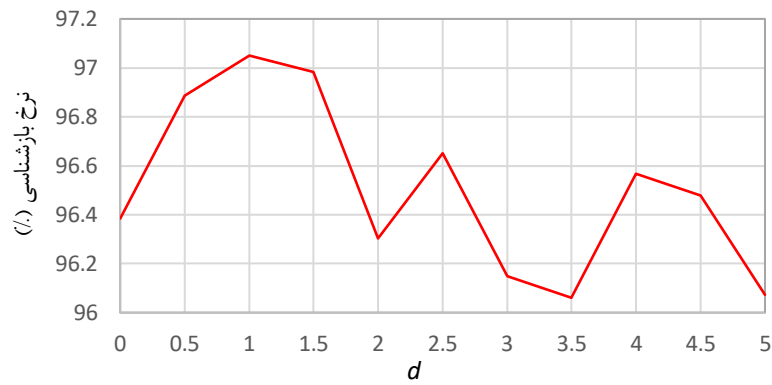
میزان تاثیر مقادير مختلف  $T$  (تعداد نهایی پیکسل‌های بدنه اصلی پس از یکسان سازی تعداد و فاصله پیکسل‌ها) بر روی نرخ بازشناسی نهایی داده‌های آزمایشی (بدون پس پردازش) در نمودار ۵ نشان داده شده است. نتایج این نمودار نشان می‌دهد اگر تعداد پیکسل‌های حروف پس از یکسان سازی، خیلی کم یا خیلی زیاد در نظر گرفته شود، دقت بازشناسی کاهش خواهد یافت. به ازای تعداد پیکسل‌های خیلی کم، ساختار و جزئیات مهم حروف به خوبی قابل توصیف نخواهند بود. از طرف دیگر اگر تعداد پیکسل‌ها خیلی زیاد در نظر گرفته شود، باعث افزایش بیش از حد تنوع نوشتار و باقی ماندن جزئیات غیر ضروری و نامطلوب می‌شود. بهترین مقدار برای تعداد



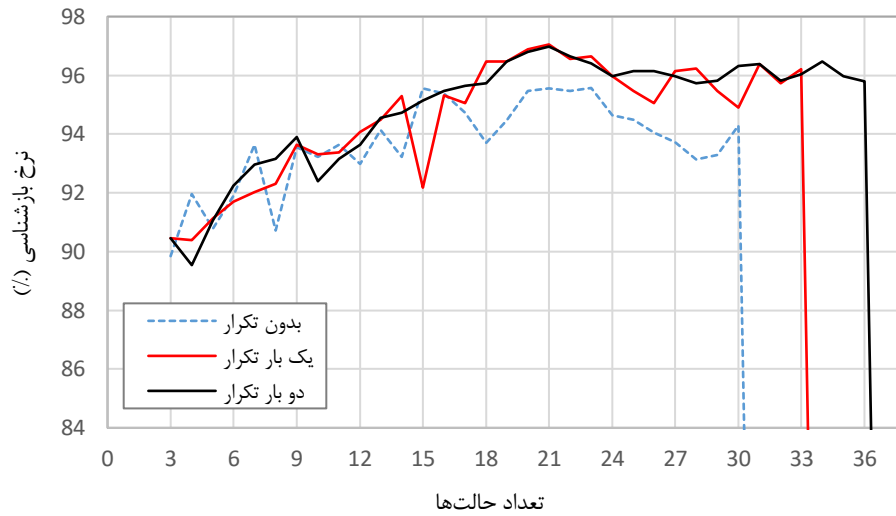
نمودار ۱- تاثیر مقادير مختلف  $\epsilon$  در نرخ بازشناسی داده آزمایشی، بدون پس پردازش و به ازای  $\Delta = 1 \times 10^{-4}$  و Top-۱



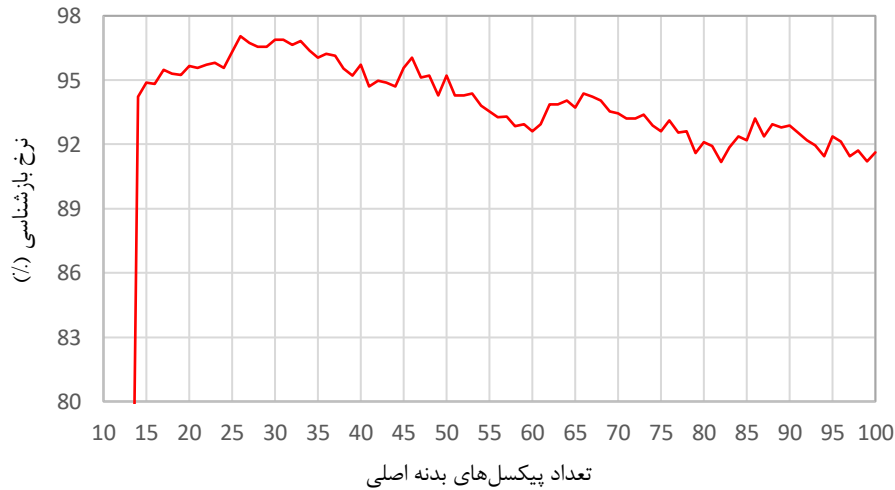
نمودار ۲- تاثیر مقادير مختلف  $\Delta$  در نرخ بازشناسی داده آزمایشی، بدون پس پردازش و به ازای  $\epsilon = 1 \times 10^{-5}$  و Top-۱



نمودار ۳- تأثیر مقادیر مختلف  $d$  در رابطه (۳) در نرخ بازشناسی داده آزمایشی بدون پس پردازش و به ازای  $\epsilon = 1 \times 10^{-5}$  و  $\Delta = 1 \times 10^{-4}$  و  $N = 21$  و Top-1



نمودار ۴- تأثیر تکرار کدهای برداشتن/گذاشتن قلم و کدهای مربوط به تک نقطه (بدون نقطه، نقطه بالا و نقطه پایین) در نرخ بازشناسی داده آزمایشی، بدون پس پردازش و به ازای  $\epsilon = 1 \times 10^{-5}$  و  $\Delta = 1 \times 10^{-4}$  و Top-1



نمودار ۵- نرخ بازشناسی حروف به ازای تغییرات تعداد پیکسل‌های بدنه اصلی برای داده آزمایشی بدون پس پردازش و به ازای  $\epsilon = 1 \times 10^{-5}$  و  $\Delta = 1 \times 10^{-4}$  و  $N = 21$  و Top-1

#### ۴- نتیجه گیری

نوشتاری فارسی از جمله جهت حرکت قلم و ویژگی‌های نقطه و سرکش و تعداد و محل آنها به بازشناسی حروف مجزای فارسی اقدام شده است. نوآوری بارز در روش پیشنهادی استفاده از ویژگی‌های

در روش پیشنهادی با استفاده از مدل مخفی مارکوف گسسته و استفاده از ویژگی‌های پیشنهادی مبتنی بر خصوصیات زبان

مخفی مارکوف، "پانزدهمین کنفرانس بین المللی سالانه انجمن کامپیوتر ایران، تهران، انجمن کامپیوتر، مرکز توسعه فناوری نیرو، ۱۳۸۸.  
 [9] V. Ghods, M. K. Sohrabi, "Online Farsi Handwritten Character Recognition Using Hidden Markov Models", Journal of computers, vol. 11, no. 2, pp. 169-175, March 2016.  
 [۱۰] سیدمحمد رضوی و احسان اله کبیر، "روشی ساده برای بازشناسی برخط حروف مجزای فارسی"، ششمین کنفرانس سراسری سیستم های هوشمند، کرمان، دانشگاه شهید باهنر کرمان، ۱۳۸۳  
 [۱۱] اعظم شکاری شهرک و مریم خادمی، "بازشناسی دست نوشته فارسی با مدل مخفی مارکوف"، همایش ملی مهندسی کامپیوتر و توسعه پایدار با محوریت شبکه های کامپیوتری، مدل سازی و امنیت سیستمها، مشهد، موسسه آموزش عالی خاوران، ۱۳۹۲  
 [۱۲] مسعود فرکی و مازیار پالهنک، "بازشناسی برخط حروف فارسی بر پایه مدل مخفی مارکوف"، فصلنامه مهندسی برق دانشگاه تبریز، جلد ۴۰، شماره ۱، ۱۳۸۹

[13] B.Q Huang and Y.B Zhang and M-T. Kechadi, "Preprocessing Techniques for Online Handwritten Recognition", Seventh International Conference on Intelligent Systems Design and Applications (ISDA), pp. 793-798, 20-24 Oct. 2007.  
 [14] Y. Zhang, G. Shi, J. Yang, "HMM-based Online Recognition of Handwritten Chemical Symbols", 10<sup>th</sup> International Conference on Document Analysis and Recognition, pp. 1255-1259, 2009  
 [15] V. Kumar, "Online Handwritten Recognition Problem: Issues and Techniques", MIT international Journal of Computer Science & Information Technology, Vol. 4, No. 1, pp. 16-24, Jan. 2014.  
 [16] M. Pastor, A. Toselli, E. Vidal, "Writing speed Normalization for Online Handwritten Text Recognition", Eight International Conference on Document Analysis and Recognition (ICDAR'05), pp. 1131-1135, Seoul, South Korea, 2005  
 [17] David H. Douglas and Thomas K. Peucker. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, pp. 112-122, October 1973.  
 [18] Lauri Vanhatalo, "Online Sketch Recognition Geometric Shapes", MS Thesis, Aalto University, School Of Science and Technology, Helsinki May 29, 2011  
 [19] Devijver, P. A. and Kittler, J. "Pattern Recognition: A Statistical Approach". Prentice-Hall International, 1982.  
 [20] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in speech recognition, Proc. IEEE, Vol. 77, No. 2, Feb 1989.  
 [21] G. Rigoll, A. Kosmala, J. Rottland, Ch. Neukirchen, " A Comparison between Continuous and Discrete Density Hidden Markov Models for Cursive Handwritten Recognition", In proceedings of the 13<sup>th</sup> International Conference on Pattern Recognition, vol. 2, pp. 205-210, Vienna, Austria, 1996  
 [22] Xiaofang Xic, "On the Recognition of Handwritten Mathematical Symbols", PhD. Thesis, the University of Western Ontario, London, Dec 2007  
 [23] J. Andersson, "Hidden Markov Model based Handwritten Recognition", MS Thesis, Center for Mathematical sciences, Lund Institute of Technology, May 21, 2002

کمتر ولی موثرتر در مرحله استخراج ویژگی و ایجاد دنباله مشاهدات است که به کمک مدل مخفی مارکوف گسسته با ۲۱ حالت (از جمله یک حالت پایانی) نتایج خوبی را بدست داده است. اعمال اصلاحات مفید و مبتنی بر خصوصیات حروف مشابه در زبان فارسی در فرایند پس پردازش پیشنهادی، باعث افزایش حدود یک درصد در نرخ بازشناسی شد. نتایج بدست آمده نرخ بازشناسی بیش از ۹۸ درصد را برای داده های آزمایشی نشان داد که بیش از سایر روش های پیاده سازی شده بر روی این مجموعه داده است. در صورت انجام پس پردازش های بیشتر، می توان انتظار داشت نرخ بازشناسی افزایش بیشتری داشته باشد.

## منابع:

[۱] سید محمود رضوی و احسان اله کبیر، "یک پایگاه داده برای بازشناسی برخط،" ششمین کنفرانس سیستم های هوشمند، ص ۲۱۸ تا ۲۲۵، کرمان، آذر ۱۳۸۳.  
 [2] V. Ghods, E. Kabir, "A STUDY ON FARSI HANDWRITTEN STYLES FOR ON LINE RECOGNITION", Malaysian journal of Computer Science Vol. 26, no. 1, pp. 44-59, 2013  
 [۳] فؤاد جلالی، "بازشناسی نویسه های برخط فارسی بر اساس متامورفیزم شکل"، پایان نامه کارشناسی ارشد، دانشگاه تربیت مدرس، تهران، اسفند ۱۳۸۷  
 [۴] هدیه ساجدی، منصور جم زاد، حسین ثامتی، باقر باباعلی، "ارائه یک روش مبتنی بر گروه بندی برای بازشناسی حروف مجزای برخط فارسی به کمک مدل مخفی مارکوف"، کنفرانس بین المللی انجمن کامپیوتر ایران، دوره دوازدهم، ص ۴۱۹ تا ۴۲۶، دانشگاه شهید بهشتی، تهران، اسفند ۱۳۸۵  
 [۵] مجید مرزانی، سید محمود رضوی، مهران تقی پور گرجی کلایی، "روشی کاربردی برای بازشناسی برخط حروف مجزای دست نویس فارسی با استفاده از همزمان از دانش مربوط به بدنه اصلی و ریزحرکات"، هوش محاسباتی در مهندسی برق، سال ششم، شماره دوم، ص ۸۷ تا ۱۰۰، تابستان ۹۴  
 [۶] وحید قدس، و احسان اله کبیر، "بازشناسی حروف برخط فارسی با استفاده از ویژگی های ساختاری"، روش های هوشمند سازی در صنعت برق، سال سوم، شماره دهم، ص ۶۶ تا ۷۱، تابستان ۹۱  
 [۷] محمدمین مهر علیان و کاظم فولادی، "بازشناسی برخط حروف مجزای دست نویس فارسی براساس تشخیص گروه و بدنه اصلی با استفاده از ماشین بردار پشتیبان"، هفتمین کنفرانس ماشین بینایی و پردازش تصویر، تهران، دانشگاه علم و صنعت، ۱۳۹۰.  
 [۸] محمدمین مهر علیان و کاظم فولادی، "بازشناسی برخط حروف مجزای دست نویس فارسی بر اساس تشخیص گروه اصلی بدنه با استفاده از مدل

## زیر نویس ها:

<sup>6</sup> Vector Quantization(VQ)

<sup>7</sup> Codebook

<sup>8</sup> Process Termination Observation

<sup>9</sup> Ergodic

<sup>10</sup> None-Ergodic

<sup>1</sup> Online

<sup>2</sup> Offline

<sup>3</sup> Dynamic Time Wrapping

<sup>4</sup> Digitizer

<sup>5</sup> Hidden Markov Model(HMM)