

شناسایی کلمات دستنویس نستعلیق فارسی با استفاده از تقطیع

مبتنی بر کانتور بالایی کلمات و مدل پنهان مارکف

رضا صفا بخش

پیمان ادیبی

دانشیار

فارغ التحصیل کارشناسی ارشد

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

چکیده

در این مقاله یک سیستم کامل برای شناسایی کلمات دستنویس فارسی با استفاده از یک مدل پنهان مارکف با چگالی پیوسته مشاهدات و طول متغیر حالات ارائه شده است. ابتدا در مرحله پیش پردازش پس از عملیات باینری کردن و حذف نویز و یافتن اجزاء متصل، سرکشها و زیرکشهای حروف کشف و از تصویر حذف می گردند. سپس یک الگوریتم تقطیع جدید مبتنی بر تحلیل کانتور بالایی اجرا می شود. در این الگوریتم هدف این است که حتی الامکان مشکل عدم تقطیع وجود نداشته باشد. طول حالات متغیر برای برطرف کردن مسأله تقطیع زیادی در نظر گرفته شده است. با یافتن ترتیب راست به چپ، این قطعه ها یا زیر حروف بعنوان دنباله سمبلهای مدل شونده محسوب می گردند. یک بردار ویژگی تلفیقی هشت عنصری که شامل سه توصیفگر فوریه و تعدادی ویژگی ساختاری می باشند برای نمایش این سمبلها در فضای ویژگیها بکار می رود. آموزش مدل به سادگی و بدون نیاز به روش تخمین مجدد انجام می گیرد. در این آموزش تعدادی از پارامترهای مدل از مجموعه تصاویر آموزشی و تعدادی دیگر از فرهنگ لغات بدست می آیند. در نهایت یک نسخه الگوریتم ویتربی اصلاح شده که قادر به تولید L بهترین مسیر سراسری بوده و می تواند برای هر حالت موقعیتها و شکلهای متفاوت را بعنوان زیر حالات در نظر گیرد و همچنین طول حالات متغیر را پشتیبانی نماید، جهت شناسایی بکار رفته است. آزمایشهایی که بر روی نمونه های دستنویس با کیفیت متوسط و یک فرهنگ لغات ۵۰ کلمه ای انجام شد نتایج بسیار خوبی برای روش بکار رفته نشان داد.

کلمات کلیدی

شناسایی کلمات، دستنویس، فارسی، نستعلیق، مدل پنهان مارکف، تقطیع، استخراج ویژگی.

Farsi Nastaaligh Handwritten Word Recognition Using Upper Contour Based Segmentation and Hidden Markov Model

R. Safabakhsh
Associate Professor

P. Adibi
Graduate Student

Computer Engineering Department,
Amirkabir University of Technology

Abstract

Automatic recognition of handwritten text has been an active research field for many years. Recognition of Farsi handwritten scripts is a very difficult task because Farsi scripts are cursive. This paper introduces the design, implementation, and evaluation of a system for recognition of Farsi Nastaaligh handwritten words. First, several preprocessing operations are applied to the

picture of the word, including binarization, noise reduction, connected-component specification, secondary stroke elimination, and ascender/descender elimination. In the second stage, a modified segmentation algorithm based on upper contour local minima is applied to the preprocessed image. The result of this stage is sub-character images which are input into a loop of feature extraction, recognition, and adjacent-segment combination operations to yield several of the best recognition results. In the feature extraction stage, eight features, including three Fourier descriptors and five structural features, are computed. This is a new feature vector with high discrimination power and invariance to scale and shift. A continuous-density variable-duration hidden Markov model is applied in the recognition stage. This is a path-discriminant HMM method in which the states are considered as pure characters (without secondary strokes) in addition to some compound forms of characters in Nastaaligh handwriting style. In the training stage, some parameters of the model are obtained from the training image set and the others from the dictionary. A modified version of Viterbi algorithm yields the final results of recognition. This algorithm considers different positions and forms of letters in Farsi words as sub-states and also supports variable duration states to compensate for the over-segmentation problem in the segmentation stage. Experiments on handwritten samples and a 50-word dictionary show very good performance of the system.

Keywords

OCR, Handwritten, Farsi, Cursive, HMM, Segmentation, Feature Extraction

مقدمه

با توجه به کاربردهای متعدد شناسایی برون خط متون دستنویس، مانند پردازش چکهای بانکی، شناسایی آدرس و کدهای پستی، حذف عملیات تایپ در بعضی از کاربردهای ورود اطلاعات و غیره، تحقیقات در این زمینه در حال گسترش است. اما با وجود پیشرفتهای بدست آمده، هنوز قدرت این سیستمها در خواندن حروف با توانایی بشر فاصله بسیار دارد. در چند دهه اخیر کارهای زیادی برای شناسایی حروف زبانهای لاتین، چینی، ژاپنی و کره‌ای انجام شده است. اما برای شناسایی حروف فارسی و عربی، با اینکه بیش از یک سوم از جمعیت جهان برای نوشتن از این حروف استفاده می‌کنند، کارهای انجام شده نسبتاً کم و پراکنده بوده است. از جمله علل این امر، عدم سرمایه‌گذاری کافی، فقدان پایگاه داده‌ها و لغتنامه‌های استاندارد و جامع برای متون فارسی و عربی، و همچنین خصلت پیوسته بودن این خط ذکر شده است [۱]. خلاصه‌ جامعی از کارهای انجام شده در شناسایی متون عربی در مرجع [۱] ارائه شده است.

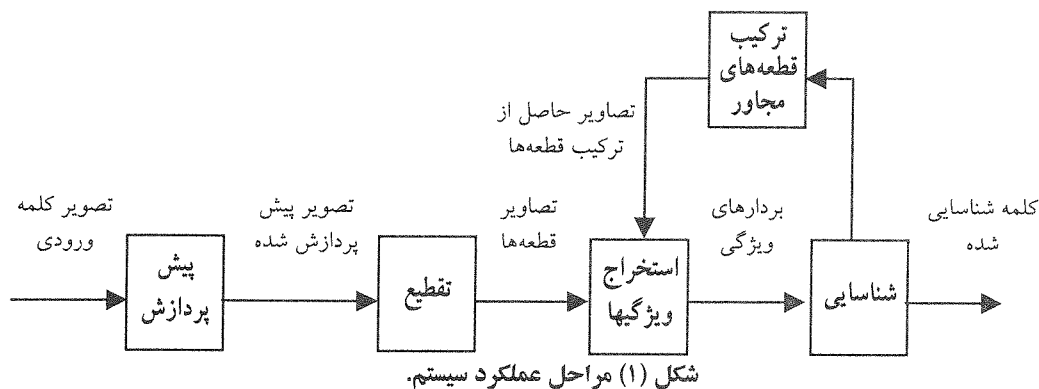
یک پارامتر مهم برای دسته‌بندی سیستمهای شناسایی حروف، وجود و نحوه عملکرد مرحله تقطیع در آنها می‌باشد. در [۲] مفهوم تقطیع و روشهای مختلف آن بررسی شده و سه استراتژی اصلی برای تقطیع مطرح شده است، بطوریکه هر روش تقطیع می‌تواند بصورت ترکیب وزن‌داری از این سه استراتژی کلی محسوب گردد. این استراتژیها عبارتند از: (۱) روش کلاسیک، که در واقع هدف آن تلاش برای تجزیه تصویر به واحدهای قابل دسته‌بندی می‌باشد، (۲) تقطیع مبتنی بر شناسایی، که در آن سیستم در تصویر بدنبال اجزایی می‌گردد که با دسته‌های موجود در الفبای آن تطبیق حاصل کنند و تصمیم‌گیری راجع به تقطیع با کمک یک فیدبک از مرحله شناسایی انجام می‌شود، و (۳) روشهای کلی، که در آنها سیستم بدنبال شناسایی کلمات بعنوان یک واحد می‌باشد.

روشهای مبتنی بر استراتژی کلی دارای این مزیت هستند که مرحله مشکل تجزیه به حروف یا شبه حروف در آنها وجود ندارد. اما ایراد این سیستمها این است که تعداد لغات قابل شناسایی توسط آنها زیاد نیست و به یک فرهنگ لغات معمولاً کوچک محدود می‌باشد. در صورتیکه روشهای کلاسیک و روشهای مبتنی بر شناسایی، از آنجا که در آنها حروف یا شبه حروف بعنوان دسته‌های قابل شناسایی مطرح می‌باشند، از نظر تعداد کلمات قابل شناسایی، بسیار قویتر هستند.

در مقاله حاضر، سیستمی برای شناسایی برون خط کلمات دستنویس فارسی ارائه شده است. مراحل عملکرد این سیستم در شکل ۱ نشان داده شده است. تصویر کلمه ابتدا تحت یک سری پیش پردازش قرار می‌گیرد. سپس در مرحله تقطیع، کلمه به حروف یا زیرحروف تشکیل دهنده آن شکسته می‌شود. پس از آن یک مجموعه ویژگیها از تصویر هر قطعه یا تصویر مرکب از قطعه‌های مجاور استخراج می‌شود تا شناسایی مبتنی بر دسته بندی این بردارهای ویژگی انجام گیرد. در مرحله شناسایی مدل

شناسایی کننده که قبلاً آموزش داده شده است، از روی این بردارهای ویژگی، کلمه را شناسایی می‌نماید. از آنجا که هر قطعه و ترکیبهای قطعه‌های مجاور برای یافتن الگوهای قابل شناسایی که بطور سراسری بهینه هستند مورد بررسی قرار می‌گیرند، می‌توان تقطیع در این روش را در دسته روشهای تقطیع مبتنی بر شناسایی جای داد. با توجه به کاربردهای موفقی که مدل‌های پنهان مارکف (HMM)^۱ در شناسایی حروف داشته‌اند [۳] [۴] [۵] [۶]، یک روش مبتنی بر مدل پنهان مارکف برای مرحله شناسایی انتخاب شده است. مدل مورد استفاده یک مدل پنهان مارکف با طول حالات متغیر و چگالی مشاهدات پیوسته می‌باشد [۴]. چرخه "شناسایی-ترکیب قطعات مجاور-استخراج ویژگیها" که توسط الگوریتم ویتربی بکار رفته برای مدل پنهان مارکف اجرا می‌گردد کلید یافتن کلمات بهینه در این سیستم می‌باشد، بدین ترتیب که تصاویر قطعات حاصل از مرحله تقطیع که ترتیب راست به چپ آنها نیز مشخص شده است در اختیار الگوریتم ویتربی اصلاح شده قرار می‌گیرند. این الگوریتم که از جمله روشهای برنامه ریزی پویا محسوب می‌شود تعداد زیادی از انتخابهای ممکن برای ترکیب قطعه‌های مجاور را بررسی می‌کند و برای هر انتخاب یک امتیاز کلی محاسبه می‌نماید. نهایتاً انتخابی که بیشترین امتیاز را داشته باشد بعنوان نتیجه شناسایی در خروجی ارائه می‌شود. عملکرد الگوریتم ویتربی اصلاح شده در بخش ۶ به تفصیل توضیح داده شده است. شکل ۱۳ مثالی از عملکرد این الگوریتم و نقش چرخه "شناسایی - ترکیب قطعات مجاور - استخراج ویژگیها" در یافتن طول حروف یک کلمه را نشان می‌دهد.

در بخش ۱، ویژگیهای خط فارسی و عربی به اختصار بیان می‌شود. در بخش ۲، مدل پنهان مارکف و روشهای شناسایی حروف مبتنی بر آن توضیح داده می‌شود. بخش ۳ به بیان پیش پردازشهای مورد استفاده می‌پردازد. در بخش ۴ روش بکار رفته برای تقطیع و یافتن ترتیب راست به چپ مطرح می‌گردد. در بخش ۵، ویژگیهای آزمایش شده و بردار ویژگی انتخابی ارائه می‌شوند و سپس آموزش مدل و مرحله شناسایی در بخش ۶ توصیف می‌گردد. در بخش ۷ نتایج حاصل از آزمایش روشها ارائه می‌شود. نهایتاً بخش ۸ خلاصه‌ای از روشها و نتایج را بیان می‌کند.



۱- ویژگیهای نوشتار فارسی و عربی و پایگاه کلمات فارسی

۱-۱- ویژگیهای نوشتار فارسی و عربی

خط فارسی و عربی از چند جهت با خطهای لاتین تفاوت دارد: (۱) شکل حروف فارسی و عربی تابعی از محل قرار گرفتن آنها در کلمه می‌باشد و لذا برای هر حرف تا چهار شکل مختلف در موقعیتهای مختلف در کلمه ("اول"، "وسط"، "آخر"، و "جدا") می‌تواند وجود داشته باشد (جدول ۱). البته در بعضی از سبکهای نوشتاری، یک حرف در یک موقعیت مشخص نیز می‌تواند بیش از یک شکل داشته باشد. (۲) خط فارسی و عربی ذاتاً بصورت پیوسته نوشته می‌شود. البته بعضی از حروف از سمت چپ به حرف بعدی نمی‌چسبند (حروف: ا، د، ذ، ر، ز، ژ، و) و به همین دلیل یک کلمه می‌تواند شامل بیش از یک بخش پیوسته باشد که به این بخشها، زیرکلمه^۲ گفته می‌شود. (۳) خط فارسی و عربی در حالت چاپی دارای فونتهای بسیار زیاد و در حالت دستنویس دارای سبکهای نوشتاری متعدد می‌باشد. هر سبک نوشتاری نیز می‌تواند شامل شکلها و فرمهای ترکیبی جدیدی از حروف باشد. بدین ترتیب اگر مثلاً یک متن فارسی دستنویس بدون محدودیت را در نظر بگیریم، تعداد الگوهای مجزایی که با آنها سروکار خواهیم داشت بسیار زیاد خواهد بود، که این امر کار شناسایی را بسیار مشکل می‌سازد. (۴) حروف

فارسی و عربی می‌توانند دارای یک تا سه نقطه در بالا یا پایین خود باشند و در بعضی موارد، تنها عامل جدا کننده بین دو حرف وجود و تعداد همین نقطه‌ها است. جدول ۱ مجموعه حروف فارسی و اشکال مختلف هر حرف برای موقعیتهای مختلف را نشان می‌دهد. خط عربی نیز مشابه خط فارسی است، تنها چهار حرف کمتر دارد (این چهار حرف در جدول ۱ بصورت زیر خط دار مشخص شده است).

انواع خطوطی که از الفبای فارسی یا عربی استفاده می‌کنند به هشت گروه کلی که هرگروه خود ممکن است دارای انواع مختلفی باشد تقسیم می‌شوند [۳۱]. برخی از این خطوط در زمانهای گذشته رایج تر بوده و امروزه منسوخ شده‌اند، و برخی در زمان حاضر مورد توجه و استفاده بیشتر قرار دارند. می‌توان گفت که اکثر نوشته‌های امروزی با خطوط نستعلیق و نسخ نوشته می‌شوند و خط نستعلیق بواسطه زیبایی خاص خود امروزه رایج‌ترین و محبوب‌ترین خط در ایران است. به همین دلیل خط مورد استفاده در این پروژه خط نستعلیق تعیین گردید.

جدول (۱) مجموعه حروف فارسی(عربی) و شکل آنها در موقعیتهای مختلف.

N0	Character	Isolated	First	Middle	Last
1	Alef	ا (أ)	ا (أ)	ا	ا
2	Be	ب	ب	ب	ب
3	Pe	پ	پ	پ	پ
4	Te	ت	ت	ت	ت
5	Se	ث	ث	ث	ث
6	Jim	ج	ج	ج	ج
7	Che	چ	چ	چ	چ
8	He	ح	ح	ح	ح
9	Khe	خ	خ	خ	خ
10	Dal	د	د	د	د
11	Zal	ذ	ذ	ذ	ذ
12	Re	ر	ر	ر	ر
13	Ze	ز	ز	ز	ز
14	Zhe	ژ	ژ	ژ	ژ
15	Sin	س	س	س	س
16	Shin	ش	ش	ش	ش
17	Sad	ص	ص	ص	ص
18	Zad	ض	ض	ض	ض
19	Ta	ط	ط	ط	ط
20	Za	ظ	ظ	ظ	ظ
21	Ayn	ع	ع	ع	ع
22	Ghayn	غ	غ	غ	غ
23	Fe	ف	ف	ف	ف
24	Ghaf	ق	ق	ق	ق
25	Kaf	ک	ک	ک	ک
26	Gaf	گ	گ	گ	گ
27	Lam	ل	ل	ل	ل
28	Mim	م	م	م	م
29	Noon	ن	ن	ن	ن
30	Waw	و	و	و	و
31	He	ه	ه	ه	ه
32	Ye	ی	ی	ی	ی

خط نستعلیق علی‌رغم محبوبیت زیاد دارای قواعد و استثنائات دشوار و متعددی است که کار شناسایی کامپیوتری این سبک نوشتاری را بسیار دشوار می‌کند. در اینجا چند مثال که نشان‌دهنده این قواعد و مشکلات کار ما خواهند بود ذکر می‌شوند:

- حروف خانواده "ب" بسته به حرف بعد از آن به صورت مختلف نوشته می‌شود، مانند:

بگیر، بکار، ببار، بپوش، بسج، بس

- حروف خانواده "ح" بسته به حرف بعد از آن به صورت مختلف ظاهر می‌شود، مانند:

حد، حلا، حلا، حجا، حح، ححی، حم، حر

- حروف خانواده "ک" بسته به محل حرف در کلمه به صورت مختلف ظاهر می‌شوند، مانند:

کمال، کمال، نگار، نگار، کنگار، کنگار، کنگار، کنگار، کنگار، کنگار

- حرف "م" به دو صورت مستطیلی و گرد نوشته می‌شود، مانند:

مرد، مرد

- حرف "ه" بسته به محل آن در کلمه به شکلهای مختلف می‌تواند ظاهر شود، مانند:

پره، پره، جهاد، جهاد، جهاد، جهاد، پیران، پیران، ده

- دندانه‌ها در وسط کلمات گاهی بصورت قوسی ظاهر می‌شوند، مانند:

سرسر، سرسرسر، سرتی

- محل حروف نسبت به خط کرسی وسط بسته به آنکه حروف دیگر کلمه چه باشند متفاوت است، مانند:

جم، حد، حح

- بعضی از حروف و یا ترکیبات آنها روی خط کرسی قرار می‌گیرند و بعضی نه. در حقیقت بخشهایی از کلمات نسبت به خط کرسی با زاویه حدود ۳۰ درجه نوشته می‌شوند، مانند:

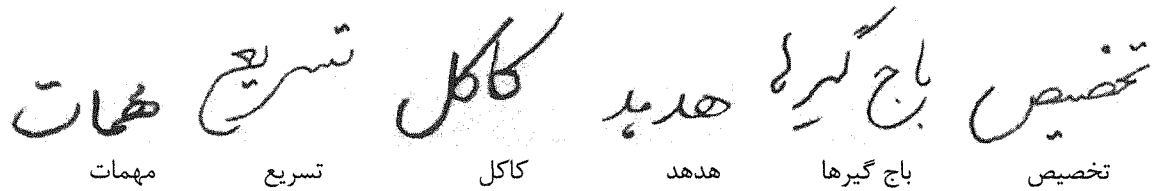
صلوات، صبر، صفات

اگر حتی از مشکلات حادث از وجود شکلهای مختلف برای بعضی حروف در حالات مختلف صرفنظر کنیم، دو قاعده آخر از موارد فوق برای نشان دادن پیچیدگی مسأله تقطیع کلمات نستعلیق به تنهایی کافی می‌باشند.

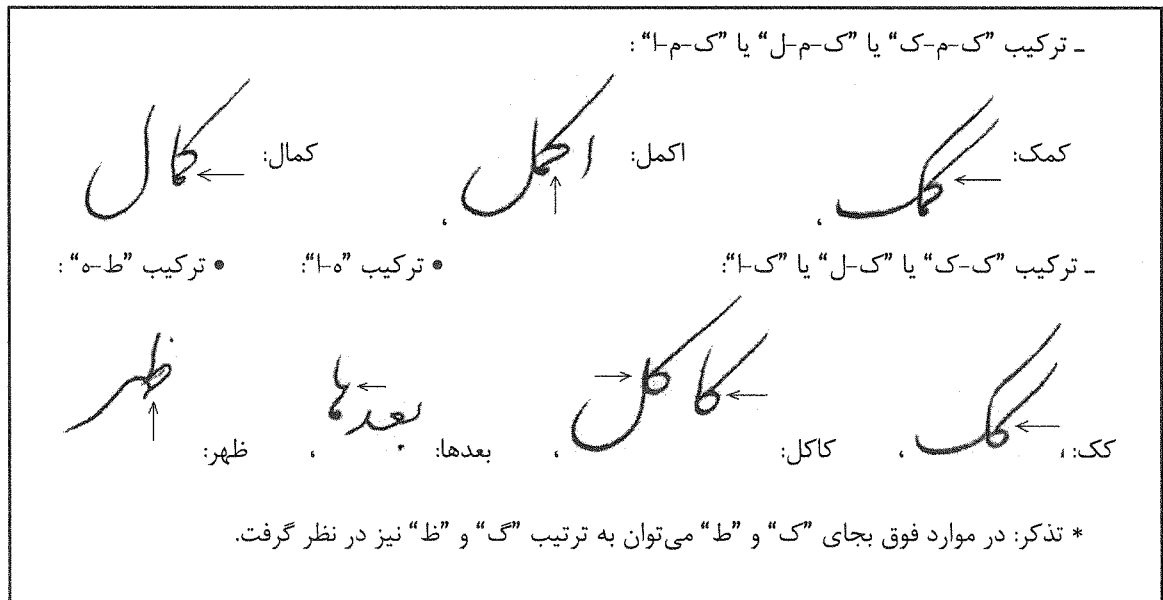
۱-۲- پایگاه کلمات فارسی بکار رفته

ما برای محدود کردن تعداد الگوهای قابل شناسایی، مجموعه‌های آموزشی و آزمایشی را محدود به سبک نستعلیق تحریری، در نظر گرفته‌ایم. البته کلمات این مجموعه‌ها به نحوی انتخاب شده‌اند که تمام حروف و الگوهای ترکیبی موجود در سبک

نستعلیق را شامل شوند. شکل ۲ چند نمونه از کلمات مجموعه آزمایشی و شکل ۳ چهار الگوی ترکیبی در سبک نستعلیق فارسی را نشان می‌دهد. ویژگیهای مجموعه های آموزشی و آزمایشی انتخاب شده در بخش ۱- بتفصیل ذکر شده است.



شکل (۲) چند نمونه از کلمات مجموعه آزمایشی.



شکل (۳) چهار الگوی ترکیبی در سبک نوشتاری نستعلیق فارسی.

۲- مدل پنهان مارکف و روشهای شناسایی متون مبتنی بر آن

۲-۱- آشنایی با مدل‌های پنهان مارکف

مدل پنهان مارکف مبتنی بر فرآیندهای تصادفی مضاعف^۲ می‌باشد که فرآیند تصادفی پایه آن مستقیماً قابل مشاهده نیست (یعنی مخفی است). انتقال سیستم از حالت جاری به حالت بعدی مبتنی بر این فرآیند تصادفی پایه انجام می‌شود. اما خروجیهای قابل مشاهده (مشاهدات) از فرآیند تصادفی دیگری حاصل می‌شوند. یک مدل پنهان مارکف با سمبل‌های مشاهده گسسته بصورت $\lambda = (A, B, \Pi)$ نمایش داده می‌شود که A ماتریس احتمال انتقال حالات، B توزیع احتمال سمبل‌های مشاهده (بفرم گسسته)، و Π توزیع احتمال حالت اولیه است [۷].

در بعضی از کاربردها چگالی مشاهدات بصورت پیوسته و احتمال تداوم حالات بطور صریح در نظر گرفته می‌شوند. مثلاً در [۴] یک مدل CDVDHMM^۴ بکار رفته است. این مدل بصورت $\lambda = (\Pi, A, \Gamma, B, D)$ نمایش داده می‌شود که پارامترها عبارتند از:

$$\Pi = \{\pi_i\}; \quad \pi_i = \Pr\{i_1 = q_i\}, \quad i_1 = \text{First State} \quad \text{احتمال حالت اولیه:} \quad (1)$$

$$A = \{a_{ij}\}; \quad a_{ij} = \Pr\{q_j \text{ at } t+1 | q_i \text{ at } t\} \quad \text{احتمال انتقال حالات:} \quad (2)$$

$$\Gamma = \{\gamma_i\}; \quad \gamma_i = \Pr\{i_T = q_i\}, \quad i_T = \text{Last State} \quad \text{احتمال حالت انتهایی:} \quad (3)$$

$$B = \{b_j(O_t^{t+d})\}; \quad O_t^{t+d} = (o_t o_{t+1} \dots o_{t+d}), \quad b_j(O) = \Pr\{O | q_j\} \quad \text{احتمال سمبل:} \quad (4)$$

$$D = \{P(d | q_i)\}; \quad P(d | q_i) = \Pr\{\text{duration}(q_i) = d\} \quad \text{احتمال تداوم حالت:} \quad (5)$$

در مطالعه مدل‌های پنهان مارکف سه مسأله اصلی مطرح است: (۱) با داشتن دنباله مشاهدات $O = \{o_1, \dots, o_T\}$ و مدل، چگونه احتمال وقوع این دنباله مشاهدات را با داشتن مدل مذکور، $P(O | \lambda)$ ، بطور موثر بدست آوریم؟ این مسأله را مسأله ارزیابی گویند. (۲) با داشتن دنباله مشاهدات و مدل، چگونه می‌توانیم یک دنباله حالات $q = \{q_1, \dots, q_T\}$ مربوطه انتخاب کنیم، بطوریکه از بعضی جهات مشخص بهینه باشد (یعنی بهتر دنباله مشاهدات را توضیح دهد)؟ به این مسأله، مسأله یافتن دنباله حالات بهینه گویند. (۳) چگونه می‌توانیم پارامترهای مدل را طوری تنظیم کنیم که $P(O | \lambda)$ را حداکثر کند؟ این مسأله مسأله آموزش مدل نام دارد.

راه حل مسأله (۱) فرآیند جلورو یا عقب رو است. برای مسأله (۲) متداولترین معیار بهینه سازی پیدا کردن یک دنباله حالات بهینه (یک مسیر بهینه) است که با الگوریتم ویتربی^۵ انجام می‌شود. برای حل مسأله (۳) نیز می‌توان از روش تخمین مجدد بام-ولش^۷ استفاده کرد [۷].

۲-۲ - شناسایی متون با مدل پنهان مارکف

دو روش کلی در شناسایی کلمات با استفاده از مدل پنهان مارکف وجود دارد [۳]. روش اول مدل پنهان مارکف تمیز دهنده مدل^۸ نامیده می‌شود. در این روش برای هر دسته از الگوها (هر کلمه در فرهنگ لغت) یک مدل ساخته می‌شود. سپس با ارائه یک الگو جهت شناسایی، امتیاز تطبیق آن با هر مدل محاسبه می‌شود و هر مدل که امتیاز بیشتری داشت، الگو به دسته مربوط به آن تعلق می‌گیرد. این روش برای مواردی که تا چند صد کلمه داریم، مفید و معقول است. اما اگر اندازه فرهنگ لغت به ۱۰۰۰ کلمه یا بیشتر برسد، از نظر حافظه و میزان محاسبات پر هزینه خواهد بود. روش دوم که مدل پنهان مارکف تمیز دهنده مسیر^۹ نامیده می‌شود و برای یک فرهنگ لغات بزرگ یا متغیر مناسبتر است، شامل ساخت تنها یک مدل برای تمام دسته ها و استفاده از مسیرهای مختلف یعنی دنباله حالات مختلف برای شناخت یک دسته از دیگری است. یک الگوی آزمایشی به دسته‌ای تعلق می‌گیرد که دارای مسیری با بیشترین احتمال از بین تمام مسیرها باشد.

بعضی از پژوهشگران روشهای تمیز دهنده مسیر را برای کلمات دستنویس لاتین بکار برده‌اند [۳][۴][۸][۹]. در [۸] از مدل مرتبه دوم نیز استفاده شده که برای لغات طولانی نسبت به مدل مرتبه اول پاسخ بهتری داده است. در [۳] کلمات بدون محدودیت فرض شده و الگوهای حاصل از تقطیع بعنوان حالات مدل در نظر گرفته شده‌اند. در این سیستم تعداد حالات ممکن است زیاد شده و به کاهش سرعت و دقت عملکرد بیانجامد. در [۴] این مشکل با استفاده از یک روش تقطیع زیادی^{۱۰}، که مشکل عدم تقطیع را ندارد، و در نظر گرفتن طول متغیر برای حالات حل شده است. همچنین با در نظر گرفتن چگالی مشاهدات پیوسته عملکرد بهبود یافته است. مشکل این سیستم، آموزش غیر مطمئن احتمال تداوم حالات در پایگاه داده‌های محدود است. برای رفع این مشکل سیستمی در [۹] ارائه شده که مستقل از احتمال طول حالات عمل می‌نماید.

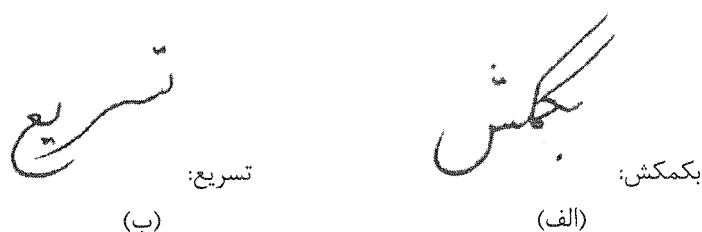
در [۹]، [۱۰]، [۱۱]، [۱۲] از روشهای تمیز دهنده مدل استفاده شده است. در این سیستمها برای حروف، مدل چپ به راست (در خط لاتین) یا راست به چپ (در خط عربی) در نظر گرفته شده و از اتصال آنها مدل کلمات بدست می‌آید. در [۱۳] و [۴۰] از HMM برای مدل کردن یک کلمه بصورت دنباله ای از فریمهای هم اندازه استفاده شده است، بنابراین این روشها جزء روشهای کلی و تمیز دهنده مسیر محسوب می‌شوند. در [۱۴] و [۱۵] مدل‌های پنهان مارکف دوبعدی برای شناسایی کلمات چاپی عربی بکار رفته‌اند و در [۱۶] از مدل برای افزایش قدرت شناسایی زیر کلمات چاپی فارسی استفاده شده است. در [۳۸] از HMM تنها برای مدل کردن حروف مجزای فارسی استفاده می‌شود. [۳۹] نیز یک الگوریتم DTW تعمیم یافته را برای شناسایی کلمات بصورت کلی بکار برده است.

۳- پیش پردازش

در مرحله پیش پردازش ابتدا تصویر ورودی با روش حداکثر واریانس [۳۲]، که یک روش تکرار شونده ساده است، باینری می‌شود. سپس عملگرهای مورفولوژیکی بستن و به دنبال آن باز کردن به ترتیب با عناصر ساختاری 3×3 و 2×2 به منظور حذف نویزهای ضربه‌ای بر روی تصویر اعمال می‌شوند [۴]. اجزاء متصل با روش ارائه شده در [۱۷] بدست می‌آیند. در این الگوریتم با شروع از سطر بالایی، مستطیل‌های فراگیر^{۱۱} بدست می‌آیند. سپس با افزودن سطرها بعدی، ابعاد این مستطیل‌ها اصلاح می‌شوند، و با رسیدن به انتهای تصویر مستطیل‌های فراگیر نهایی و اجزاء متصل مربوطه بدست می‌آیند. پهنای قلم با روش ارائه شده در [۳] تخمین زده می‌شود. در این روش متناوباً مقدار متوسط طول تداوم عمودی در هر ستون محاسبه شده و تداومهایی که بیش از $1/5$ برابر این متوسط باشند در مراحل بعدی در میانگین گیری وارد نمی‌شوند.

برای کلمات دستنویس فارسی، و بویژه سبک نوشتاری نستعلیق، یافتن خط کرسی کاری مشکل و غیر قابل اطمینان است. در بعضی موارد برای سبک نستعلیق بیش از یک خط کرسی و یا یک خط کرسی زاویه‌دار نیز در نظر گرفته شده است [۳۳]. بنابراین اطلاعات هیستوگرام افقی کلمات نمی‌تواند برای یافتن خط کرسی کافی باشد. روشهای دیگری هم برای یافتن خط کرسی پیشنهاد شده است (مثلاً در [۱۸])، که به علت پیچیدگی زیاد تنها در مواردی که واقعاً ضروری است باید بکار روند. بنابراین عملیات سیستم برای شناسایی کلمات بگونه‌ای طراحی شده است که مستقل از خط کرسی باشد.

در این سیستم، از آنجا که مدل مورد استفاده برای شناسایی یک دنباله از مشاهدات را دریافت می‌کند، پس از انجام تقطیع باید ترتیب راست به چپ قطعه‌ها مشخص گردد. در غیاب خط کرسی، وجود سرکشهای حروف "ک و گ" (شکل ۴-الف)، و همچنین زیرکشهای حروف "ج، چ، ح، خ، ع، و غ" (شکل ۴-ب) باعث تشخیص نادرست ترتیب راست به چپ قطعه‌ها می‌گردد. بنابراین مناسب است که در بخش پیش پردازش، سرکشا و زیرکشهایی که مشکل ساز هستند را حذف نماییم. این کار همچنین می‌تواند از تقطیعهای نادرست این سرکشا و زیرکشا جلوگیری نموده و به این ترتیب عملکرد الگوریتم تقطیع را بهبود بخشد. همچنین از آنجا که اساس شناسایی با استفاده از بدنه اصلی کلمه و حروف آن انجام می‌شود، در بخش پیش پردازش باید نقاط و علائم اضافه بر بدنه نیز از تصویر کلمات حذف شوند. سپس این نقاط و علائم در یک بخش پس پردازش مورد بررسی قرار گرفته و شناسایی را کامل می‌کنند. برای حذف سرکشا، زیرکشا، و نقاط، از الگوریتمهای جدیدی استفاده شده است که در ادامه توضیح داده می‌شود.



شکل (۴) مسائلی ناشی از سرکش و زیرکش:

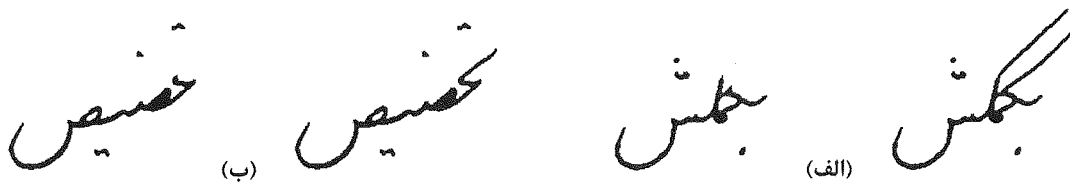
(الف) "ک" جلوتر از "ب" تشخیص داده می‌شود. (ب) "ع" جلوتر از "ی" تشخیص داده می‌شود.

۳-۱- حذف سرکشا

حروف "ک و گ" دارای سرکشهایی هستند که می‌خواهیم آنها را حذف نماییم. ویژگیهای خاصی که سرکش را از سایر بخشهای یک کلمه قابل شناسایی می‌کنند، شیب تقریبی 45° درجه، مستقیم بودن، و طول نسبتاً بزرگ آن است. این سه ویژگی اساس عملکرد الگوریتم کشف و حذف سرکش می‌باشد. این الگوریتم ابتدا برای هر جزء متصل، مرز پایینی را با استفاده از کد زنجیره‌ای مرز خارجی، که در مرحله یافتن اجزاء متصل بدست آمده است، پیدا می‌کند. جهات شرق، شمال شرق و جنوب شرق از کد زنجیره‌ای مرز خارجی بیانگر مرز پایینی می‌باشند. سپس بالاترین نقطه مرز پایینی بدست می‌آید و پیمایش مرز پایینی از آن نقطه آغاز شده و تا وقتی که شرط توقف غلط باشد ادامه می‌یابد. شرط توقف وقتی برقرار می‌شود که حرکت بیش از تعداد دفعات مشخصی در جهت قطری به سمت پایین و چپ نباشد. این تعداد دفعات برای هر جهت بطور مجزا تعیین

شده است [۱۹].

در حین این پیمایش تعداد انتقالهای صحیح، $k1$ ، (یعنی در جهت قطری به سمت پایین و چپ) و تعداد کل انتقالها، k ، (یعنی طول این قطعه از مرز) شمارش می‌شود. پس از برقرار شدن شرط توقف اگر k نسبتاً کوچک باشد (کمتر از $4/95$ برابر پهنای قلم) و $k1/k$ نسبتاً بزرگ باشد (بیش از $0/33$)، یک سرکش کوچک پیدا شده است. اگر k نسبتاً بزرگ باشد (بیش از $4/95$ برابر پهنای قلم) و $k1/k$ نیز نسبتاً بزرگ باشد (بیش از $0/58$)، یک سرکش بزرگ پیدا شده است. در این صورت انتهای سرکش بریده شده و از مابقی زیر کلمه جدا می‌شود. سپس مجدداً زیر کلمه برای یافتن سرکشهای دیگر مورد جستجو قرار می‌گیرد. اعداد بکار رفته بعنوان حد آستانه‌ای با آزمایش بر روی نمونه‌های آزمایشی بهترین کارایی را نشان داده‌اند. پس از یافتن سرکشها، یک سری شرایط برای اعتبارسنجی آنها بررسی می‌شود تا حتی‌الامکان سرکش را از موارد مشابه تشخیص دهد [۱۹]. در صورتیکه سرکش معتبر تشخیص داده شد، از تصویر حذف می‌گردد. در شکل ۵ نمونه‌ای از عملکرد این الگوریتم نشان داده شده است. همانطور که در بخش (ب) از شکل ۵ ملاحظه می‌شود حرف "ت"، از آنجا که با شیب 45 درجه و طول نسبتاً زیاد نوشته شده، اشتباهاً بجای سرکش حذف گردیده است. برای رفع اینگونه مشکلات می‌توان مرحله اعتبارسنجی سرکشها را قویتر نمود.



شکل (۵) عملکرد الگوریتم حذف سرکشها: (الف) عملکرد صحیح (ب) عملکرد نادرست.

۳-۲ - حذف زیرکشها

برای کشف زیرکشهای حروف "ج، چ، ح، خ، ع، و غ" که ترتیب راست به چپ را بر هم می‌زنند، از ویژگیهای زیر استفاده می‌شود: تصویر شامل زیرکش بدین صورت است که اگر از سمت راست ترین ستون حاوی پیکسل سیاه در آن شروع کرده و با انتخاب پایین ترین قطعه پیوسته سیاه، ستون به ستون این قطعه سیاه را به سمت چپ دنبال کنیم تا اینکه به یک تکه سیاه دیگر متصل شود، اولاً طول همپوشانی این قطعه سیاه با قطعه‌های سیاه بالایی نسبتاً بزرگ است (بیش از $2/5$ برابر پهنای قلم). ثانیاً طول خود این قطعه نیز نسبتاً بزرگ است (بیش از 5 برابر پهنای قلم). ثالثاً حداقل در یک ستون بیش از 2 قطعه سیاه موجود می‌باشد. این ویژگیها، زیرکشها را از سایر تکه نوشته‌های یک کلمه به خوبی مجزا می‌کنند. شکل ۶ نمونه‌ای از عملکرد این الگوریتم را نشان می‌دهد.



شکل (۶) عملکرد الگوریتم حذف زیرکشها.

۳-۳ - حذف نقاط و علائم

برای کشف نقاط و علائم می‌توان ویژگیهایی چون اندازه کوچک و قرار داشتن در محدوده پهنای یک زیرکلمه بزرگتر را در نظر گرفت. در اینجا اجزاء متصلی که پهنایشان کمتر از $2/5$ برابر پهنای قلم و ارتفاعشان کمتر از 5 برابر پهنای قلم یا برعکس باشد و حداقل در 25% از پهنای خود با یک جزء متصل بزرگتر همپوشانی داشته باشند، بعنوان نقطه یا علامت فرض شده و از تصویر حذف می‌گردند.

مقادیر به‌گونه‌ای در نظر گرفته شده‌اند که حتی‌الامکان زیر کلمه دیگری بجای نقطه‌ها یا علائم حذف نشود. این امر باعث شده‌است که تعدادی از نقاط و علائم در بعضی از نمونه‌ها که بزرگتر از حد معمول یا با فاصله زیاد از بدنه کلمه نوشته شده‌اند،

حذف نشوند. برای بهبود عملکرد الگوریتم حذف نقاط و علائم، یک راه، استفاده از یک دسته‌بندی کننده اولیه در مرحله پیش پردازش است که تنها قادر به جدا کردن نقاط و علائم از حروفی که از لحاظ اندازه به نقاط و علائم نزدیک هستند (مثل فرم جدای "و، ه، ر، ز، د، ذ، ا") باشد [۱۹]. از آنجا که در این حالت تعداد دسته‌ها کمتر است، ویژگیهای استخراج‌شونده از تصاویر می‌توانند ساده‌تر بوده و برای تشخیص مورد نظر بهینه شوند.

۴- تقطیع و یافتن ترتیب راست به چپ

در این سیستم هدف رسیدن به یک تقطیع زیادی است، بطوریکه حتی الامکان بین هر دو حرف یک برش انجام شود. بدین ترتیب می‌توان حروف را بعنوان حالتها در مرحله شناسایی در نظر گرفت [۴]. در مواقعی که یک حرف به بیش از یک قطعه شکسته می‌شود مدل پنهان مارکف با طول حالات متغیر، که در این سیستم بکار رفته است، این مسأله را پوشش می‌دهد. پس از عمل تقطیع باید ترتیب راست به چپ قطعه‌ها بدست آید تا در مرحله شناسایی مورد استفاده قرار گیرد. روشهای مختلف پیشنهاد شده برای تقطیع متون فارسی و عربی، از نظر قابلیت برقرار سازی این معیار بررسی شده‌اند. روشهای مبتنی بر خط کرسی و روشهای مبتنی بر نماهای عمودی [۲۰][۲۱][۲۲][۲۳] برای کلمات دستنویس، خصوصاً سبک نستعلیق، مناسب نمی‌باشند. همچنین روشهایی که پهنای عمودی خط نوشتار را مبنای یافتن محلها برش قرار می‌دهند [۲۴] نیز برای خطوط دستنویس نسبتاً آزاد چندان مناسب بنظر نمی‌رسند. ما دو روش بهبود یافته برای تقطیع کلمات ارائه کرده‌ایم.

روش اول از ایده اجزاء منظم^{۱۲} و منفرد^{۱۳} استفاده می‌کند و اجزاء منظم را بعنوان محلهای کاندید برای انجام برش در نظر می‌گیرد. روش دوم بر اساس تحلیل مرز بالایی کلمات، عمل تقطیع را انجام می‌دهد. در ادامه این دو روش تقطیع و روش بکار رفته برای یافتن ترتیب راست به چپ قطعه‌ها توضیح داده می‌شوند.

۴-۱- تقطیع با استفاده از اجزاء منظم و منفرد

ایده تقطیع با استفاده از اجزاء منظم و منفرد در [۲۵]، [۳]، و [۴] برای کلمات دستنویس لاتین و در [۱۷] و [۳۴] برای کلمات دستنویس عربی و فارسی بکار رفته است. ما با الهام گرفتن از این مراجع و انجام بعضی اصلاحات روشی مبتنی بر این ایده برای تقطیع کلمات دستنویس فارسی پیاده سازی کرده‌ایم. در این روش پس از انجام مراحل پیش پردازش، ابتدا حفره‌های موجود در تصویر (مانند حلقه‌های حروف "ص"، "ط"، "ه"، و غیره) پر می‌شود. این کار از آن جهت انجام می‌شود که در محل این حفره‌ها برش انجام نشود. سپس با استفاده از یک عنصر ساختاری عمودی با ارتفاع اندکی بیش از پهنای قلم (به اندازه یک یا دو پیکسل) عمل مورفولوژیکی باز کردن بر روی تصویر اعمال می‌شود. بدین ترتیب بخشهای تقریباً عمودی تصویر بدست می‌آید. سپس عمل مورفولوژیکی بستن با یک عنصر ساختاری افقی با پهنای کم (در حدود سه تا پنج پیکسل) انجام می‌گیرد و طی آن بخشهای تقریباً عمودی بدست آمده از مرحله قبل که به یکدیگر نزدیک می‌باشند، بهم متصل می‌شوند. حاصل این عمل، اجزاء منفرد است که جزیره‌ها^{۱۴} نیز نامیده شده‌اند.

با تفریق این اجزاء از تصویر اولیه، اجزاء منظم یا پله‌ها^{۱۵} بدست می‌آیند. در این مرحله ممکن است بعضی حروف دارای اجزاء منظم متعددی باشند که به یک تقطیع زیادی غیر قابل قبول بیانجامد. برای کاهش این مشکل، اجزاء منظمی که پهنای آنها از حدی کمتر است (مثلاً کمتر از پهنای قلم تخمین زده شده) حذف می‌گردند، یعنی به اجزاء منفرد اضافه می‌شوند. بدین ترتیب خطاهای باقیمانده از عمل بستن اصلاح می‌شود. سپس از میان اجزاء منظم باقیمانده (پله‌ها) آنهایی که دو جزء منفرد (دو جزیره) را بهم وصل نمی‌کنند، یعنی شروع یا پایان زیر کلمه هستند حذف می‌شوند، یعنی به اجزاء منفرد اضافه می‌شوند و در وسط اجزاء منظم باقیمانده برش انجام می‌شود.

پارامترهای زیر در عملکرد این الگوریتم تاثیر گذار هستند:

- ارتفاع عنصر ساختاری باز کردن (VELength): هر چه این پارامتر بزرگتر باشد تعداد اجزاء منفرد کمتر شده و به تبع آن مشکل تقطیع زیادی کاهش یافته و مشکل عدم تقطیع بیشتر می‌شود. با توجه به اینکه تأکید ما بر کاهش مشکل عدم تقطیع است، برای این پارامتر مقدار $Est + 2$ که در آزمایشها جواب بهتری داده انتخاب شده است (Est پهنای قلم تخمین

زده شده است).

– پهناى عنصر ساختارى بستن (HELlength): این پارامتر نیز مثل پارامتر قبلى هر چه بزرگتر باشد مشکل عدم تقطیع را افزایش و تقطیع زیادى را کاهش مى‌دهد. در [۴] مقدار ۵ برای آن پیشنهاد شده است. ما مقدار ۳ را برای این پارامتر انتخاب کرده‌ایم.

در شکل ۷ اجزاء منفرد و منظم و نحوه تقطیع با استفاده از آنها برای چند کلمه نشان داده شده است.

بی فکر محبوب یاسمن
(الف)

بی فکر محبوب یاسمن
(ب)

بی فکر محبوب یاسمن
(پ)

شکل (۷) عملکرد تقطیع مبتنی بر اجزاء منظم و منفرد: (الف) تصاویر باینری و حذف نویز شده سه کلمه "بی فکر"، "محبوب"، "یاسمن" (ب) اجزاء منفرد و منظم به ترتیب با رنگهای سیاه و خاکستری (پ) نتیجه تقطیع.

۴-۲- تقطیع با استفاده از مینیمم‌های مرز بالایی

در [۲۶] مینیمم‌های محلی مرز بالایی بعنوان نقاط کاندید انجام برش در نظر گرفته شده‌اند. سپس در صورت برقرار بودن یک سری از شرطها، تقطیع در این نقاط انجام می‌شود. همچنین محل‌های همپوشانی نیز کشف شده و در صورت لزوم تقطیع در آن محلها انجام می‌شود. در [۳۵]، [۳۶]، [۳۷]، [۴۱] نیز از اطلاعات کانتور بالایی برای تقطیع کلمات فارسی استفاده شده است. ما با انجام یک سری اصلاحات در روش ارائه شده در [۲۶] آن را برای تقطیع کلمات دستنویس نستعلیق فارسی بهبود داده‌ایم.

از آنجا که در خط نستعلیق حرف "ر" چسبان بدون وجود مینیمم محلی در قبل از آن نوشته می‌شود، این روش قادر به تقطیع آن نیست. بنابراین ابتدا یک الگوریتم جدید برای کشف و تقطیع حرف "ر" چسبان بکار می‌رود. سپس همپوشانیها کشف شده و تقطیع مناسب در آن محلها انجام می‌شود. آنگاه مرز بالایی با یک روش ساده بدست آمده و مینیمم‌های محلی آن بعنوان نقاط اولیه تقطیع بدست می‌آید. سپس این نقاط اعتبارسنجی می‌شوند و در مواردی که معتبر تشخیص داده می‌شوند به طریق مناسب تقطیع انجام می‌شود. الگوریتمهای این مراحل در ادامه توضیح داده می‌شوند.

۴-۲-۱- کشف و تقطیع "ر" چسبان

برای کشف "ر" چسبان از ایده‌ای مشابه با روش کشف سرکشها استفاده شده، اما ایده به‌نحو کاراتری پیاده‌سازی گردیده‌است. ویژگیهای مجزاکننده "ر" چسبان از سایر تکه‌نوشته‌ها، دارا بودن شیب تقریبی ۴۵ درجه و بلند بودن نسبی طول آن است. الگوریتم پیشنهادی به قرار زیر است:

- الگوریتم کشف "ر" چسبان و تقطیع آن از زیر کلمه

- ۱- برای هر جزء متصل معتبر:
 - ۱-۱- برای هر ستون از این جزء متصل، از ستون سمت چپ تا ستون سمت راست:
 - ۱-۱-۱- اگر (بیش از یک تداوم سیاه در ستون جاری وجود داشت یا پهنای یکی از تداومها بزرگتر از $PR0*Est$ بود): به مرحله ۱-۲ برو.
 - ۱-۱-۲- اگر (پایین‌ترین نقطه از ستون جاری، بیش از $PR1$ پیکسل پایین‌تر از پایین‌ترین نقطه از ستون قبلی باشد یا بیش از $PR2$ پیکسل پایین‌تر از پایین‌ترین نقطه از اولین ستون از روند کاشی جاری باشد (برای لحاظ کردن بالآمدگی احتمالی انتهای "ر")): به مرحله ۱-۲ برو.
 - ۱-۲- اگر (تعداد ستونهایی که در مرحله ۱-۱ پیمایش شد بیشتر از $PR3*Est$ نباشد): زیر کلمه شامل "ر" نیست. در غیر اینصورت:
 - ۱-۱-۲- مرز پایینی پیمایش شده بصورت قطعه مرزهایی بطول ۳ در نظر گرفته شده و به هر قطعه بر حسب اینکه بیشتر نزدیک به افقی است یا شیبدار به ترتیب برچسب‌های 'H' یا 'S' زده می‌شود.
 - ۱-۲-۲- توسط یک ماشین حالت، دنباله حاصل از برچسب‌های 'H' و 'S' هموار می‌شود (مثلاً "SHS" تبدیل به "SSS" می‌شود و غیره).
 - ۱-۲-۳- اگر (تعداد ستونهایی که شیبدار محسوب شده‌اند بیش از $PR4*Est$ بود) در انتهای سمت راست ستونهای پیمایش شده یک برش به رنگ SEGMENTATION_COLOR ایجاد می‌شود.

- نکات مربوط به الگوریتم کشف "ر" چسبان

مقادیر پارامترها عبارتند از: $PR0=2.9$, $PR1=1$, $PR2=1$, $PR3=4.8$, $PR4=2$. عملکرد روش پیشنهادی بسیار مناسب است. شکل ۸ نمونه‌ای از عملکرد این روش را نشان می‌دهد.



شکل (۸) عملکرد الگوریتم کشف "ر" چسبان برای سه کلمه "ظهر"، "شرف"، "تیررس".

۴-۲-۴- کشف همپوشانی و انجام تقطیع مناسب

تقطیع با استفاده از مینیمم‌های مرز بالایی علاوه بر ناتوانی در تقطیع "ر" چسبان، برای مواردی مثل "ح" موجود در وسط کلمه در سبک نستعلیق (مثل **مُحَرَّب**) یا "ی" چسبان آخر (مثل **انگلیسی**) یا "م" وسط کلمه در سبک نستعلیق (مثل **بِاسْمِ**) نیز عمل تقطیع را انجام نمی‌دهد. این مشکل می‌تواند با یافتن همپوشانی‌ها و تقطیع مناسب در آن محلها رفع شود (شکل ۹). ما الگوریتمی برای کشف همپوشانی‌ها و تقطیع مناسب آنها ارائه کرده‌ایم که در ادامه توضیح داده می‌شود:

- الگوریتم کشف همپوشانی و انجام تقطیع مناسب

- ۱- برای هر جزء متصل معتبر:
 - ۱-۱- برای هر ستون از این جزء متصل، از ستون سمت راست تا ستون سمت چپ:
 - ۱-۱-۱- تعداد، طول و محل تداومهای سیاه موجود در ستون جاری بدست آید.
 - ۱-۱-۱-۱- برای هر تداوم سیاه، با شروع از پایین‌ترین آنها:

تقطیع علامتگذاری می‌شود.

۳-۱- انجام تقطیع: در محلهای علامتگذاری شده برای تقطیع برشی به رنگ SEGMENTATION_COLOR ایجاد شود، به شرطی که در مجاورت دو سمت برش (چپ و راست) قطعه‌های سیاه وجود داشته باشد. اگر در یک سمت، برش با حلقه‌ای مجاور است، یک ستون از حلقه سیاه می‌شود تا این مجاورت از بین برود.

۴-۱- پس از انجام تقطیع، برشهایی که به ایجاد قطعه‌های با پهنای کوچکتر از یک حد مشخص (در اینجا ۳ پیکسل) منجر شده‌اند، پر می‌شوند.

شکل ۱۱ نمونه‌ای از عملکرد این الگوریتم تقطیع را نشان می‌دهد.



شکل (۱۱) عملکرد تقطیع مبتنی بر مینیمم‌های مرز بالایی.

۳-۴- یافتن ترتیب راست به چپ

برای یافتن ترتیب راست به چپ در [۴] الگوریتمی ارائه شده است که از پیچیدگی محاسباتی نسبتاً بالایی برخوردار است. الگوریتمی که در این پروژه برای یافتن این ترتیب ارائه شده است ساده‌تر عمل می‌کند. پس از حذف سرکشها و زیرکشها، می‌توان بدون داشتن خط کرسی ترتیب را بدست آورد. ابتدا ترتیب زیر کلمات را مشخص می‌کنیم. سپس در هر زیرکلمه ترتیب قطعه‌ها را با در نظر گرفتن سمت راست ترین قطعه بعنوان اولین قطعه زیرکلمه و با پیمایش کانتور خارجی تا انتهای زیرکلمه، بدست می‌آوریم. الگوریتم این روش در [۱۹] آمده است.

۵- استخراج ویژگیها

در یک سیستم شناسایی حروف انتخاب روش استخراج ویژگیها، احتمالاً از میان عوامل تأثیرگذار، مهمترین عامل در رسیدن به کارایی بالای عمل بازشناسی می‌باشد [۲۷]. تا کنون تعداد زیادی روش استخراج ویژگی در منابع مختلف گزارش شده است و این سؤال مطرح است که کدام روش برای یک کاربرد خاص بهترین روش است؟ برای پاسخ دادن به این سؤال، ابتدا باید شناختی از اصول مختلفی که می‌توانند در این زمینه بکار روند بدست آورد. سپس هنگامی که چند روش برای کاربردی مناسب تشخیص داده می‌شود، باید بهترین روش از طریق آزمایش شناخته شود.

ما پس از بررسی روشهای مختلف استخراج ویژگیها و آزمایش تعدادی از آنها [۱۹] یک بردار ویژگی تلفیقی از ویژگیهای مختلف از ارائه باینری و کانتور خارجی حروف بدست آورده‌ایم. معیارهای مدنظر در انتخاب ویژگیها قدرت بالای جداکنندگی^{۱۶}، استقلال از تغییرات موجود در مجموعه نمونه‌ها (در اینجا تغییر اندازه و انتقال)، سرعت بالای مرحله استخراج ویژگیها، و تعداد کم ویژگیها می‌باشد.

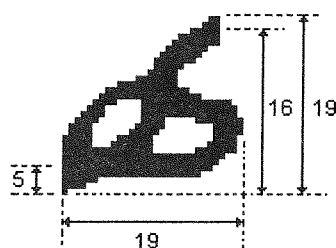
ویژگیهای آزمایش شده شامل گشتاورهای هندسی [۲۸] از ارائه باینری و نازکسازی شده، توصیفگرهای فوریه [۲۹] از ارائه کانتور باینری، ویژگیهای گسسته شامل حلقه، نسبت ارتفاع به پهنای، نسبت نقاط سیاه به کل نقاط، و محل اتصال از چپ و راست [۱۹] از ارائه باینری، و ویژگیهای توزیع پیکسلها و ویژگیهای گسسته دیگر شامل نقاط انتهایی، اتصالات T شکل، اتصالات X شکل و ویژگیهای قطع محورها (Zero Crossing) [۳] از ارائه نازکسازی شده بوده‌اند.

ترکیبات مختلفی از این ویژگیها بر روی نمونه‌های تقطیع شده ایده‌آل و با استفاده از یک دسته بندی کننده مجموع توابع گاسی، که در بخش ۶-۱-۱ توضیح داده می‌شود، مورد آزمایش قرار گرفته‌اند.

در نهایت یک بردار ویژگی هشت عنصری شامل سه توصیفگر اول فوریه (بدون جمله صفر)، ویژگی حلقه، نسبت ارتفاع به

پهنا، تعداد نقاط سیاه به کل نقاط و محل اتصال از چپ و راست انتخاب شده است. مقدار ویژگی حلقه برای قطعه هایی که حلقه ندارند برابر صفر، برای تک حلقه ها برابر $0/5$ ، و برای دو حلقه یا بیشتر برابر ۱ می باشد. مقدار دو ویژگی محل اتصال از چپ و راست برابر با ارتفاع محل اتصال قطعه به قطعه چپ و راستش تقسیم بر ارتفاع کل قطعه می باشد. اگر قطعه ای از یک یا هر دو سمت به قطعه دیگری متصل نباشد، مقدار این ویژگی در یک یا دو سمت مربوطه برای آن صفر قرار داده می شود. این بردار ویژگی تلفیقی علاوه بر داشتن قدرت بالای جدا کنندگی، طول کمی دارد که باعث افزایش سرعت عملکرد مراحل بعدی می گردد. همچنین در صورت استفاده از این ویژگیها، نیاز به نازک سازی وجود ندارد. لذا پیچیدگی محاسباتی مرحله استخراج ویژگیها بطور چشمگیری کاهش می یابد. توصیفگرهای فوری به نسبت به تغییر اندازه، دوران، نقطه شروع پیمایش کانتور، و انتقال نامتغیر هستند. سایر ویژگیها نیز نسبت به اندازه و انتقال نامتغیر هستند. بنابراین کل بردار ویژگی نسبت به اندازه و انتقال نامتغیر خواهد بود.

ما بجای نرمالیزه کردن مقادیر این ویژگیها که موجب یکسان شدن تأثیر جدا کنندگی آنها می شود، برای هر یک وزنهایی بطور تجربی بدست آوردیم. این وزنها، که بیانگر میزان اهمیت هر ویژگی می باشد، موجب افزایش قدرت جدا کنندگی در آزمایشها شد. توصیفگرهای فوری که در آغاز با مقدار جمله اول نرمالیزه شده اند، با وزن واحد بکار می روند. ویژگی حلقه دارای وزن 40 ، ویژگی نسبت ارتفاع به پهنا دارای وزن 15 ، ویژگی تعداد نقاط سیاه به تعداد کل نقاط دارای وزن 44 ، و ویژگیهای محل اتصال از چپ و راست دارای وزن 40 می باشند. این بردار ویژگی کارایی بالایی از خود نشان داده است. در شکل ۱۲ مقادیر ویژگیهای استخراج شده از یک حرف (قطعه) نشان داده شده است. جدول ۳ کارایی این ویژگیها در مقایسه با ویژگیهای پیشنهاد شده در [۳] را نشان می دهد.



شماره	ویژگی	مقدار
۱	توصیفگر فوری شماره ۱	۸/۶۶۷۳
۲	توصیفگر فوری شماره ۲	۲/۳۲۰۲
۳	توصیفگر فوری شماره ۳	۰/۹۰۸۲
۴	محل اتصال از راست	۴۱/۲۵
۵	محل اتصال از چپ	۱۲/۵
۶	تعداد نقاط سیاه به کل نقاط	۱۸/۱۸۶۷
۷	نسبت ارتفاع به پهنا	۱۵/۰۰
۸	حلقه	۴۰/۰۰

شکل (۱۲) تصویر حرف "ه" بدست آمده از تقطیع کلمه مهمات و بردار ویژگی استخراج شده از آن. ارتفاع، پهنا، و ارتفاع محلهای اتصال از چپ و راست بر حسب پیکسل نشان داده شده اند.

۶- شناسایی

برای مرحله شناسایی یک مدل CDVDHMM با اقتباس از [۴] بکار رفته است. بعضی از دلایل مفید بودن این مدل عبارتند از: (۱) خاصیت ترتیبی نوشتار: مدل پنهان مارکف در مدل کردن اطلاعات ترتیبی قدرتمند است. (۲) حالات مخفی: در شناسایی حروف هدف یافتن دنباله حروف (حالات مخفی) از روی دنباله تصاویر (مشاهدات) است. (۳) چگالی مشاهدات پیوسته: خطای کوانتیزاسیون برداری را حذف کرده و چند شکلی بودن حروف فارسی با مجموع توابع گاسی بخوبی مدل می شود. (۴) طول متغیر حالات: تقطیع زیادی را مدل می کند.

ما حالتها را فرمهای بدون نقطه حروف و الگوهای ترکیبی موجود در سبک نستعلیق در نظر گرفته ایم. بدین ترتیب مدل دارای ۲۵ حالت خواهد بود. می دانیم که حروف فارسی در موقعیتهای مختلف در کلمه می توانند شکلهای متفاوت داشته باشند و بعضاً در سبک نستعلیق یک حرف در یک موقعیت نیز می تواند دارای اشکال متفاوت باشد (مثل ه و ه). بنابراین در نظر گرفتن تمام اشکال یک حرف تحت یک الگوی قابل شناسایی نتیجه مناسبی نخواهد داد. برای رفع این مشکل ما یک مفهوم جدید بنام "زیر حالت" برای مدل تعریف کرده ایم. بدین ترتیب که شکلهای مختلف هر حرف را بعنوان زیر حالت های آن حرف در نظر گرفته و در مرحله آموزش با استفاده از تصاویر (بخش ۶-۱-۱)، پارامترها را برای هر زیر حالت بطور مجزا بدست می آوریم. در ادامه نقش زیر حالات در بالا بردن کارایی روشنتر خواهد شد.

۶-۱- آموزش مدل

در مرحله آموزش، هدف تخمین زدن پارامترهای مدل $\lambda = (\Pi, A, \Gamma, B, D)$ می باشد (روابط (۱) تا (۵)). همانطور که ذکر شد، در این سیستم حالات متناظر با حروف در نظر گرفته شده اند. بدین ترتیب حالات با معنی بوده و با پرهیز از روش بام-ولش مرحله آموزش بسیار ساده تر شده است [۴]. آموزش با استفاده از دو منبع آموزشی انجام می شود که عبارتند از تصاویر آموزشی و فرهنگ لغات موجود. پارامترهای B و D از مجموعه تصاویر آموزشی و مابقی پارامترها از فرهنگ لغات بدست می آیند.

۶-۱-۱- آموزش با استفاده از تصاویر

در این مرحله از آموزش، پارامترها برای هر زیر حالت بطور جداگانه محاسبه می شوند (مفهوم زیر حالت در ابتدای بخش ۶ توصیف شده است). بنابراین در این زیر بخش منظور از "حالت" همان "زیر حالت" می باشد. پس از اجرای الگوریتم تقطیع بر روی تصاویر آموزشی، با شمارش تعداد قطعه ها برای هر کاراکتر بطور دستی، احتمال طول حالات (D) بدست می آیند. احتمال اینکه حالت q_i دارای طول تداوم d باشد ($P(d|q_i)$) برابر است با تعداد دفعاتی که حرف q_i به d قسمت تقطیع شده است، تقسیم بر تعداد دفعاتی که حرف q_i در کلمات آموزشی ظاهر شده است. در نمونه های آموزشی حداکثر طول حالات یعنی مواردی که یک حرف به بیشترین تعداد قطعات شکسته می شد، چهار بود. اما برای لحاظ کردن تقطیع های بدتر، حداکثر طول حالات را شش در نظر گرفته ایم ($d = 1, 2, \dots, 6$).

چگالی احتمال مشاهدات برای حالات (B) بصورت یک مجموع محدود و وزن دار از توابع گوسی بصورت زیر در نظر گرفته می شود:

$$b_j(x) = \sum_{m=1}^{M_j} c_{jm} \cdot N[x, \mu_{jm}, U_{jm}], \quad 1 \leq j \leq N \quad (6)$$

که در این رابطه N یک توزیع گاسین را با بردار میانگین μ_{jm} و ماتریس کوواریانس U_{jm} ، برای m امین جزء ترکیب در حالت j، نشان می دهد. x برداری است که مدل می شود، و M_j تعداد ترکیبات گاسین در حالت j، و c_{jm} ضریب ترکیب برای جزء m در حالت j است. برای محدودیتهای احتمالاتی شرط زیر باید برقرار باشد:

$$\sum_{m=1}^{M_j} c_{jm} = 1, \quad 1 \leq j \leq N, \quad c_{jm} \geq 0, \quad i \leq m \leq M_j \quad (7)$$

برای یافتن تعداد توابع گاسین در هر حالت (M_j)، از الگوریتم خوشه بندی K-means با نرخ سیگنال به نویز (SNR) مشخص استفاده می شود و تعداد خوشه های بدست آمده برای هر حالت همان M_j مطلوب خواهد بود. برای محاسبه SNR از معیار J4 پیشنهاد شده در [۳۰]، که بر اساس معکوس ماتریس پراکندگی مختلط ضریبدر ماتریس پراکندگی درون کلاسی کار می کند، استفاده شده که مقدار ۰/۹ بطور تجربی بعنوان مقدار بهینه برای شرط توقف الگوریتم (SNR) بدست آمده است.

ضرائب ترکیب، c_{jm} ها عبارتند از تعداد نمونه‌های آموزشی موجود در H_{jm} تقسیم بر تعداد کل نمونه‌های آموزشی برای حالت q_j .

H_{jm} مجموعه اعضای خوشه m از حالت q_j است. برای هر خوشه در حالت q_j ، پارامترهای مربوط به توزیع گاسین بصورت زیر تخمین زده می‌شود:

$$\mu_{jm} = \sum_{x \in H_{jm}} \frac{1}{N_{jm}} x \quad (8)$$

$$U_{jm} = \sum_{x \in H_{jm}} \frac{1}{N_{jm}} (x - \mu_{jm})(x - \mu_{jm})^T \quad (9)$$

که x بردار ویژگی مربوط به نمونه آموزشی است و N_{jm} تعداد نمونه‌ها در H_{jm} می‌باشد. در پیاده سازی، ماتریس کوواریانس U_{jm} قطری در نظر گرفته می‌شود. بعلت محدود بودن تعداد نمونه‌های آموزشی، یک ثابت کوچک ρ به عناصر قطر اضافه می‌شود تا از منفرد شدن ماتریس جلوگیری شود [۴]. برای ρ مقدار ۰/۱ انتخاب شده است. در مرحله شناسایی، چگالی احتمال سمبل برای مشاهده O می‌تواند بصورت زیر محاسبه گردد:

$$b_j(O) = \sum_{m=1}^{M_j} c_{jm} \cdot \frac{1}{(\sqrt{2\pi})^n \cdot \det[U_{jm}]^{n/2}} \times \exp[-(O - \mu_{jm})^T U_{jm}^{-1} (O - \mu_{jm})]. \quad (10)$$

مشاهده O می‌تواند از یک یا چند قطعه متوالی تشکیل شده باشد. از آنجا که پس از عمل تقطیع قطعه‌های متوالی از لحاظ شکل بهم وابسته بوده و مستقل نمی‌باشند، احتمال سمبل بصورت زیر تعریف می‌شود:

$$b_j(o_1 o_2 \dots o_d) = b_j(O_1^d)^d \quad (11)$$

که O_1^d تصویری است که از ترکیب تصاویر o_1, o_2, \dots, o_d بدست آمده و توان d برای تنظیم کردن احتمال سمبل برای تعداد مختلف قطعه‌ها بکار می‌رود. این کار برای هنگامی که هر گره در شبکه ویتربی بیانگر یک قطعه است لازم می‌باشد [۴]. بدین ترتیب یک دسته‌بندی کننده خواهیم داشت که با داشتن یک الگوی آزمایشی، برای هر دسته پارامترهای مربوط به آن دسته را بازیابی می‌کند و طبق رابطه (۱۰) یک امتیاز برای آن دسته محاسبه می‌نماید.

۶-۱-۲ - آموزش با استفاده از فرهنگ لغات

سه مجموعه از پارامترهای CDVDHMM، یعنی احتمالات حالات اولیه (II)، احتمالات انتقال حالات (A) و احتمالات حالات انتهایی (Γ) باید از فرهنگ لغات بدست آیند. در اینجا حالات مورد نظر همان فرمهای بدون نقطه حروف هستند که تعدادشان ۲۵ عدد می‌باشد. پارامترها بصورت زیر بدست می‌آیند [۴]:

$$\pi_i = \frac{n1}{n2}, \quad a_{ij} = \frac{n3}{n4}, \quad \gamma_j = \frac{n5}{n6} \quad (12)$$

که در این روابط $n1$ تعداد کلمات شروع شونده با حرف متناظر با q_i ، $n2$ تعداد کل کلمات موجود در فرهنگ لغات، $n3$ تعداد انتقالها از حرف متناظر با q_i به حرف متناظر با q_j ، $n4$ تعداد کل انتقالها از حرف متناظر با q_i ، $n5$ تعداد کلماتی که با حرف متناظر با q_j خاتمه می‌یابند، و $n6$ تعداد کل کلمات موجود در فرهنگ لغات می‌باشد. بدین ترتیب در صورت تغییر فرهنگ لغات برای یک کاربرد جدید، این احتمالات به سادگی قابل محاسبه مجدد هستند و این ویژگی به سیستم قابلیت جابجایی

۳-۱-۶- فرآیند شناسایی

روش متداول مورد استفاده در مدل پنهان مارکف برای شناسایی یک دنباله از مشاهدات ورودی، الگوریتم ویتربی است. در این سیستم از نسخه اصلاح شده الگوریتم ویتربی، ارائه شده در [۴]، استفاده شده و اصلاحات جدیدی نیز با افزودن ایده زیر حالات در آن بکار رفته است تا برای خط فارسی با کارایی بالا قابل استفاده باشد. نسخه اصلاح شده الگوریتم ویتربی در ضمیمه ارائه شده است. این نسخه از سه جهت نسبت به نسخه استاندارد این الگوریتم تعمیم یافته است:

- ۱- با طول متغیر حالات سازگار شده است. بدین منظور الگوریتم ویتربی بر اساس رابطه زیر اصلاح می‌شود [۴]:

$$\delta_t(j) = \max_{1 \leq i \leq N, 1 \leq d \leq D} [\delta_{t-d}(i) a_{ij} P(d | q_j) b_j(O_{t-d+1})^d] \quad (۱۳)$$

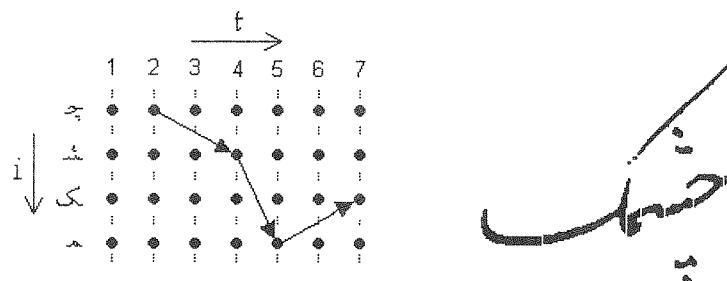
که D ، حداکثر طول حالات مجاز را نشان می‌دهد (در اینجا $D=6$) و O_{t-d+1} بیانگر بردار ویژگی استخراج شده از تصویر بدست آمده از اتصال تصویر d قطعه آخری به یکدیگر می‌باشد.

۲- نسخه استاندارد، تنها بهترین مسیر سراسری را در شبکه ویتربی بدست می‌دهد. برای بدست آوردن L بهترین مسیر سراسری در [۳] دو روش پیشنهاد شده است که عبارتند از روش موازی و روش سری و نشان داده شده که نسخه سری کارا تر عمل می‌کند. ما نیز همین روش را پیاده سازی کرده‌ایم.

۳- در الگوریتم بکاررفته در این سیستم، هر حالت می‌تواند دارای زیر حالت‌های مختلف (موقعیتها و شکل‌های مختلف حرف متناظر با آن حالت) باشد. موقعیت قطعه ورودی مشخص است (اول، وسط، آخر، یا جدا). بنابراین هنگام محاسبه احتمال مشاهده، تنها همین موقعیت از هر حالت بررسی می‌گردد. اما هر حالت در هر موقعیت می‌تواند دارای شکل‌های مختلف باشد که آنها نیز جزو زیرحالتها هستند. بدین ترتیب در هنگام محاسبه احتمال مشاهده X بردار ویژگی استخراج شده از تصویر ورودی است) در حالت Z ($b_j(X)$)، ابتدا آن زیر حالتی از حالت Z که بیشترین امتیاز را بدست می‌دهد پیدا می‌شود ($k = \arg \max_i \{b_{j_i}(X)\}$) و احتمال طول حالت ($P(d | q_{j_k})$) و امتیاز مربوطه ($b_{j_k}(X)$) از پارامترهای آن زیر حالت بدست می‌آید.

در این الگوریتم، شبکه ویتربی از دو بعد t و i به سه بعد t و i و l تعمیم یافته‌است. اما در نسخه سریال تمام عناصر بعد سوم پر نمی‌شوند بلکه تنها گره‌هایی از شبکه دو بعدی که برنده می‌شوند اجازه دارند که در بعد سوم بهترین مقدارهای بعدی را ایجاد کنند [۴]. این الگوریتم امتیاز هر گره از این شبکه (Lattice) را بدست آورده و سپس بهترین مسیرهای سراسری که از $t=1$ به $t=T$ ختم می‌شوند را بدست می‌آورد.

در هر تکرار الگوریتم، یک مسیر بهینه بدست می‌آید که بیانگر یک دنباله حروف (دنباله حالتها) یعنی یک کلمه است. مثلاً شکل ۱۳ بهترین مسیر را برای کلمه ورودی نشان داده شده، در یک شبکه دو بعدی نشان می‌دهد. در این مثال کلمه به هفت قطعه شکسته شده و بهترین مسیر با توجه به طول حالات بدست آمده است. در این شکل فقط بخشی از شبکه ویتربی و در حالت دوبعدی نشان داده شده است و شبکه واقعی دارای ابعاد $N * T * L$ می‌باشد (در اینجا $N=25$ تعداد حالات، $T=7$ تعداد قطعه‌ها، و $L=20$ حداکثر تعداد مسیرهای بهینه می‌باشند).



شکل (۱۳) مثالی از یافتن بهترین مسیر در شبکه ویتربی برای کلمه "چشمک".

۷- ارزیابی روشها

۷-۱- مجموعه‌های آموزشی و آزمایشی

برای آموزش و آزمایش این سیستم، نه نفر تعداد ۵۰ کلمه و مجموعه‌ای از علائم و نقاط را در سه نسخه مختلف به سبک نستعلیق تحریری و نسبتاً آزاد نوشته‌اند. این ۵۰ کلمه که فرهنگ لغات سیستم را نیز تشکیل می‌دهند بگونه‌ای انتخاب شده‌اند که تمام حروف فارسی و تمام ترکیبات ممکن از آنها در خط نستعلیق تحریری را شامل شوند. نمونه‌ای از این کلمات در شکل ۲ نشان داده شده است. این یک ویژگی مهم است زیرا سیستم با مشاهده این مجموعه کلمات در مرحله آموزش، بصورت بالقوه قادر به شناسایی کلماتی که تا کنون ندیده است نیز خواهد بود. البته هرچه مجموعه آموزشی بزرگتر باشد کیفیت این شناسایی نیز بالاتر خواهد بود. جدول ۲ مشخصات مجموعه تصاویر مورد استفاده را نشان می‌دهد.

جدول (۲) مجموعه تصاویر استفاده شده.

نام مجموعه	تعداد تصاویر کلمات	تعداد کلمات	توضیحات
مجموعه آموزشی TRN	۷۰۰	۵۰	-
مجموعه آزمایشی TST1	۳۵۰	۵۰	دستخط دیده شده - کلمات دیده شده
مجموعه آزمایشی TST2	۱۰۰	۵۰	دستخط ناشناس - کلمات دیده شده
مجموعه آزمایشی TST3	۲۱	۷	دستخط ناشناس - کلمات ناشناس

مجموعه تصاویر آموزشی شامل ۷۰۰ کلمه (دو نسخه ۵۰ کلمه‌ای از هفت نویسنده) می‌باشد. سه مجموعه آزمایشی برای ارزیابی مراحل مختلف بکار رفته است که عبارتند از: مجموعه TST1 شامل ۳۵۰ کلمه (نسخه ۵۰ کلمه‌ای سوم از همان هفت نویسنده)، مجموعه TST2 شامل ۱۰۰ کلمه (دو نسخه ۵۰ کلمه‌ای از دو نویسنده)، و مجموعه TST3 شامل ۲۱ کلمه (لغات خارج از فرهنگ لغات). مجموعه تصاویر آموزشی و آزمایشی از لحاظ کیفیت تصویر در درجات متفاوت، از ضعیف تا خوب، قرار دارند. مثلاً در میان نمونه‌های آموزشی و آزمایشی، حلقه‌های نوشته شده بصورت توپر، کلمات مخدوش و دارای شکستگی، نقطه‌های چسبیده به بدنه حروف، و بعضاً انحناهای غیر عادی مشاهده می‌شود. دقت پوششگر ۳۰۰ نقطه در اینچ با ۲۵۶ سطح خاکستری بوده است.

۷-۲- آزمایشها

۷-۲-۱- کارایی مراحل پیش پردازش و تقطیع

برای آزمایش عملکرد این مراحل مجموعه آزمایشی TST1 بکار رفته است. در بخش پیش پردازش تنها عملکرد مرحله حذف سرکشها داری خطا است. این الگوریتم ۹۷/۶۹٪ از سرکشها را بدرستی و ۰/۰۲۳٪ از سایر قطعه‌ها را به غلط بعنوان سرکش تشخیص داده و حذف می‌کند. این مشکل برای حروفی که در سبک نستعلیق بصورت شیبدار و کشیده، مشابه سرکش، نوشته می‌شوند پیش می‌آید. نمونه‌ای از این مسأله در شکل ۵-ب) نشان داده شده است. در مرحله تقطیع، همانگونه که ذکر شد، هدف پرهیز از مشکل عدم تقطیع بوده است. روش مبتنی بر اجزاء منفرد و منظم این معیار را در ۷۷/۶۲٪ موارد و روش مبتنی بر کانتور بالایی در ۹۵/۶۸٪ موارد برآورده می‌سازند. بنابراین روش دوم در سیستم مورد استفاده قرار گرفت. الگوریتم یافتن ترتیب راست به چپ در ۹۶/۸۵٪ از کلمات مجموعه آزمایشی عملکرد صحیح از خود نشان می‌دهد. اکثر خطاهای این مرحله در مواردی که الگوریتم تقطیع در وسط کلمه قطعه‌ای بوجود می‌آورد که تنها با یک قطعه مجاور است، رخ می‌دهد (شکل ۱۴).

شکل (۱۴) نمونه‌هایی که الگوریتم یافتن ترتیب راست به چپ برای آنها درست عمل نمی‌کند.

برای اصلاح این مشکل، می‌توان در این مرحله تمام قطعه‌هایی از یک زیرکلمه که تنها به یک قطعه دیگر چسبیده‌اند را پیدا کرد و اگر این قطعه‌ها، قطعه‌های اول یا آخر نبودند (که به راحتی برحسب مختصات آنها قابل تشخیص است) برش مربوط به آن قطعه پر شود و مجدداً اجزاء متصل بدست آیند و الگوریتم یافتن ترتیب بر روی این اجزاء متصل اجرا گردد.

۷-۲-۲- کارایی مرحله استخراج ویژگیها

برای بررسی ویژگیهای انتخاب شده، دسته بندی کننده مجموع توابع گاسی (مذکور در بخش ۶-۱-۱) با بردارهای ویژگی حروف مجموعه آموزشی TRN، که بصورت ایده‌آل از کلمه تقطیع شده‌اند، آموزش داده شده است. سپس این ویژگیها از حروف مجموعه تصاویر آزمایشی TST1 که بصورت ایده‌آل تقطیع شده‌اند، استخراج شدند. مجموعه آموزشی شامل تقریباً ۲۹۵۰ حرف و مجموعه آزمایشی شامل ۱۴۷۷ حرف می‌باشند. جدول ۳ صحت دسته‌بندی بردارهای آزمایشی را تا پنج انتخاب اول به همراه نتایج ارائه شده در [۳] نشان می‌دهد. مشاهده می‌شود که بردار ویژگی ما با تعداد ویژگیهای بسیار کمتر، عملکرد بهتری داشته است. افزودن تعداد دیگری از ویژگیهای ساختاری مانند نیم‌حلقه‌ها و جهت آنها نیز می‌تواند کارایی را افزایش دهد.

جدول (۳) کارایی ۸ ویژگی پیشنهادی ما در مقایسه با ۳۵ ویژگی [۳].

تعداد انتخاب قابل قبول	کارایی ۸ ویژگی پیشنهادی ما بر روی حرف ۱۴۷۷	کارایی ۳۵ ویژگی پیشنهاد شده در [۳] بر روی حرف ۶۱۵
۱ انتخاب اول	٪۸۱/۱۱	٪۵۱/۹
۲ انتخاب اول	٪۹۳/۲۲	٪۷۱/۹
۳ انتخاب اول	٪۹۶/۴۷	٪۸۲/۱
۴ انتخاب اول	٪۹۷/۶۳	٪۸۸/۳
۵ انتخاب اول	٪۹۸/۱۷	٪۹۱/۲

۷-۲-۳- کارایی مرحله شناسایی و نسخه الگوریتم ویتربی بکار رفته

نهایتاً الگوریتم ویتربی اصلاح شده، نتایج شناسایی را بدست می‌دهد. جدول ۴ این نتایج را برای کلماتی از مجموعه‌های آزمایشی که مراحل پیش پردازش، یافتن ترتیب راست به چپ و تقطیع برای آنها درست انجام شده است در ۵، ۱۰، و ۲۰ تکرار الگوریتم ویتربی نشان می‌دهد. ما نتایج شناسایی را در غیاب آمار فرهنگ لغات (با یکسان قرار دادن احتمالات حاصل از فرهنگ لغات برای تمام حالات) نیز بدست آورده‌ایم تا میزان تاثیر فرهنگ لغات را در شناسایی بسنجیم. دو ستون از جدول ۴ نتایج را برای مجموعه TST1 در حضور و عدم حضور فرهنگ لغات نشان می‌دهد. مشاهده می‌شود که ٪۹۶/۸ از این کلمات در ۲۰ انتخاب اول الگوریتم ویتربی شناسایی شده‌اند (شناسایی مستقیم [۴]). بنابراین بدون ایجاد کاندیدهای فرضی، نتایج بسیار بهتری در مقایسه با [۴] بدست آمده است (بهترین نتیجه ارائه شده در [۴] برای شناسایی مستقیم، ٪۸۸/۳ با فرهنگ لغات ۱۰ کلمه‌ای بوده است). همانگونه که در جدول ۴ مشاهده می‌شود فقدان فرهنگ لغات، نرخ شناسایی را از ٪۹۶/۸۰ به ٪۸۶/۱۷ کاهش داده است. در واقع یک حسن روشهای تمیز دهنده مسیر نسبت به روشهای تمیز دهنده مدل همین وابستگی کمتر به فرهنگ لغات است. این نتیجه همچنین توانایی مراحل پیش پردازش، تقطیع، استخراج ویژگیها، و منطق الگوریتم ویتربی ارائه شده در این مقاله را برای استفاده در یک سیستم شناسایی حروف کلاسیک (مستقل از فرهنگ لغات) نشان می‌دهد.

جدول (۴) درصد صحت مرحله شناسایی برای آزمایشهای مختلف.

تعداد تکرار الگوریتم ویتربی	TST1 بدون فرهنگ لغات	TST1 با فرهنگ لغات	TST2 با فرهنگ لغات	TST3 با فرهنگ لغات
۵	٪۶۷/۰۲	٪۸۸/۶۵	٪۶۹/۰۰	٪۵۲/۳۸
۱۰	٪۷۸/۸۲	٪۹۴/۶۸	٪۸۰/۰۰	٪۷۶/۱۹
۲۰	٪۸۶/۱۷	٪۹۶/۸۰	٪۹۱/۰۰	٪۹۰/۴۸

همانگونه که ذکر شد مجموعه آزمایشی TST1 از کلمات نوشته شده توسط نویسندگان هابی انتخاب شده است که در مجموعه آموزشی موجود هستند. لذا برای بررسی میزان حساسیت سیستم به دستخط نویسنده، آزمایش با مجموعه آزمایشی TST2 که شامل دستخط نویسنده هابی است که سیستم در مرحله آموزش ندیده است (دستخط ناشناس) نیز تکرار شد. نتایج حاصل در ستون چهارم جدول ۴ نشان داده شده است. این نتایج نشان می‌دهد که افت عملکرد سیستم برای دستخطهای ناشناخته نسبت به دستخطهای مشاهده شده به اندازه ۵/۸٪ بوده است که قابل قبول می‌باشد.

برای آزمایش عملکرد سیستم بر روی کلماتی که در مجموعه آموزشی موجود نبوده اند (کلمات ناشناس)، مجموعه آزمایشی TST3 بکار رفته است که نتیجه شناسایی برای آن در ستون پنجم جدول ۴ نشان داده شده است. این کلمات با آمار بدست آمده از فرهنگ لغات نامتناقض هستند. در صورتیکه چنین تناقضی وجود داشته باشد (یعنی مثلاً یکی از انتقال حالات، حالات اولیه، یا حالات انتهایی برای این کلمات، در آمار حاصل از فرهنگ لغات دارای احتمال صفر باشد یا آمار مربوطه در مجموعه آزمایشی و آموزشی شدیداً متضاد باشند) سیستم قادر به شناسایی کلمه نخواهد بود. در این آزمایش هم فقدان آمار فرهنگ لغات و هم فقدان آموزش تصویری وجود دارد. اما با این وجود نتایج همچنان قابل قبول هستند. یک دلیل مهم برای این رفتار مناسب، انتخاب مجموعه کلمات آموزشی به نحویکه ذکر شد می‌باشد. یعنی در نظر داشتن تمام حروف فارسی و ترکیبات متنوع آنها در کلمات نستعلیق تحریری انتخاب شده تأثیر مثبت خود را در شناسایی کلمات ناشناس نشان داده است. این امر مشکل فقدان آموزش تصویری را تقریباً مرتفع می‌سازد. بنابراین برای توسعه دادن این سیستم، می‌توان تنها تعداد نویسندگان این مجموعه ۵۰ کلمه‌ای را هرچه بیشتر نمود و از این نمونه‌ها برای آموزش تصاویر استفاده کرد، در حالیکه فرهنگ لغات مورد استفاده مستقلاً و بنابر کاربرد (با توجه به ویژگی جابجایی پذیری سیستم) می‌تواند انتخاب گردد.

در نتیجه در صورتیکه آموزش مبتنی بر تصویر سیستم با تعداد زیادی از نمونه تصاویر این ۵۰ کلمه انجام شود، سیستم قادر خواهد بود کلمات ورودی به سبک نستعلیق و متعلق به فرهنگ لغات دلخواه را شناسایی کند. اگر در کاربرد مورد نظر فرهنگ لغات خاصی مد نظر باشد، می‌توان از آمار فرهنگ لغات آن کاربرد بعنوان آمار فرهنگ لغات سیستم استفاده نمود که دقت شناسایی را بالا می‌برد. اما در صورتیکه فرهنگ لغات خاصی مد نظر نباشد می‌توان اثر فرهنگ لغات را با برابر قرار دادن احتمالاتی مربوط به آمار فرهنگ لغات حذف نمود، و شناسایی را مستقل از فرهنگ لغات انجام داد. همانطور که دیدیم در آزمایش آخر که این وضعیت برقرار بود عملکرد قابل قبولی بدست آمد. این بدان معناست که سیستم محدود به شناسایی ۵۰ کلمه مذکور نیست و تنها برای شناسایی این ۵۰ کلمه طراحی نشده است.

سرعت عملکرد کل سیستم بر روی یک کامپیوتر شخصی با پردازنده Pentium II-533MHz بطور متوسط ۱/۰۹ کلمه بر ثانیه یا ۴/۶ حرف بر ثانیه بوده است.

۸- جمع بندی

در این مقاله از یک مدل پنهان مارکف با چگالی مشاهدات پیوسته و طول حالات متغیر برای شناسایی کلمات دستنویس فارسی که ذاتاً پیوسته هستند استفاده شد. در مرحله پیش پردازش روشهای جدیدی برای حذف سرکشها و زیرکشهایی که ترتیب راست به چپ قطعه‌ها را برهم می‌زنند ارائه شد که نتایج مناسبی بر روی تصاویر آزمایشی داشته‌اند، و تنها روش حذف سرکش دارای خطایی در حدود ۲/۵٪ بوده است. همچنین دو روش تقطیع بهبود یافته ارائه شد که روش اول پس از یافتن اجزاء منظم نهایی تقطیع را در آن محلها انجام می‌دهد، و روش دوم مینیمم‌های محلی مرز بالایی، محلهای همپوشانی، و محلهای اتصال "ر" به حرف قبلش را یافته و از میان آنها موارد مناسب را بعنوان محل تقطیع در نظر می‌گیرد. درصد صحت عملکرد این روشها برای برآورده کردن معیار پرهیز از خطای عدم تقطیع بر روی تصاویر آزمایشی به ترتیب ۷۷/۶۲٪ و ۹۵/۶۸٪ بوده است. بنابراین روش دوم مورد استفاده قرار گرفته است. بردار ویژگی بکار رفته دارای قابلیت ایجاد تمایز بالا (کارایی ۱۷/۹۸٪ در پنج انتخاب اول)، خصوصیت نامتغیر بودن نسبت به تغییر اندازه و انتقال و طول کم (که این امر سرعت عملیات را در مرحله شناسایی بطور مؤثری افزایش می‌دهد) بوده و پیچیدگی محاسبه این ویژگیها پایین است.

مدل پنهان مارکف دارای این مزیت است که مسأله شناسایی کلمات را در یک چارچوب احتمالاتی خوش تعریف و قدرتمند تعریف می‌نماید. بنابراین می‌توان با حفظ این چارچوب، تغییرات لازم را برای بهبود عملکرد سیستم اعمال نمود.

ویژگی پیوسته بودن چگالی مشاهدات، با حذف خطاهای کوانتیزاسیون، دقت عملکرد را افزایش می‌دهد، و استفاده از دسته بندی کنندهٔ مجموع توابع گاسی این ظرفیت را در اختیار قرار می‌دهد که بتوان برای هر حالت شکل‌های نوشتاری مختلف در نظر گرفت. بنابراین می‌توان با این ساختار، سبک‌های نوشتاری کلمات آموزشی و آزمایشی را بصورت آزادتر هم در نظر گرفت. ما برای حالات، زیر حالت‌هایی با شکل‌های کاملاً متفاوت در نظر گرفتیم (مثلاً شکل‌های مختلف حرف "ع") که نتیجه کار کاملاً مناسب بود. ویژگی متغیر بودن طول حالات به خوبی تقطیع زیادی انجام شده در مرحله تقطیع را جبران می‌نماید.

در روش‌های شناسایی کلمات مبتنی بر مدل پنهان مارکف همیشه یک مجموعه لغات آشنا، یا فرهنگ لغات بکار می‌رود. روش‌های جداکننده با مدل کاملاً وابسته به فرهنگ لغات می‌باشند. در این سیستم‌ها نتیجه شناسایی یا یکی از این لغات است و یا شناسایی ناموفق اعلام می‌شود. اما در روش‌های جداکننده با مسیر، که روش ما نیز از جمله همین روش‌هاست، تنها بخش آموزش فرهنگ لغات دارای این وابستگی می‌باشد. در آزمایش‌ها مشاهده شد که سیستم برای کلمات خارج از فرهنگ لغات نیز می‌تواند جوابگو باشد، به شرطی که ترتیب حروف آنها با آمار استخراج شده در پارامترهای (Π, A, Γ) متناقض نباشد.

ضمیمه - الگوریتم ویتربی اصلاح شده

در این الگوریتم، T تعداد قطعه‌های کلمه جاری، N تعداد حالات، و L تعداد بهترین مسیرهای سراسری است که باید بدست آید.

- ساختمان داده‌های مورد نیاز (Storage):

t	: time index
c	: iteration index
$\Psi_t(i, l), 1 \leq t \leq T, 1 \leq i \leq N, 1 \leq l \leq L$: survivor (i^*, l^*, d^*) terminating in i_t
$\delta_t(i, l), 1 \leq t \leq T, 1 \leq i \leq N, 1 \leq l \leq L$: survivor score in i_t
$count(i, t), 1 \leq i \leq N, 1 \leq t \leq T$: count of passes allowed at node i_t

۱- مقدار دهی اولیه (Initialization):

$$k = \arg \max_j \{b_{i_j}(o_1)\}$$

$$\delta_1(i, 1) = \pi_i b_{i_k}(o_1) \cdot P(d = 1 | q_{i_k}) \quad \text{for } 1 \leq i \leq N$$

$$\delta_1(i, l) = 0 \quad \text{for } 1 \leq i \leq N, 2 \leq l \leq L$$

$$\Psi_1(i, l) = (0, 0, 1) \quad \text{for } 1 \leq i \leq N, 1 \leq l \leq L$$

$$count(i, t) = 1 \quad \text{for } 1 \leq i \leq N, 1 \leq t \leq T$$

$$c = 1$$

۲- پیش بازگشت (Pre-recursion):

$$\text{for } 2 \leq t \leq T, 1 \leq j \leq N : \{$$

$$k = \arg \max_i \{b_{j_k}(o_{cur})\}$$

$$\delta_t(j, 1) = \max_{1 \leq i \leq N, 1 \leq d \leq D} [\delta_{t-d}(i, 1) a_{ij} P(d | q_{j_k}) b_{j_k}(O'_{t-d+1})^d]$$

$$\Psi_t(j, 1) = \arg \max_{1 \leq i \leq N, 1 \leq d \leq D} [\delta_{t-d}(i, 1) a_{ij} P(d | q_{j_k}) b_{j_k}(O'_{t-d+1})^d]$$

$$\} \text{ //end of for}$$

۳- حرکت به عقب (Backtracking):

$$P^* = (c - th) \max_{1 \leq i \leq N, 1 \leq l \leq count(i, T)} [\delta_T(i, l) \gamma_i]$$

$$(i_T^*, l_T^*) = \arg(c - th) \max_{1 \leq i \leq N, 1 \leq l \leq count(i, T)} [\delta_T(i, l) \gamma_i]$$

$$count(i_T^*, T) = count(i_T^*, T) + 1$$

که $[\cdot] \max(c - th)$ بیانگر c امین ماکزیمم می‌باشد.

$$d_T^* \leftarrow \frac{d}{\Psi_T(i_T^*, l_T^*)}$$

```

t = T
while (t - d_t^* ≥ 1) do : {
    i_{t-d_t^*}^*, l_{t-d_t^*}^* ←  $\frac{i,l}{d}$  Ψ_t(i_t^*, l_t^*)
    t = t - d_t^*
    d_t^* ←  $\frac{d}{d}$  Ψ_t(i_t^*, l_t^*)
    if (d_t^* < 0)
        break; //go to after the while loop
    count(i_t^*, t) = count(i_t^*, t) + 1
} //end of while

```

اگر c امین دنباله حالات بهینه یا c امین مسیر سراسری بهتر ($I^* = \{i_1^* i_{1+d_1}^* \dots\}$)، شرایط مورد نظر را برقرار کرد (یعنی شناسایی صحیح بود) یا $c > L$ ، کار خاتمه یابد در غیر اینصورت ادامه بده:

۴ - حرکت به جلو (Forward-tracking).

```

while(t ≤ T) {
    j^* = i_t^*
    l = count(j^*, t)
    k = arg max_i {b_{j_i^*}(o_{cur})}
    δ_t(j^*, l) = (l - th) max_{1 ≤ i ≤ N, 1 ≤ d ≤ D, 1 ≤ m ≤ count(i, t-d)} [δ_{t-d}(i, m) a_{ij^*} P(d | q_{j_k^*}) b_{j_k^*}(O_{t-d+1}^t)^d]
    Ψ_t(j^*, l) = arg(l - th) max_{1 ≤ i ≤ N, 1 ≤ d ≤ D, 1 ≤ m ≤ count(i, t-d)} [δ_{t-d}(i, m) a_{ij^*} P(d | q_{j_k^*}) b_{j_k^*}(O_{t-d+1}^t)^d]
} //end of while

```

زیر نویس ها

- | | |
|--|------------------------|
| 1-Hidden Markov Model | 10-Over-segmentation |
| 2-Subword | 11-Bounding Rectangles |
| 3-Doubly Stochastic Process | 12-Regularities |
| 4-Continuous Density Variable Duration HMM | 13-Singularities |
| 5-Viterbi Algorithm | 14-Islands |
| 6-Reestimation | 15-Bridges |
| 7-Baum-Welch | 16-Discrimination |
| 8-Model Discriminant HMM | 17-Portability |
| 9-Path Discriminant HMM | |

مراجع

- [1] A. Amin, "Off-line Arabic Character Recognition: The State of the Art," Pattern Recognition, vol. 31, no. 5, pp. 517-530, 1998.
- [2] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, No. 7, pp. 690-706, 1996.
- [3] M.-Y. Chen, A. Kundu, and J. Zhou, "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network," IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, No. 5, May 1994.
- [4] M.-Y. Chen, A. Kundu and S. N. Srihari, "Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition," IEEE Trans. On Image processing, vol.4, No. 12, Dec. 1995.
- [5] H. Miled, C. Olivier, M. Cheriet and Y. Lecourtier, "Coupling Observation/Letter for a Markovian Modelisation Applied to the Recognition of Arabic Handwriting," IEEE Proc. of 4th Int. Conf. on Document Analysis and Recognition, vol. 2, pp. 580-583, 1997.
- [6] A. El-Yacoubi, M. Gilloux, R. Sabourin and C. Y. Suen, "An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition," IEEE Trans. Pattern Anal. Mach. Intell. Vol. 21, No. 8, Aug. 1999.

- [7] L. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall 1993.
- [8] A. Kundu, Y. He and P. Bahl, "Recognition of handwritten word: first and second order hidden Markov model based approach," *Pattern Recognition* vol. 22, No. 3, pp. 283-297, 1989.
- [9] A. Kundu, Y. He, and M.-Y. Chen, "Alternatives to Variable Duration HMM in Handwriting Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 20, no.11, Nov. 1998.
- [10] A. El-Yacoubi, M. Gilloux, R. Sabourin and C. Y. Suen, "An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 21, No. 8, Aug. 1999.
- [11] H. Miled, C. Olivier, M. Cheriet and Y. Lecourtier, "Coupling Observation/Letter for a Markovian Modelisation Applied to the Recognition of Arabic Handwriting," *IEEE Proc. of 4th Int. Conf. on Document Analysis and Recognition*, vol. 2, pp. 580-583, 1997.
- [12] I. Bazzi, R. Schwartz and J. Makhoul, "An Omnifont Open-Vocabulary OCR System for English and Arabic," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, No. 6, Jun. 1999.
- [13] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar, "Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM," *Pattern Recognition*, vol. 34, pp. 1057-1065, 2001.
- [14] S. Kuo and O. E. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 16, pp. 842-848, 1994.
- [15] N. BenAmara and A. Belaid, "Printed PAW recognition based on planar hidden Markov models," In *13th Int. Conf. On Pattern Recognition*, vol. B, Vienna, Austria, 1996.
- [۱۶] عزمی، رضا، کبیر، احسان... و بدیع، کامبیز. "بکارگیری مدل مخفی مارکف و الگوریتم ویتربی تغییر یافته برای کاهش میزان واژدگی در بازشناسی نهایی زیر کلمات چاپی"، در مجموعه مقالات چهارمین کنفرانس بین‌المللی انجمن کامپیوتر ایران، ص ۲۹۱-۲۹۷، بهمن ۱۳۷۷.
- [17] D. Motawa, A. Amin and R. Sabourin, "Segmentation of Arabic Cursive Script," *Proc. 4th Int'l Conf. Document Analysis and Recognition*, vol. 2, pp. 625-628, 1997.
- [18] J. Wang, M. K. H. Leung and S. C. Hui, "Cursive Word Reference Line Detection," *Pattern Recognition*, vol. 30, No. 3, pp. 503-511, 1997.
- [۱۹] ادیبی، پیمان. شناسایی کلمات دستنویس فارسی با استفاده از مدل پنهان مارکف، پایان نامه کارشناسی ارشد هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، شهریور ۱۳۸۰.
- [20] A. Amin and G. Masini, "Machine Recognition of Multifonts Printed Arabic Texts," *Proc. 8th Int. Conf. On Pattern Recognition*, paris, pp. 392-395, 1986.
- [21] F. El-Khaly and M. Sid-Ahmed, "Machine Recognition of Optically Captured Machine Printed Arabic Text," *Pattern Recognition* 23, pp. 1207-1214, 1990.
- [22] H. Almuallim and S. Yamaguchi, "A Method of Recognition of Arabic Cursive Handwriting," *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-9*, pp. 715-722, 1987.
- [23] A. Amin and S. Al-Fedaghi, "Machine Recognition of Printed Arabic Text Utilising a Natural Language Morphology," *Int'l. J. Man-Machine Stud.* 35, pp. 769-788, 1991.
- [24] T. El-Sheikh and R. Guindi, "Computer Recognition of Arabic Cursive Script," *Pattern Recognition* 21, pp. 293-302, 1988.
- [25] J.C. Simon, "Off-line Cursive Word Recognition," *Proc. IEEE*, p. 1150, July 1992.
- [26] C. Olivier, H. Miled, K. Romeo and Y. Lecourtier, "Segmentation and Coding of Arabic Handwritten Words," *IEEE Proc. of 13th Int. Conf. on Pattern Recognition*, vol. 3, pp. 264-268, 1996.
- [27] O. D. Trier, A. K. Jain and T. Taxt, "Feature Extraction Methods For Character Recognition – A Survey," *Pattern Recognition*, vol. 29, No. 4, pp. 641-662, 1996.
- [28] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis and Machine Vision*, London:Champion & Hall, 1993.
- [29] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd Ed., Massachusetts: Addison-Wesley, 1987.
- [30] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Ed., Boston:Academic Press, 1990.
- [۳۱] فضایی، حبیب‌الله. *تعلیم خط، چاپ ششم، تهران: انتشارات سروش، ۱۳۷۰.*
- [۳۲] صفابخش، رضا. *بینایی ماشین. جزوه درسی، دانشگاه صنعتی امیرکبیر، ۱۳۷۸.*
- [۳۳] شایگان، محمدمین. *شناسایی متون دستنویس فارسی با روشهای آماری، پایان نامه کارشناسی ارشد هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، بهمن ۱۳۷۳.*
- [34] E. Kabir, G. R. Ardeshir, "Structural Recognition of Hand-Printed Farsi Characters", *Proc 3rd Iranian Conf. Elect. Eng. ICEE'95*, pp.295-300, Tehran, 1995. (In Farsi)
- [35] R. Azmi, E. Kabir, "Character Segmentation for Printed Farsi Text", *5th Iranian Conf. Elect. Eng., ICEE'97*, pp.302-309, Tehran, 1997. (In Farsi)
- [36] R. Azmi, E. Kabir, "A New Segmentation Technique for Omni-Font Farsi Text", *Esteghlal Journal of Engineering, Esfahan Univ. of Technology*, vol.18, no.2, pp.1-10, 2000. (In Farsi)
- [37] R. Azmi, E. Kabir, "A New Segmentation Technique for Omni-Font Farsi Text", *Pattern Recognition Letters*, vol.22, no.2, pp.97-104, 2001.

[38] A. R. Dehghani, and M. A. Masnadi-Shirazi, "Isolated handwritten Persian character recognition using multiple hidden Markov models," Proc. 1st Intl. Conf. Machin Vision and Image Processing, pp. 100-105, Birjand Univ., March 2001.

[۳۹] مسروری، کیوان ، کبیر، احسان‌اله ، و بدیع، کامبیز، "شناسایی کلمات دستنویس با استفاده از الگوریتم DTW تعمیم یافته"، در مجموعه مقالات هفتمین کنفرانس مهندسی برق ایران، ص ۵۹-۶۶، ۱۳۷۸.

[40] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar, "Unconstrained Farsi Handwritten word recognition using Fuzzy Vector Quantization and Hidden Markov Models," Pattern Recognition Letters, vol. 22, no. 2 pp. 209-214, 2001.

[۴۱] نظام آبادی پور، ح. ، کبیر، ا. ، و عزمی، ر. ، "جداسازی حروف در متون چاپی قدیمی"، مجموعه مقالات ششمین کنفرانس سالانه انجمن کامپیوتر ایران، اصفهان، ص ۵۷۹-۵۹۰، اسفند ماه ۱۳۷۹.