یک چارچوب توزیعی مبتنی بر خوشه‌بندی دومرحله‌ای برای شناسایی چهره در مقیاس بالا

سید محمد احمدی* و روح الله دیانت

گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشکده فنی و مهندسی، دانشگاه قم، قم، ایران

چکیده

شناسایی چهره از پرکاربردترین حوزه‌های بینایی ماشین در دهه اخیر محسوب می‌شود. با قدرت‌مندترشدن مدل‌های هوشمند در زمینه بازشناسی چهره، استفاده عملی از آن در کاربردهای با تعداد افراد بسیار بالا رایج شده‌است؛ اما افزایش تعداد افراد در شناسایی چهره، یک چالش عمده محسوب می‌شود که به نوبه خود به سه زیرچالش دیگر قابل تحویل است: زیرچالش افت دقت؛ زیرچالش افزایش نیازمندی به حافظه؛ و زیرچالش افزایش پیچیدگی زمانی. به‌منظور حل این چالش‌ها، راه‌حل‌های گوناگونی ارائه شده‌اند. برخی از راه‌حل‌ها به تقویت عناصر موجود در شبکه‌های عصبی عمیق، همچون تابع هزینه و یا ارائه بلاک‌های جدید در شبکه‌ها پرداخته‌اند. دسته دیگری از راه‌حل‌ها، به‌خصوص برای حل چالش‌های حافظه و زمان، به راه‌حل‌های توزیعی پناه برده‌اند. این پژوهش، یک رویکرد دومرحله‌ای توزیعی معرفی کرده و مدعی است که هر سه چالش را به‌صورت همزمان، تا حد خوبی، مرتفع کرده است. روش پیشنهادی، دربرگیرنده سه واحد است: واحد زیرشبکه‌ها، واحد خوشه‌یاب، و واحد تصمیم‌گیر نهایی. یکی از تفاوت‌های روش پیشنهادی با روش‌های توزیعی موجود، این است که بر خلاف سایر روش‌ها که توزیع دسته‌ها در زیرشبکه‌ها به‌صورت تصادفی انجام می‌شوند، روش ارائه‌شده، از خوشه‌بندی به‌عنوان روش توزیع مسئله به زیرشبکه‌ها استفاده می‌کند. هر زیرشبکه، یک شبکه عصبی عمیق نظارتی است که با داده‌های آموزشی مربوط به دسته‌های خود آموزش می‌بیند. واحد خوشه‌یاب، شباهت بردارهای ویژگی داده‌های آزمون را با میانگین بردارهای ویژگی دسته‌ها مقایسه و بهترین خوشه را پیدا می‌کند؛ درنهایت، واحد تصمیم‌گیر نهایی با ترکیب نتایج دو واحد قبلی، بهترین دسته را انتخاب می‌کند. نتایج نشان می‌دهد که روش پیشنهادی، در مقایسه با روش‌های مشابه، از نظر صحت، بازخوانی، و امتیاز F1 عملکرد بهتری دارد. این روش ضمن سریع‌تربودن، دارای دقت بالاتری نسبت به روش‌های بدون توزیع است و در مقایسه با روش‌های توزیعی تصادفی، سرعتی برابر و دقتی بالاتر دارد. آزمایش‌ها بر روی تعدادی از مشهورترین و معتبرترین مجموعه‌دادگان حوزه شناسایی چهره اجرا شده‌است، همچون VGGFace2و MS-Celeb-1M و Glint360K که هر یک شامل تصاویر تعداد زیادی از افراد هستند. نتایج پیاده‌سازی نشان می‌دهند که روش پیشنهادی، علاوه‌بر عملکرد بهتر، مقیاس‌پذیری بالاتری را در شناسایی چهره دارد.

واژگان کلیدی: بازشناسی چهره، شناسایی چهره، خوشه‌بندی، یادگیری عمیق، یادگیری توزیعی

A two-stage clustering-based distributed framework for large-scale face identification

Sayed Mohammad Ahmadi* & Rouhollah Dianat

Department of Computer Engineering and Information Technology, Faculty of Engineering, University of Qom, Qom, Iran.

Abstract

Face recognition can be divided into two types: identification and verification of faces. Face identification is one of the most prevalent fields in computer vision in recent decades. With the enhancement of intelligent models in face recognition, its practical use has become widespread, especially in applications involving a large number of individuals. Compared to other image processing tasks, such as object detection, face identification poses greater challenges. In object detection, the categories that need to be distinguished are distinct enough from each other, but in face identification,

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۴۰۱/۱۲/۱۶ • تاریخ پذیرش: ۱۴۰۲/۱۲/۵ • تاریخ انتشار: ۱۴۰۳/۵/۱۰ • نوع مطالعه: پژوهشی

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

all the categories to be identified have a similar oval structure, and the identification system must focus on the details of the face images. In addition to this complexity, the increase in the number of individuals in face identification poses a major challenge, leading to three additional sub-challenges: accuracy degradation, increased memory requirements, and increased temporal complexity. Various solutions have been proposed to address these challenges. Some solutions focus on strengthening elements within deep neural networks, such as loss functions or introducing new blocks in the networks. Another category of solutions, particularly for memory and time complexity challenges, resort to distributed approaches. Many methods have fallen short of addressing one or two additional challenges. This study introduces a two-stage distributed approach claiming to effectively tackle all three challenges simultaneously. The proposed method comprises three units: a *sub-nets* unit, a *clustering* unit, and a *final decision-making* unit. A distinguishing feature of the proposed method from existing distributed approaches is its utilization of clustering as a distribution method to *sub-nets* unit. Each *sub-net* is a supervised deep neural network trained with training data relevant to its classes. The advantage of using clustering instead of random distribution is that with high confidence, the cluster containing the main category can be found by comparing the feature vector of the test image with the representative features of all clusters. Then, the identification operation is performed on the desired cluster, which contains fewer categories. Therefore, the clustering unit compares the similarity of feature vectors of test data with the average feature vectors of the classes and finds the best cluster. This is not possible in methods based on random distribution. In other words, no entity from each sub-net can provide a good and suitable representation of the members of the sub-net. The *clustering* unit compares feature vectors of test data with the average feature vectors of classes to find the best cluster. Ultimately, the *final decision-making* unit selects the best class by combining the results of the two preceding units. Results indicate that the proposed method outperforms similar methods in terms of accuracy, recall, and F1 score. This method is not only faster but also more accurate than non-distributed methods, and compared to randomly distributed methods, it offers comparable speed and higher accuracy. Experiments were conducted on several well-known and reputable face recognition datasets such as VGGFace2, MS-Celeb-1M, and Glint360K, each containing images of numerous individuals. Implementation results demonstrate that the proposed method exhibits superior performance and scalability in face recognition.

Keywords: face identification, clustering, deep learning, distributed learning.

۱- مقدمه

بازشناسی چهره، یکی از کاربردهای مهم بینایی ماشین [۱-۵] ، به هر دو شکل شناسایی چهره و یا تصدیق چهره، اطلاق می‌شود. در تصدیق چهره، دو تصویر چهره به سامانه داده می‌شود. سامانه باید تشخیص دهد که آیا این دو چهره، مربوط به یک شخص است یا دو شخص متفاوت. در شناسایی چهره، یک تصویر ورودی به سامانه داده می‌شود. سامانه باید هویت صاحب چهره را مشخص نمایند. تصدیق چهره، یک مسئله دسته‌بندی دودسته‌ای است، در حالی که شناسایی چهره، یک مسئله دسته‌بندی N دسته‌ای است که N تعداد افراد موجود در سامانه شناسایی است. بیش از یک دهه است که روش‌های بازشناسی چهره عمدتاً مبتنی بر یادگیری عمیق، و به‌صورت خاص، شبکه‌های عصبی کانولوشنی ارائه می‌شوند. بازشناسی چهره، با موضوعات دیگری در حوزه پردازش چهره در ارتباط است؛ از قبیل تشخیص چهره[۱] [۶-۹]، بازشناسایی شخص[۲] [۱۰]، ترازبندی چهره[۳] [۱۱-۱۵]، کشف ضد جعل چهره (برای تشخیص تصویر زنده از غیر آن) [۱۶-۱۸]، تکثیر داده‌ها [۱, ۱۹]، تحلیل چهره مظنون[۴]، محلی‌سازی ویژگی‌های چهره[۵]، مدل‌سازی چهره[۶]، تحلیل حالات چهره[۷]، طبقه‌بندی احساسات، بازشناسی ژست[۸] چهره، تخمین سن، دسته‌بندی جنسیت و ...[۲۰]. بازشناسی چهره، یکی از روش‌های مشهور تأیید هویت زیست‌سنجی[۹] بوده و در کاربردهای متنوعی، همچون تجاری و حقوقی و قانونی و نظامی و ... قابل استفاده است. با توجه به شیوع گسترده ویروس Covid19 در سال‌های اخیر، مسئله تأیید هویت زیست‌سنجی با کمترین دخالت اعضای فیزیکی انسان، بسیار دارای اهمیت شده است [۲۱, ۲۲]. از این جنبه، استفاده گسترده از سامانه‌های بازشناسی چهره، به‌عنوان تأیید هویت، بسیار مورد تأکید است. بسیاری از کاربردهای پردازش تصویر چالش‌های مشترکی با کاربرد شناسایی چهره دارند. از آن جمله می‌توان به تفاوت شرایط نوری (همچون میزان روشنایی[۱۰] و تقابل[۱۱]) اشاره کرد. همچنین چالش‌های مشابهی بین شناسایی چهره و سایر کاربردهای پردازش چهره وجود دارد؛ مانند زوایای متفاوت قرارگیری چهره یک فرد در تصاویر گوناگون، ویژگی‌های ظاهری متفاوت شخص (همچون ریش و آرایش)، پوشش‌های گوناگون فرد (مانند حجاب یا کلاه) اشاره کرد. علاوه‌بر چالش‌های مشترک یادشده، خود چهره، دارای یک ویژگی

[1] Face detection
[2] Person reidentification
[3] Face Alignment
[4] Suspect face analysis
[5] Face feature localization
[6] Face modeling
[7] Face expression analysis
[8] Gesture
[9] Biometric
[10] Brightness
[11] Contrast

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]

خاصی است که چالش جدیدی را موجب می‌شود. گاهی فاصله درون‌دسته‌ای اعضای درون یک دسته، بیشتر از فاصله برون‌دسته‌ای اعضای دسته‌های گوناگون می‌شود [۲۳].

بر خلاف مسائل دسته‌بندی دیگری همچون تشخیص اشیا که اختلاف اشیا به شکل به‌نسبه بارزی مشهود و قابل درک است (برای مثال تشخیص ماهی از انسان و از دوچرخه و از اتومبیل و از توپ و ... به‌آسانی قابل انجام است)، اما در مسئله شناسایی چهره، به‌دلیل ساختار واحد همه چهره‌ها که حالت تقریباً بیضوی دارند و این که همه چهره‌ها دارای اجزای یکسان در موقعیت‌های تقریباً یکسانی هستند، بسیار اتفاق می‌افتد که حتی چشم انسان در تشخیص یکسانی هویت دو تصویر از یک شخص واحد در موقعیت‌ها و شرایط گوناگون دچار اشتباه می‌شود. و یا هویت تصاویر اشخاص گوناگون، به اشتباه، یکسان فهمیده می‌شود. به دلیل همین چالش‌های عمومی و ویژگی خاص چهره، هر چه تعداد افراد (دسته‌ها) در مسئله شناسایی چهره، بیشتر باشد، سه اشکال کلی به وجود می‌آید: اشکال در ذخیره‌سازی؛ افزایش پیچیدگی زمانی؛ و افزایش شیب افت دقت. بنابراین به‌صورت کلی، پرسش اساسی این مطالعه، از این قرار است: چگونه می‌توان راه‌حل‌هایی برای غلبه بر این مشکلات ارائه کرد؟

در شناسایی چهره، با توجه به این که اغلب به‌هنگام انطباق چهره، از یک لایه دسته‌بند استفاده می‌شود که به‌صورت تمامی متصل با لایه پیش از خود در ارتباط است، ماتریس وزن مربوط به این بخش، ابعاد بسیار بالایی خواهد داشت. نگهداری این ماتریس وزن در حافظه با دسترسی تصادفی[1]، نیاز به حافظه بالایی دارد که گاهی اوقات، خارج از محدوده حافظه‌های GPU فعلی است؛ علاوه‌بر این، آموزش یک شبکه بسیار بزرگ، از نظر زمانی بسیار طولانی است. از سوی دیگر، با توجه به نکته‌ای که در مورد ماهیت خاص چهره بیان شد، افزایش هر چه بیشتر تعداد دسته‌ها، افت دقت بیشتری را نیز به همراه دارد. همچنین زمانی که همه دسته‌ها در کنار یکدیگر آموزش داده و نیز مورد شناسایی واقع می‌شوند، پراکندگی دسته‌ها بسیار زیاد است و تمایز میان دسته‌های دشوار و به‌نسبه شبیه، با احتمال بیشتری شبکه را دچار خطا می‌کنند. شکل (۱)

برای غلبه بر این چالش‌ها، همان‌گونه که به‌تفصیل در بخش پیشینه توضیح داده خواهد شد، دو رویکرد کلی اتخاذ شده‌است. در رویکرد نخست، تلاش می‌شود معماری‌های جدید و یا توابع هزینه جدیدی معرفی شوند که تفکیک‌پذیری بیشتری داشته باشد و لذا افزایش تعداد دسته‌ها آسیب کمتری به دقت شناسایی وارد کند. از جمله این تلاش‌ها می‌توان به L-Softmax [۲۴]، A-Softmax [۲۵]، NormFace [۲۶]، CosFace [۲۷]، ArcFace [۲۸] اشاره کرد. این تلاش‌ها، از نظر تئوری بسیار ارزنده هستند. اما درعمل، برای تعداد دسته‌های بسیار بالا، امکان پیاده‌سازی ندارند و به‌ناچار باید با رویکرد دوم ترکیب شوند. برای امکان‌پذیرکردن پیاده‌سازی برای تعداد دسته‌های بسیار بالا، از تکنیک‌هایی همچون تجزیه سافت‌مکس[۲۹]، انتخاب دسته‌های فعال[۳۰] و یا نمونه‌برداری[۳۱] بهره‌برداری شده‌است و البته خود مقالات، تأکید کرده‌اند که این کار موجب اندکی افت دقت می‌شود.

رویکرد دوم تلاش می‌کند که از طریق روش تقسیم و حل، مسئله را به زیرمسئله‌های کوچک‌تری تجزیه کند. ساده‌ترین و سرراست‌ترین روشی که مطابق این رویکرد پیشنهاد شده و مورد توجه مطالعه حاضر نیز هست، مدل سافت‌مکس مستقل[۳۲] است. با این ایده ساده که تعداد بسیار زیاد دسته‌ها، به تعدادی زیرشبکه، به‌صورت تصادفی، توزیع می‌شود. هر زیرشبکه، به‌صورت مستقل آموزش داده و سپس نتایج زیرشبکه‌ها با یکدیگر ترکیب می‌شوند. رویکرد تصادفی در توزیع دسته‌ها به زیرشبکه‌ها، موجب بروز نوعی خطا می‌شود که در بخش روش پیشنهادی بیشتر توضیح داده خواهد شد؛ درنتیجه، مطالعه حاضر، توزیع هوشمند را از طریق خوشه‌بندی پیشنهاد می‌کند. به‌کارگیری الگوریتم‌های خوشه‌بندی برای توزیع دسته‌ها، موجب می‌شود که در درون هر خوشه، دسته‌های مشابه با هم قرار داشته باشند؛ ازاین‌رو، پراکندگی داده‌ها در هر خوشه، بسیار کم‌تر از پراکندگی مجموع داده‌ها در حالت بدون توزیع است. هر زیرشبکه، با تمرکز بیشتری بر روی داده‌های شبیه به هم آموزش می‌بیند و لذا انتظار می‌رود که دقت نهایی، افزایش یابد. شکل (۱) ضمن این که توزیع دسته‌ها بین زیرشبکه‌های گوناگون، به‌صورت تصادفی، کاهش پراکندگی داده‌ها در هر زیرشبکه را تضمین نمی‌کند.

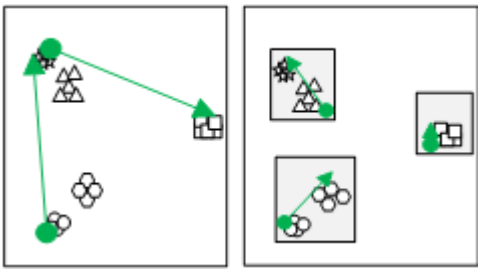
روش پیشنهادی مقاله، شامل سه واحد است: الف) زیرشبکه‌ها؛ ب) خوشه‌یاب؛ و ج) تصمیم‌گیر نهایی. واحد زیرشبکه‌ها از شبکه‌های عصبی نظارتی برای طبقه‌بندی تصاویر چهره به N/m دسته استفاده می‌کنند (که N تعداد کل دسته‌ها و m تعداد زیرشبکه‌ها است). واحد خوشه‌یاب، از طریق محاسبه میزان شباهت داده آزمون ورودی با هر خوشه، مسئول یافتن بهترین خوشه است؛ یعنی خوشه‌ای که احتمال تعلق داده آزمون ورودی

[1] Random Access Memory (RAM)

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

در آن بیشتر باشد. تصمیم‌گیر نهایی بهترین دسته را از خوشه منتخب برمی‌گزینند.



(شکل-۱) الف راست: میزان پراکندگی دسته‌ها، در هر خوشه، پس از خوشه‌بندی؛ ب چپ: میزان پراکندگی دسته‌ها، پیش از خوشه‌بندی. خطوط سبز، میزان پراکندگی داده‌ها را در هر شبکه (زیرشبکه) نشان می‌دهند.

(Figure-1) (a: right) data dispersion within each cluster after clustering; (b: left) data dispersion before clustering. The green lines show the amount of scatter in the data.

رویکرد اتخاذشده این مقاله، نسبت به روش‌های مشابه پیش‌تر از خود، این مزیت را دارد که علاوه‌بر کاهش هزینه‌های ذخیره‌سازی و محاسباتی، از افت دقت نیز جلوگیری می‌کند و حتی موجب بهبود دقت می‌شود. برخلاف روش‌های دیگر، به‌جای توزیع تصادفی دسته‌ها میان زیرشبکه‌ها، از خوشه‌بندی برای توزیع دسته‌ها میان زیرشبکه‌ها استفاده شده‌است.

مشارکت اصلی نویسندگان در این مقاله را می‌توان به‌صورت زیر خلاصه کرد:

- پیشنهاد یک روش جدید برای ارتقای دقت در شناسایی چهره با ترکیب الگوریتم‌های خوشه‌بندی و مدل‌های یادگیری عمیق.
- ارزیابی روش پیشنهادی با مجموعه‌دادگان معتبر و رایج در حوزه بازشناسی چهره (VGGFace2 [۳۳] و MS-Celeb-1M [۳۴] و Glint360K [۳۱]).
- ارائه تحلیل جامع بر روی نتایج آزمایش‌ها با تأکید بر مقایسه با روش‌های رقیب.

۲- کارهای مرتبط

تعداد زیادی از پژوهش‌گران، تلاش کرده‌اند تا با ارتقا و ارائه توابع زیان جدیدتر و بهتر، عملکرد شبکه را آن‌چنان دقیق و قوی سازند (به‌عبارت دیگر، شبکه دارای تفکیک‌پذیری بسیار بالایی باشد) تا بتوانند تعداد زیادی از دسته‌ها را با دقتی بسیار بالا، بازشناسی کنند. از سوی دیگر، برخی پژوهش‌گران برای امکان پیاده‌سازی سامانه شناسایی چهره برای تعداد دسته‌های بالا، در صدد ارائه راه‌حل‌های توزیعی شدند.

رویکرد نخست: تقویت توابع هزینه

این رویکرد، خود به دو دسته کلی تقسیم می‌شود:

۱) تعدادی از روش‌ها، بر روی مسئله دسته‌بندی[1] تمرکز کرده‌اند و با ارائه یک تابع زیان جدید (اغلب مبتنی بر تابع انتروپی متقاطع[2] بر روی لایه سافت‌مکس یا به بیان خلاصه، تابع زیان سافت‌مکس) تلاش کرده‌اند که به حل مسئله و چالش بپردازند.

تابع زیان سافت‌مکس، در خصوص فاصله برون‌دسته‌ای شرایط نسبتاً مناسبی دارد، اما در زمینه فاصله درون‌دسته‌ای خوب عمل نمی‌کند.[۲۶] به همین دلیل، روش‌های متعددی ارائه شدند که ویژگی‌ها را به‌گونه‌ای تنظیم کنند که فاصله درون‌دسته‌ای از فاصله برون‌دسته‌ای کمتر باشد. یک راه‌کار مناسب برای انجام این کار، نرمال‌سازی ویژگی‌ها و وزن‌ها (مانند تابع زیان Norm-Face)[۲۶] و به‌کارگیری مفهوم «زاویه» (مانند تابع زیان A-Softmax)[۲۵] برای تنظیم هدف ذکر شده بود. به علاوه، دخالت مفهوم «حاشیه[3]»در بعضی روش‌ها مانند L-Softmax)[۲۴]، موجب تقویت هر چه بیشتر عملکرد سامانه می‌شود. در برخی دیگر از روش‌ها، فرض می‌کنند حاشیه و زاویه رابطه جمعی با هم دارند. روش Cos-Face [۲۷] حاشیه جمعی را در فضای کسینوس تعریف می‌کند. در مقابل، ArcFace [۲۸] حاشیه زاویه‌ای جمعی دارد.

۲) برخی روش‌ها نیز مسئله بازشناسی چهره را در قالب «تصدیق چهره» در نظر گرفته‌اند. و مدل‌هایی ارائه کرده‌اند که در آن‌ها همزمان دو یا سه تصویر به‌عنوان ورودی (مانند تابع زیان تقابلی[4][۳۵] و تابع زیان سه‌گانه[5][۳۶])، به دو یا سه یا چند شبکه (که از نظر معماری به‌طورکامل یکسان هستند و تنها تفاوتشان در وزن‌ها است) داده می‌شود. هر شبکه یک بردار ویژگی تولید می‌کند. این روش‌ها تلاش می‌کنند که بردارهای ویژگی تصاویر همسان (مربوط به یک هویت) را به یکدیگر نزدیک کرده و بردارهای ویژگی تصاویر ناهمسان (مربوط به دو هویت) را از یکدیگر دور کنند. با توجه به این که در کابردهای با تعداد بالای دسته‌ها، تعداد انتخاب‌های تصاویر دوگانه و سه‌گانه، به بی‌نهایت میل می‌کند، تابع زیان مرکزی[6][۳۷] و توابع آن، تلاش دارند که فاصله تصاویر هر دسته را با مرکز همان دسته کاهش دهند.

[1] Classification
[2] Cross entropy
[3] Margin
[4] Contrastive Loss
[5] Triplet Loss
[6] Center Loss

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]

رویکرد دوم: تجزیه یا کاهش مسئله

الف) اصلاح و ساده‌سازی تابع هزینه

با بالارفتن تعداد دسته‌ها در حد چندین میلیون یا حتی چندصدهزار، ابزارهای فعلی حافظه‌ای و پردازشی، پاسخ‌گوی ذخیره‌سازی و اجرای عملیات مربوط به بازشناسی چهره نیستند؛ بنابراین راه‌کارهایی ارائه شدند که اصل امکان پیاده‌سازی سامانه‌های بازشناسی چهره را عملی سازند.

در این راستا، چند روش معرفی شدند:

۱) با توجه به این که برای هر نمونه ورودی، تنها یک دسته (یا تعداد خیلی محدودی از دسته‌ها) پاسخ لایه خروجی بالایی دارند، می‌توان این دسته‌ها را دسته‌های فعال برای این نمونه خاص نامید. روش ارائه‌شده در [۳۰] سعی می‌کند برای هر نمونه، ابتدا دسته‌های فعال را پیدا کرده و آموزش را در همان دسته‌ها متمرکز کند. از سوی دیگر با توجه به این که می‌توان بردارهای وزن را به‌عنوان مراکز دسته تفسیر کرد، این روش، پیشنهاد می‌کند به‌جای ذخیره‌سازی و انجام عملیات روی کل بردارهای وزن، تنها روی بردارهای وزن متناظر با دسته‌های فعال کار شود. این روش، سرعت و مصرف حافظه را بسیار تقویت می‌کند و در مقایسه با حالت استفاده کامل از همه دسته‌ها، افت بسیار کمی در دقت دارد. یافتن دسته‌های فعال، از طریق عادی بسیار زمان‌بر است. برای حل این مشکل، پیشنهاد شده‌است که از روش جنگل درهم تصادفی استفاده شود؛ اما یافتن دسته‌های فعال، حتی با کمک جنگل درهم تصادفی و پیچیدگی‌های مرتبط با به‌روزرسانی آن، جزء مشکلات این روش محسوب می‌شود.

۲) تابع زیان ArcFace [۲۸]، برای امکان‌پذیرکردن پیاده‌سازی برای مسئله با دسته‌های بسیار بالا، به سادگی، ماتریس وزن را که بسیار حجیم است، به تعدادی زیرماتریس تقسیم کرده (به‌ترتیب) و هر زیرماتریس را در یک GPU ذخیره می‌کند. داده‌های ورودی به همه GPU ها وارد می‌شوند و عملیات به‌صورت موازی روی همه GPU ها انجام می‌شود. با این روش، عملیات آموزش یک میلیون دسته، که بدون موازی‌سازی امکان‌پذیر نبود، با استفاده از هشت عدد GPU امکان‌پذیر شده است. در این روش، بر خلاف [۳۰]، ارتباطات کمتری میان GPUها وجود دارد.

۳) روش دسته‌بندی سریع‌تر چهره [۳۸] از مخزن دسته پویا[1] برای ذخیره‌سازی و به‌روزرسانی ویژگی‌های دسته‌ها استفاده می‌کند. این مخزن می‌تواند به‌عنوان جایگزینی برای لایه تمام متصل در نظر گرفته شود و به‌دلیل اندازه کوچک‌تر خود، موجب صرفه‌جویی در مصرف حافظه و زمان می‌شود. این روش در ادامه از بارگیرهای مبتنی بر دسته و نیز مبتنی بر نمونه دوگانه بهره می‌گیرد که موجب می‌شود عملیات به‌روزرسانی پارامترهای مخزن دسته پویا به شکل کاراتری صورت پذیرد.

۴) روش تجزیه سافت‌مکس[2][۲۹] اجزای فرمول سافت‌مکس را به دو بخش مجزا تجزیه کرده‌است:

$$P_i = \frac{e^{W_i^T x + b_i}}{\sum_{j=1}^{N} e^{W_j^T x + b_j}} \quad (۱)$$

در فرمول (۱) صورت کسر بیان‌گر فاصله درون‌دسته‌ای و مخرج کسر بیان‌گر فاصله برون‌دسته‌ای است. مشکلی که فرمول سافت‌مکس دارد، این است که تلاش در کمینه‌سازی صورت، به‌ناچار موجب افت مقدار مخرج نیز خواهد شد و بالعکس. اما تجزیه آنها، به دو بخش به‌طورکامل مجزا مشکل یادشده را ندارد. از سوی دیگر، محاسبه‌صورت کسر، بسیار آسان است؛ اما محاسبه مخرج کسر بسیار زمان‌بر است. همچنین مشاهدات نشان داده‌اند که دسته‌های منفی، تأثیری به مراتب کمتر از دسته مثبت در نتیجه نهایی دارد؛ بنابراین به جای آن که از همه دسته‌های منفی در قسمت فاصله‌های برون‌دسته‌ای استفاده کنیم، نمونه‌برداری از آنها کافی است که به این ترتیب، به‌شدت در مصرف حافظه و پردازش صرفه‌جویی خواهد شد. این روش نیز اندکی دچار افت دقت است.

۴) روش ArcFace با نمونه‌برداری: این روش[۳۱]، بسیار مشابه با روش به‌کاررفته در ArcFace[۲۸] است، با این تفاوت که با الهام از ایده مطرح‌شده در روش تجزیه Softmax[۲۹]، به‌جای آن که از همه وزن‌های موجود در هر GPU استفاده شود، تنها از وزن‌های متناظر با دسته مثبت و نیز نمونه‌برداری از دسته‌های منفی استفاده می‌شود. این روش نیز کمترین افت دقت را دارد.

چهار روش بالا، همگی در صدد یافتن راهی برای پیاده‌سازی سافت‌مکس یا روش‌های هم‌خانواده آن هستند و هدف نهایی آنها کاستن از مصرف حافظه و افزایش سرعت است. برای رسیدن به هدف، در گام آموزش مدل‌ها، تغییراتی پدید آورده‌اند.

ب) تغییر معماری با افزودن پیش‌پردازش یا پس‌پردازش

در مقابل، برخی پژوهش‌ها، به جای آن که در خود مدل‌ها دست ببرند و تغییراتی را در گام آموزش پدید آورند، تنها با ارائه برخی پیش‌پردازش یا پس‌پردازش، سعی در برطرف ساختن مشکلات یادشده کرده‌اند.

[1] Dynamic Class Pool

[2] Softmax Dissection

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

این دسته پژوهش‌ها اغلب از سازوکار توزیعی برای حل مسئله استفاده کرده‌اند و معتقدند که تمرکز تنها بر روی یک شبکه (یا درنهایت دو یا سه شبکه) نمی‌تواند مشکل چالش تعداد زیاد دسته‌ها را در مسئله شناسایی چهره، حل کند. درکل، زمانی که تعداد دسته‌ها از حد خاصی فراتر رود، به‌دلیل پیچیده‌شدن شبکه و افزایش بسیار زیاد پارامترهای شبکه، تعداد زیاد عملیات‌های محاسباتی ضرب (برای مثال، ضرب ورودی در وزن، یا ضرب‌های موجود در توابع فعالیت، یا عملیات‌های مربوط به محاسبه توابع زیان، یا عملیات‌های مربوط به پس‌انتشار و به‌روزرسانی وزن‌ها)، اندکی خطای محاسباتی بر اثر نابودی بخش‌هایی از اعشار تولید می‌شود؛ بنابراین حتی در بهترین حالت‌ها، و زمانی که بهترین توابع زیان و معماری را انتخاب کرده باشیم، از پس مشکل چالش تعداد زیاد دسته‌ها بر نخواهیم آمد؛ بنابراین، بسیار مناسب است که وظیفه شناسایی همه دسته‌ها را به یک شبکه واحد تحمیل نکنیم و این وظیفه را بین تعدادی شبکه موازی با هم تقسیم کنیم تا بتوانیم خطای بازشناسی را کاهش دهیم.

در این راستا، چند رویکرد کلی اتخاذ شده‌است:

۱. مدل سافت‌مکس مستقل[1][۳۲]، به جای آن که یک شبکه با N دسته داشته باشیم، m شبکه موازی، با اندازه حدوداً $\frac{N}{m}$ دسته مستقل، خواهیم داشت. تعیین دسته پیش‌بینی شده برای داده آزمون، با رویکرد بیشینه-بیشینه[2] انجام می‌شود. به این صورت که داده آزمون، به همه زیرشبکه‌ها داده می‌شود. دسته منتخب، بیشینه مقادیر سافت‌مکس از دسته‌های منتخب هر زیرشبکه است. این روش، از نظر سرعت و زمان اجرا بسیار مناسب است. اما مشکلی که دارد، این است که دسته‌ها را به شکل کاملاً تصادفی میان شبکه‌ها توزیع می‌کند. توزیع تصادفی میان شبکه‌ها موجب می‌شود که خطای بازشناسی، در مواردی که یک ورودی متعلق به یک دسته، شباهت زیادی به دسته یا دسته‌های دیگری هم داشته باشد و به‌صورت تصادفی اکثر این دسته‌ها در یک شبکه واقع شده باشند، و تعداد اندک دیگری نیز در زیرشبکه‌های دیگر پراکنده شده باشند، افزایش یابد ((شکل- ۲). با توجه به این که روش پیشنهادی مقاله، مبتنی بر این روش است، توضیح بیشتری از مشکل در بخش روش پیشنهادی خواهد آمد.

۲. مدل دسته‌بند شناختی چندگانه[3][۳۹]: در این روش، کل معماری به کار رفته در [۳۲] را به عنوان یکی از اجزای خود، تحت عنوان یک واحد شناختی، تعریف می‌کند. سپس چند واحد شناختی را به‌صورت موازی آموزش می‌دهد. داده آزمون، به همه واحدهای شناختی داده می‌شود. در نهایت، با مکانیزم رأی‌گیری، دسته‌ای که بیشترین رأی را در میان چند واحد شناختی دریافت کرده است، به عنوان دسته پیش‌بینی تعیین می‌شود. این روش، به منظور غلبه بر مشکل یاد شده در روش [۳۲] و برای خنثی کردن اثر خطاهای ناشی از توزیع نامناسب تصادفی، ارائه شده‌است. این روش، از نظر دقت، عملکرد بهتری دارد، ولی تعداد پارامترهای بیشتری دارد. در نتیجه زمان آموزش و پیش‌بینی و نیز میزان حافظه مصرفی بیشتری می‌طلبد.

۳. کدهای خروجی تصحیح خطا[4][۴۰]: این روش، شبکه N دسته‌ای را به تعدادی شبکه دو دسته‌ای موازی با هم تقسیم می‌کند. هر یک از شبکه‌های دودویی، به‌سان یک خوشه‌بندی عمل می‌کنند. ترکیب نتایج شبکه‌های دو دسته‌ای، برچسب دسته اصلی را نمایش خواهد داد. این روش نیاز به برقراری یک تناظر میان برچسب‌های مربوط به شبکه‌های دودویی و نیز برچسب‌های شبکه اصلی دارد. تصمیم‌گیری در مورد تعداد شبکه‌های دو دسته‌ای و نیز نحوه برقراری تناظر، بسیار اهمیت دارد. این کار باید به نحوی صورت پذیرد که استقلال سطری و ستونی در بیشترین حد خود صورت پذیرد تا با کمترین خطا مواجه باشیم. این روش، برای بازشناسی چهره مورد استفاده قرار نگرفته است. اما با توجه به این که برای غلبه بر دسته‌های متعدد ارائه شده‌است، می‌تواند ایده‌های خوبی برای پژوهش‌های جدید و بیشتر در این حوزه، فراهم آورد.

۴. نگاشت برچسب[5][۴۱]: توسعه‌ای از روش کدهای خروجی تصحیح خطا است. در این روش، به جای آن که شبکه N دسته‌ای، به زیرشبکه‌های دو دسته‌ای تقسیم شود، به زیرشبکه‌های با اندازه‌های بزرگ‌تر تقسیم می‌شود. اگر اندازه زیرشبکه‌ها یکسان باشد، روش را نگاشت برچسب ساده می‌نامیم، در غیر این صورت، آن را نگاشت برچسب ترکیبی می‌نامیم. این روش نیز تا کنون برای بازشناسی چهره مورد استفاده قرار نگرفته است. اما به نظر می‌رسد ایده خوبی برای به‌کارگیری در حوزه بازشناسی چهره باشد.

[1] Independent Softmax Model
[2] Max-Max
[3] Multi Cognition Softmax Model
[4] Error Correcting Output Codes (ECOC)
[5] Label Mapping

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]

نقطه‌ضعف مدل سافت‌مکس مستقل: با توجه به این که رویکرد معرفی شده در این مقاله، بر پایه مدل مستقل سافت‌مکس است، ما در این‌جا به نقد این روش می‌پردازیم.

همان‌طور که به شکل خلاصه اشاره شد، ضعف رویکرد *ISM*، ناشی از توزیع کاملاً تصادفی دسته‌ها در زیرشبکه‌ها (یا خوشه‌ها) است. برای توضیح این نقطه ضعف، از دو سناریو به عنوان نمونه استفاده خواهیم کرد.

در سناریوی اول، فرض کنید:

الف) یک مسئله با چهار دسته x، y، z و w داریم.

ب) این مسئله، به شکل تصادفی، به دو زیرشبکه A و B تقسیم شده‌است.

ج) زیرشبکه A شامل دسته‌های x و w است و زیرشبکه Bشامل دسته‌های y و z.

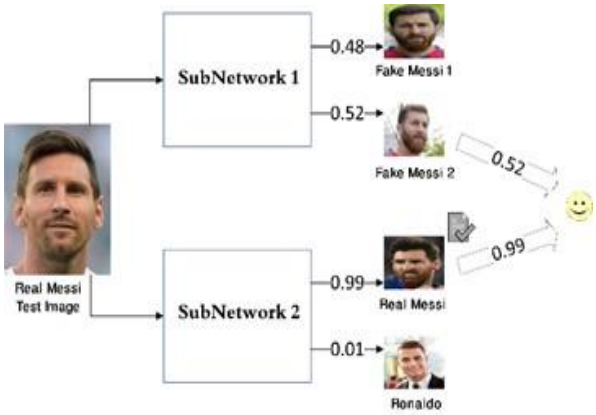
د) دسته‌های x, y, z بسیار مشابه با یکدیگر هستند و همگی فاصله بسیار زیادی با دسته w دارند.

هـ) برچسب واقعی تصویر آزمون، x است.

تصویر آزمون به هر دو زیرشبکه داده می‌شوند. با توجه به مشابهت بین دسته x و تصویر آزمون، امتیازهای خروجی به ترتیب در حدود ۰.۹۹ و ۰.۰۱ خواهند بود. به همین دلیل، با استفاده از رویکرد بیشینه-بیشینه در انتخاب زیرشبکه، دسته x از زیرشبکه A انتخاب خواهد کرد که یک حالت خوش‌شانسی است.

در سناریوی دوم، و فرض ج را با نام جدید ججِ به‌صورت زیر تعریف می‌کنیم و سایر فرض‌ها را حفظ می‌کنیم:

ج) دسته‌های x و y در زیرشبکه A قرار دارند و دسته‌های w و z در زیرشبکه B.



(شکل- ۱) خوش‌شانسی در مدل سافت‌مکس مستقل: زمانی که دسته واقعی هیچ رقیب جدی در زیرشبکه خود ندارد.

Figure 2. Fortunate in ISM: when the actual class does not face significant competition within its submodel.

در این حالت، امتیازهای خروجی زیرشبکه A به‌ترتیب برابر با ۰.۵۲ و ۰.۴۸ برای دسته‌های x و y خواهند بود. از طرف دیگر، زیرشبکه B برای دسته‌های w و z امتیازهای به‌ترتیب ۰.۸ و ۰.۲ را تولید می‌کند. به همین دلیل، با استفاده از رویکرد بیشینه-بیشینه در انتخاب زیرشبکه، دسته w از زیرشبکه B به‌عنوان دسته نهایی انتخاب می‌شود که یک حالت بدشانسی است؛ بنابراین، برای آن که رویکرد *ISM* با حالت بدشانسی مواجه شود، دو شرط ضروری وجود دارد: الف) دسته واقعی با دسته‌های مشابه در یک خوشه قرار داشته باشد. ب) دسته‌های مشابه در خوشه‌های دیگر منتشر شده باشند.

۳- روش پیشنهادی

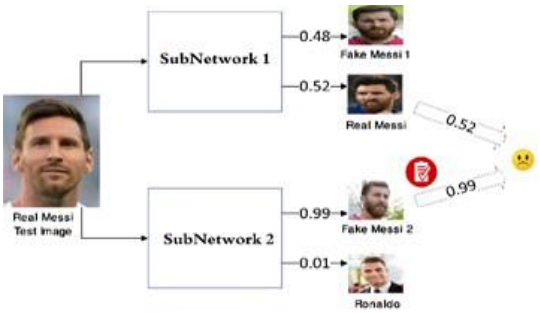
روش پیشنهادی این مقاله، با افزودن یک پیش‌پردازش و یک پس‌پردازش بیشتر نسبت به مدل سافت‌مکس مستقل[۳۲] یا به اختصار ISM، موجب بهبود دقت می‌شود.

همان‌گونه که در [۴۲] به تفصیل بیان شده‌است، توزیع تصادفی دسته‌ها میان زیرشبکه‌ها منجر به بروز خطاهای بدشانسی می‌شود. پژوهش حاضر، با إعمال یک خوشه‌بندی، به‌عنوان یک پیش‌پردازش، این نوع خطاها را کاهش می‌دهد. اما در عین حال چالش جدیدی که ممکن است مطرح شود، این است که توزیع دسته‌های مشابه، در خوشه‌های مشترک، موجب می‌شود که احتمال بروز خطا به‌دلیل مشابهت زیاد میان دسته‌های هر خوشه، نسبت به زمانی که دامنه تغییرات دسته‌ها در هر خوشه زیاد باشد، افزایش یابد؛ اما با در نظر داشتن چند نکته زیر، درخواهیم یافت که نه تنها روش پیشنهادی بر اثر خوشه‌بندی دچار افت دقت نمی‌شود، بلکه با افزایش دقت نیز مواجه خواهیم بود.

۱) درست است که تفکیک افراد متمایز، آسان‌تر از تفکیک افراد مشابه است، اما حتی در توزیع تصادفی دسته‌ها، امکان حضور دو یا چند دسته مشابه در یک خوشه وجود دارد و بنابراین امکان مشتبه شدن بین این دسته‌های مشابه، منتفی نیست. در مقایسه این دو وضعیت، به نظر می‌رسد که خطای شناسایی در حالت خوشه‌بندی کم‌تر باشد: وضعیت الف) توزیع دسته‌ها با خوشه‌بندی: دامنه تغییرات داده‌ها اندک است. و دسته‌های داخل هر زیرشبکه، شباهت بالایی با یکدیگر دارند. وضعیت ب) توزیع دسته‌ها به‌صورت تصادفی: دامنه تغییرات داده‌ها زیاد است. و از سوی دیگر، دسته‌های مشابه و نزدیک به هم در هر زیرشبکه وجود دارند.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]



(شکل- ۲) بدشانسی در مدل سافت‌مکس مستقل: زمانی که دسته واقعی با رقبای قوی در زیرشبکه خودش مواجه می‌شود و دسته‌های مشابه دیگری بدون رقیب جدی، در زیرشبکه‌های دیگری هستند.

Figure 3. Unfortunate in ISM: When the actual class faces competition from similar classes within its submodel, as well as similar classes in other submodels that lack serious competitors.

برای تقریب به ذهن، می‌توان تصور کرد که اگر با یک ذره‌بین بر روی مجموعه‌ای که دامنه تغییرات اندکی دارند، بزرگ‌نمایی کنیم، دسته‌های هر زیرشبکه، به شکل نسبتاً واضح و متمایزی از هم قابل تفکیک هستند و فاصله دسته‌ها در حالت پس از بزرگ‌نمایی بیش از فاصله آنها در حالت پیش از بزرگ‌نمایی خواهد شد. این در حالی است که برای وضعیت دوم (دامنه تغییرات بالا)، امکان بزرگ‌نمایی چندانی وجود ندارد. و بنابراین در وضعیت نخست، شبکه با تمرکز و دقت بالاتری قادر به شناسایی دسته‌ها از یکدیگر خواهد بود؛ اما در وضعیت دوم، دسته‌های دشوار ومشابه، کماکان چالش بیشتری ایجاد خواهند کرد. شکل (۱)

۲) حتی با فرض این که در توزیع تصادفی، هیچ دو دسته مشابهی در یک زیرشبکه واحد قرار نمی‌گیرد، باز هم چالش در توزیع تصادفی بیشتر خواهد بود. در توزیع تصادفی، پس از آن که هر زیرشبکه، بهترین دسته خود را معرفی کرد، تصمیم‌گیری نهایی منوط به مقایسه امتیازات دسته‌های برگزیده از هر زیرشبکه است.

با توجه به این که دسته‌های برگزیده از هر زیرشبکه، از رقابت‌های متعدد و متفاوتی پیروز شده‌اند، لذا مقایسه امتیازات میان این دسته‌ها، نمی‌تواند دقیق باشد. چرا که ممکن است یک دسته برگزیده با امتیاز متوسط، به‌دلیل این که رقبای بسیار قدرتمندی داشته است، امتیازش شکسته شده‌است؛ و از سوی دیگر، یک دسته با امتیاز بالا، تنها به‌دلیل این که رقیب جدی‌ای نداشته است، امتیاز بالاتری دریافت کرده است. این در حالی است که در روش مبتنی بر خوشه‌بندی، به دلیل آن که تصویر آزمون ورودی، به یک خوشه، بیش از سایرین شباهت دارد، ما با قطعیت به‌نسبه بالایی مطمئنیم که دسته مورد نظر در همین زیرشبکه واقع است. و بنابراین بحث رقابت‌های متعدد در هر زیرشبکه، برای این روش چالشی محسوب نمی‌شود. از سوی دیگر، دقت یک زیرشبکه با N/m دسته، بالاتر از دقت یک شبکه با N دسته خواهد بود.

نتیجه این که روش مبتنی بر خوشه‌بندی افزایش دقت بیشتری نسبت به روش رقیب خود خواهد داشت.

۳-۱- نگاهی کلی به چارچوب

روش پیشنهادی بر روی ویژگی‌های استخراج‌شده از مدل‌های پیش‌آموزش‌دیده کار می‌کند. بر این اساس در این مقاله، از شبکه‌های عصبی کانولوشنی با معماری InceptionResnet [۴۳, ۴۴] برای استخراج ویژگی‌ها از تمام تصاویر در هر مجموعه‌دادگان استفاده شده‌است؛ بنابراین مجموعه‌دادگان ما از تصویر به بردارهای ویژگی $F_D = \{f_i\}_{i=1}^{N}$ تبدیل می‌شوند که منظور از f_i بردارهای ویژگی مربوط به تصویر i ام از همه تصاویر مجموعه‌دادگان D است و N نشان‌دهنده تعداد تصاویر در مجموعه‌دادگان است. نمایی کلی از روش پیشنهادی در شکل (۵) قابل مشاهده است.

۳-۲- واحدهای چارچوب

چارچوب ارائه شده، شامل سه واحد می‌شود:

۳-۲-۱- زیرشبکه‌ها

هر زیرشبکه، یک شبکه عصبی نظارتی است. اگر فرض کنیم که تعداد کل دسته‌های مسئله N باشد، و تعداد زیرشبکه‌ها را نیز m در نظر بگیریم، هر زیرشبکه حاوی حدوداً N/m دسته خواهد بود. این دسته‌ها با کمک الگوریتم‌های خوشه‌بندی در میان زیرشبکه‌ها توزیع می‌شوند. مقاله حاضر، از الگوریتم $KMeans$ استفاده می‌کند.

تعداد خوشه‌ها جزء پارامترهایی است که با آزمایش می‌توان آن را تخمین زد و مقدار بهینه آن بستگی به عوامل متعددی همچون کیفیت داده‌های مجموعه‌دادگان، تعداد کل دسته‌ها و دقت استخراج‌گر ویژگی دارد، اما به‌صورت کلی، آزمایش‌های ما نشان دادند که تعداد دوهزار تا سه‌هزار دسته برای هر خوشه، تعداد مناسبی محسوب می‌شود و خطر بیش‌برازش برای آن وجود نخواهد داشت. اندازه‌های کمتر برای هر خوشه، تعداد خوشه‌ها را افزایش خواهد داد که خطر افزایش خطای خوشه‌بندی را به دنبال خواهد داشت. اندازه‌های بزرگ‌تر، موجب افزایش تعداد دسته‌ها و در نتیجه افزایش احتمال بیش‌برازش می‌شود.

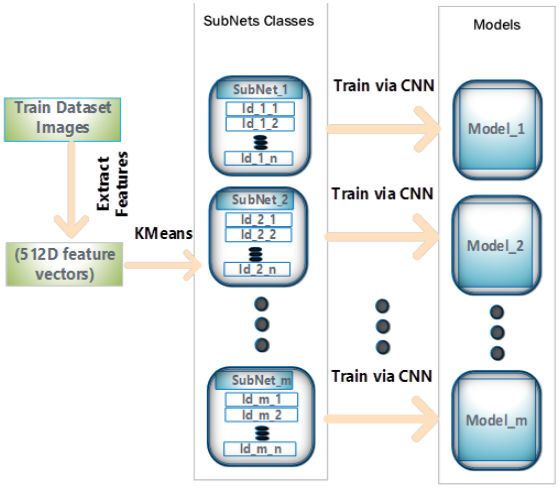
پس از آن که خوشه‌بندی انجام شد، در گام آزمون، داده آزمون ورودی، با هر یک از خوشه‌ها (زیرشبکه‌ها) مقایسه می‌شود و احتمال تعلقش به یکی از خوشه‌ها بیش از سایرین خواهد بود. این همان فایده و امتیازی است که روش ISM فاقد آن است؛ اما در مورد این که این مقایسه و تعلق داده آزمون به خوشه‌ها به چه روشی صورت پذیرد، باید خاطرنشان کرد که چند روش زیر قابل استفاده هستند:

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

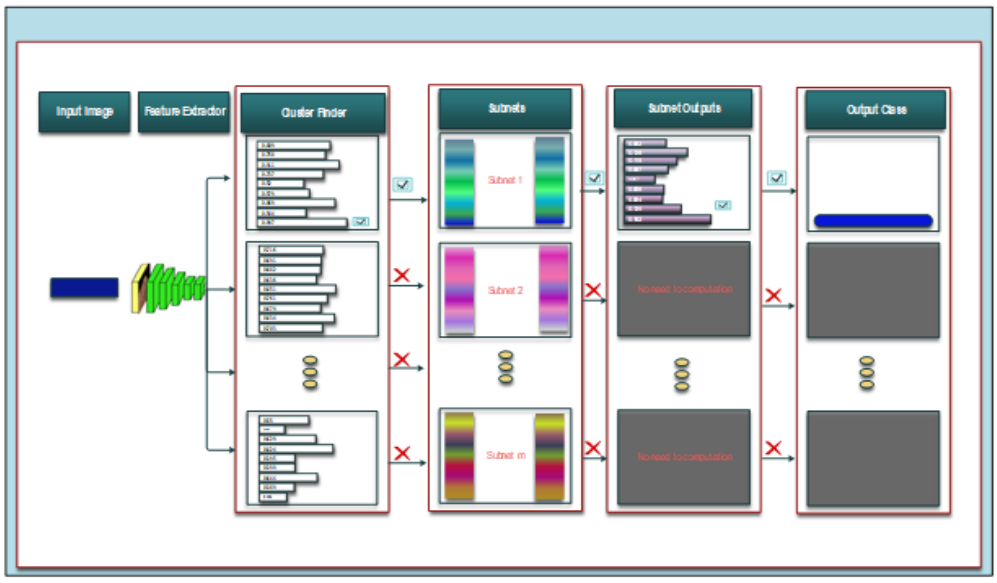
۱. استفاده از میانگین خوشه‌ها و مقایسه بردار ویژگی داده آزمون با میانگین خوشه‌ها.
۲. استفاده از معیار بیشترین مقدار خروجی مورد استفاده در روش سافت‌مکس مستقل. به‌عبارت دیگر، به‌ازای هر داده آزمون، نماینده هر زیرشبکه، دسته‌ای است که بیشترین مقدار خروجی را دریافت کرده باشد.
۳. استفاده از نزدیک‌ترین همسایه هر زیرشبکه و مقایسه مقادیر شباهت داده آزمون با هر زیرشبکه.

آزمایش‌های ما نشان دادند که روش‌های نخست و دوم چندان قابل اعتماد نیستند؛ اما از نظر تحلیلی، چرا میانگین خوشه، نماینده خوبی برای خوشه نباشد؟ میانگین خوشه، در صورتی می‌تواند معیار خوبی برای نمایندگی خوشه باشد که تمامی اعضای هر خوشه، به‌خوبی متمایز از اعضای سایر خوشه‌ها باشند؛ اما به‌دلیل ابعاد به‌نسبه بالای ویژگی‌های تصاویری، چنین مرز و حاشیه‌ای میان خوشه‌ها، درعمل به‌صورت واقعی وجود ندارد. ضمن این که ویژگی‌های استخراج‌شده از تصاویر ذاتاً با هدف شناسایی آموزش دیده شده‌اند، و نه خوشه‌بندی و لذا میانگین خوشه‌ها نمی‌توانند تضمین‌کننده دقت کافی برای یک نمایندگی از همه اعضای یک خوشه باشد.



(شکل- ۳) ایده اصلی مقاله در گام پیش‌پردازش: همه دسته‌ها بین زیرشبکه‌های متعدد با استفاده از خوشه‌بندی توزیع می‌شود. هر زیرشبکه حاوی دسته‌های مشابه است.

(Figure-4) The main concept in the pre-processing phase involves distributing all classes into multiple subnetworks using clustering. Each subnetwork contains similar classes.



(شکل- ۵) . ایده اصلی مقاله در گام آزمون، ابتدا ویژگی‌های داده آزمون ورودی، استخراج، سپس با استفاده از ماژول خوشه‌یاب، بهترین خوشه متناسب با داده آزمون ورودی انتخاب می‌شود و سپس با استفاده از مدل زیرشبکه انتخاب‌شده، پیش‌بینی صورت می‌گیرد و واحد تصمیم‌گیر نهایی، دسته با بالاترین مقدار خروجی را به‌عنوان دسته برگزیده مشخص می‌کند

(Figure -5) . The main idea of the paper in the testing phase involves initially extracting features from the test input image. Subsequently, using the clustering module, the best cluster corresponding to the test input data is selected. Then, employing the selected subnetwork model, predictions are made, and the final decision unit identifies the selected class as the one with the highest output value.

پرسش بعدی این است که چرا رویکرد بیشینه-بیشینه معیار خوبی برای نمایندگی از خوشه‌ها نیست؟ گفتنی است که طبق این معیار، نماینده هر خوشه، به‌ازای هر داده ورودی آزمون متفاوت است. نماینده هر خوشه، دسته‌ای است که بیشترین مقدار خروجی را از آن خوشه، به‌ازای داده ورودی آزمون داشته باشد. روش مورد استفاده مدل سافت‌مکس مستقل برای تعیین نماینده زیرشبکه‌ها، همین است. با این تفاوت که در مدل سافت‌مکس مستقل، زیرشبکه‌ها با استفاده از خوشه‌بندی به‌وجود نیامده‌اند و تنها به‌صورت تصادفی، در زیرشبکه‌ها توزیع شده‌اند، اما

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]

حتی درصورتی‌که زیرشبکه‌ها با خوشه‌بندی به وجود آمده باشند، باز هم این معیار، کافی به نظر نمی‌رسد. با توجه به این که نماینده هر خوشه، با رقبای مختلفی در حال رقابت است، لذا این احتمال وجود دارد که نماینده هیچ خوشه‌ای به‌صورت واضحی مقدار خروجی بالاتری دریافت نکند. در خصوص زیرشبکه واقعی (زیرشبکه‌ای که دسته واقعی داده آزمون ورودی در آن وجود دارد)، از آن جا که دسته‌های مشابه را در خود جای داده است، لذا امتیاز دسته واقعی به دلیل رقابل با دسته‌های دشوار و مشابه، شکسته می‌شود و احتمال این که مقدار خروجی اندکی دریافت شود، وجود دارد. از سوی دیگر، در زیرشبکه‌هایی که حاوی دسته واقعی نیستند، از آن جا که همگی وضعیت به‌نسبه مشابهی نسبت به داده آزمون ورودی دارند، امتیاز دسته برگزیده از هر زیرشبکه، از نظر تئوری پایین است و لذا ممکن است که تفکیک واضحی بین امتیازهای خوشه‌ها (اعم از خوشه واقعی و خوشه‌های غیرواقعی) برقرار نشود. درنهایت، درصورتی‌که ویژگی‌های استخراج‌شده، کیفیت و دقت بالایی داشته باشند، معیار نزدیک‌ترین همسایه در هر خوشه، می‌تواند بسیار موفق عمل کند. به عبارت دیگر، نماینده هر خوشه، دسته‌ای است که بیشترین شباهت را با داده آزمون ورودی داشته باشد. در این معیار هم نماینده خوشه، بسته به داده آزمون ورودی متفاوت خواهد بود.

با فرض آن که دقت و تفکیک‌پذیری ویژگی‌های استخراج‌شده با استفاده از روش KNN برابر با x باشد، به‌راحتی قابل اثبات است که دقت خوشه‌بندی با استفاده از این روش، حداقل برابر با x خواهد بود. از آن جا که در اغلب موارد خطاهای KNN ناشی از مشابهت زیاد میان دسته‌ها است، و از سوی دیگر دسته‌های مشابه، با احتمال بسیار بالایی در خوشه‌های مشترک واقع شده‌اند، لذا در صورتی که نزدیک‌ترین همسایه داده آزمون ورودی، هم‌دسته با آن نباشد، اما با قطعیت بسیار بالایی، دسته واقعی داده آزمون ورودی، با احتمال بالایی در همان خوشه قرار دارد.

پس از آن که با دقت بسیار بالایی توانستیم، خوشه حاوی دسته اصلی را شناسایی کنیم، در مرحله بعد، با یک مسئله شناسایی چهره به اندازه N/m روبرو هستیم که قطعاً دقت بالاتری نسبت به مسئله شناسایی چهره برای N دسته خواهد داشت.

تحلیل پیچیدگی زمانی: واحد زیرشبکه‌ها در گام پیش‌آموزش، شامل یک خوشه‌بندی است که با فرض آن که از رویکرد نخست استفاده شود (استفاده از میانگین ویژگی‌های نمونه‌های دسته)، و بخواهیم مسئله N دسته‌ای را به m خوشه (زیرشبکه) توزیع کنیم و خوشه‌بندی هم به‌صورت افرازی انجام شود، آن‌گاه پیچیدگی زمانی این مرحله، همان پیچیدگی زمانی الگوریتم $KMeans$ است که عبارت است از $O(imN)$ که منظور از i، تعداد گام‌ها تا رسیدن به نقطه توقف یا هم‌گرایی است.

در گام آموزش، هر زیرشبکه، با فرض این که هر دسته شامل t داده آموزشی است، شامل حدوداً N/m دسته و $N \times t/m$ نمونه آموزشی است. هر زیرشبکه، در قالب یک شبکه عصبی نظارتی، حداقل دارای یک لایه ورودی و یک لایه خروجی (دسته‌بند) آموزش می‌بینند. هر زیرشبکه، تنها با داده‌های آموزشی مربوط به خودش آموزش می‌بینند و لایه ورودی پذیرنده بردار ویژگی هر داده آموزشی است. اگر هر بردار دارای d عنصر باشد، آن‌گاه ماتریس وزن حاوی $d \times N/m$ عنصر خواهد بود؛ بنابراین هر زیرشبکه به اندازه $t \times d \times N^2/m^2$ عملیات در هر اپوک خواهد داشت (صرفنظر از عملیات‌های مربوط به به‌روزرسانی وزن‌ها).

با توجه به استقلال کامل زیرشبکه‌ها از یکدیگر در گام آموزش، امکان موازی‌سازی در چندین ماشین مجزا وجود دارد و در آن صورت، زمان اجرای کل چارچوب، در گام آموزش، حدوداً m برابر می‌شود.

زمانی که شبکه با N دسته، از روش توزیعی برای حل مسئله استفاده نکند، ماتریس وزن حاوی $d \times N$ عضو خواهد داشت. تعداد نمونه‌های آموزشی $N \times t$ است و در نتیجه در هر اپوک نیاز به $t \times d \times N^2$ عملیات دارد. که m برابر کندتر از روش توزیعی سریالی، و m^2 برابر کندتر از روش توزیعی موازی خواهد بود. برای مثال، برای یک مسئله با اندازه پنجاه‌هزار دسته، در صورتی که مسئله را به ۲۵ زیرشبکه توزیع کنیم و از اجرای موازی برای آموزش استفاده کنیم، کل فرآیند آموزش در اجرای موازی توزیعی، در حدود ۶۲۵ برابر سریع‌تر از فرآیند آموزش در اجرای بدون توزیع خواهد بود.

تحلیل پیچیدگی حافظه‌ای: داده‌های آموزشی ورودی هر زیرشبکه، در هر گام از هر اپوک، به بسته‌های b عضوی تقسیم می‌شوند؛ بنابراین در هر اپوک، کل مجموعه‌دادگان آموزشی به $N/(m \times b)$ بسته قابل تقسیم است. (N تعداد کل دسته‌ها، t تعداد نمونه‌های مربوط به هر دسته، m تعداد زیرشبکه‌ها و b تعداد اعضای هر بسته است). ماتریس وزن‌ها حاوی $d \times N/m$ عنصر است که با فرض آن که هر عنصر چهار بایت حافظه را اشغال می‌کند، هر گام نیاز به $4 \times b \times d \times N/m$ حافظه با دسترسی تصادفی دارد. به طور مثال، برای یک مسئله پنجاه‌هزار دسته‌ای با بسته‌های ۲۵۶ عضوی و

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

بردارهای ویژگی ۵۱۲ عنصری و ۲۵ زیرشبکه، هر گام، نیاز به حدوداً یک گیگابایت حافظه دسترسی تصادفی دارد.

۲-۲-۳- واحد خوشه‌یاب

این واحد، در گام آزمون استفاده می‌شود. به‌ازای هر داده آزمون جدید، ابتدا بردارهای ویژگی داده آزمون استخراج می‌شود و سپس این واحد مورد استفاده قرار می‌گیرد. برای یافتن خوشه (زیرشبکه)، از روش نزدیک‌ترین همسایه استفاده می‌شود. به عبارت دیگر، خوشه مورد نظر برای داده آزمون، عبارت از خوشه‌ای است که نزدیک‌ترین دسته به نمونه آزمون، در آن باشد.

$$CC = \arg\max_{i \in N}(Sim(F_{test}, \overline{F_{c_i}})), \quad (۱)$$

که منظور از Sim معیار شباهت است. F تابع استخراج‌گر ویژگی است. $\bar{F}_{c_i}$ میانگین بردارهای ویژگی مربوط به دسته iام است. منظور از CC[26] ، نزدیک‌ترین دسته به داده آزمون ورودی است.

پس از آن که نزدیک‌ترین دسته به داده ورودی آزمون، مشخص شد، بررسی می‌کنیم که دسته مورد نظر در کدام زیرشبکه واقع شده‌است:

$$FindCluster = \{clst \in Clusters \mid CC \in clst\} \quad (۲)$$

یافتن شبیه‌ترین دسته به داده آزمون، به اندازه $O(N \times d)$ زمان می‌برد. همچنین یافتن خوشه مورد نظر، به اندازه $O(N)$ پیچیدگی زمانی دارد.

۳-۳-۳- واحد تصمیم‌گیر نهایی

پس از آن که خوشه (زیرشبکه) مناسب داده آزمون ورودی، توسط واحد خوشه‌یاب مشخص شد، واحد تصمیم‌گیر نهایی، بررسی می‌کند که کدام دسته در خوشه انتخاب شده، بهترین امتیاز خروجی را به ازای داده آزمون ورودی دریافت کرده است و آن را به‌عنوان دسته برگزیده مشخص کند. از آن‌جا که دسته برگزیده خوشه، برچسب جدیدی دریافت کرده است، یک عملیات نگاشت از برچسب جدید به برچسب اصلی انجام می‌دهد و دسته نهایی را مشخص می‌کند.

الگوریتم ۱، توصیف یک‌پارچه‌ای از گام پس‌پردازش (گام آزمون) ارائه می‌کند.

۴- آزمایش‌ها

مجموعه‌دادگان: در این مطالعه از مجموعه‌دادگان زیر استفاده گردیده است:

۱. مجموعه‌دادگان MS-Celeb-1M [۳۴]: شامل ۱۰۰ هزار فرد است، و هر فرد دارای ۱۰۰ تصویر چهره است.
۲. مجموعه‌دادگان VGG-Face2 [۳۳]: حاوی بیش از ۳.۳ میلیون تصویر مربوط به بیش از نُه هزار فرد است.
۳. مجموعه‌دادگان Glint360K [۳۱]: حاوی ۳۶۰ هزار شخص و حدود ۱۷ میلیون تصویر. از جامع‌ترین و تمیزترین پایگاه‌دادگان در حوزه پردازش چهره محسوب می‌شود.

برای هر سه مجموعه‌دادگان، دسته‌ها به‌صورت تصادفی انتخاب شده‌اند. ۶۰ درصد از تصاویر هر دسته به منظور آموزش، ۲۰ درصد به منظور آزمون و ۲۰ درصد به منظور ارزیابی اختصاص داده شده‌است. به منظور اطمینان از منصفانه‌بودن و دقت نتایج، در آزمون از تعداد یکسانی نمونه از هر دسته استفاده شده‌است (پنج نمونه). برای دسته‌هایی با تعداد نمونه کمتر از پنج، داده‌های جدیدی با تکثیر داده و ایجاد تبدیل‌هایی نظیر دوران، وارونه[27]، تغییر روشنایی یا تغییر تقابل[28]، تولید شده‌اند.

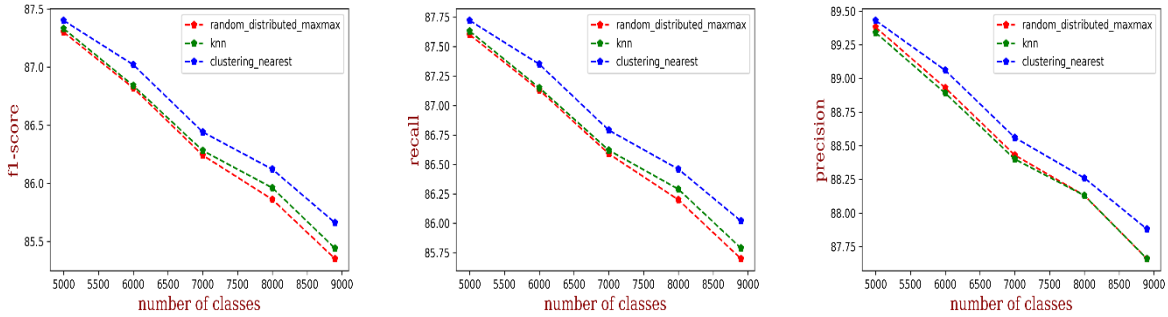
الگوریتم ۱. گام پس‌پردازش (تصمیم‌گیری)
Algorithm 1: Post-processing (Decision) phase

ورودی‌ها	face: تصویر چهره با ابعاد ۱۱۲ × ۱۱۲ RGB m: تعداد زیرشبکه‌ها N: تعداد دسته‌ها
خروجی	y_{pred}: برچسب دسته پیش‌بینی‌شده
#	استخراج ویژگی
۱	v: بردار ویژگی face استخراج شده توسط استخراج‌گر ویژگی F.
#	خوشه‌یابی: الف) یافتن نزدیک‌ترین دسته به عنوان نماینده
۲	برای همه دسته‌ها: از دسته $cls = 1$ تا $cls = N$:
۳	میزان شباهت بردار ویژگی داده آزمون ورودی (v) را با میانگین بردارهای ویژگی نمونه‌های هر یک از دسته‌ها ($\bar{F}(Sample_{i_{cls}})$)، با استفاده از معیار شباهت کسینوسی به دست آور (با نام Sim_{cls})
۴	از بین همه Sim_{cls}ها، دسته‌ای که بیشترین مقدار شباهت را دارد، با نام $ClosestCls$ در نظر بگیر.
#	خوشه‌یابی: ب) یافتن خوشه حاوی نزدیک‌ترین دسته
۵	از بین همه خوشه‌ها: از خوشه $clst = 1$ تا $clst = m$:
۶	خوشه منتخب ($SelectedClst$): خوشه حاوی نزدیک‌ترین دسته ($ClosestCls$)
#	تصمیم‌گیری نهایی
۷	برچسب دسته‌ای که بیشترین مقدار خروجی را برای داده ورودی آزمون برگردانده، مشخص کن.
۸	y_pred : نگاشت برچسب را انجام بده و برچسب نهایی دسته برگزیده را به دست آور.
۹	y_{pred} را برگردان.

[26] Closest Class

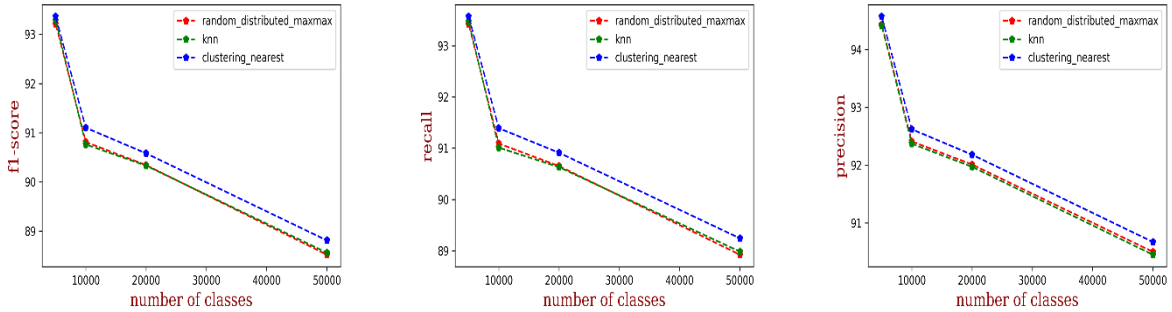
[27] Flip
[28] Contrast

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]



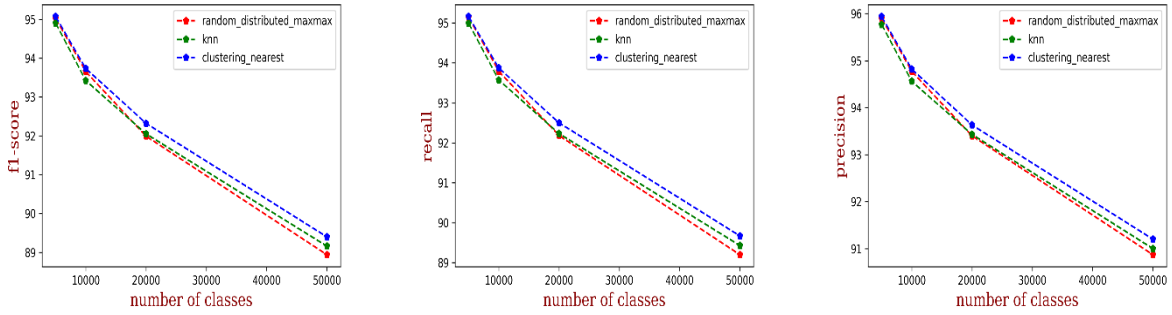
(شکل-۶) میزان افت دقت در سناریوهای گوناگون برای پایگاه دادگان VGGFace2 [۳۳] و InceptionResnetV1 [۴۴] برای استخراج ویژگی. از راست به چپ: صحت، دقت و f1.

(Figure- 6) Accuracy drops rates in various scenarios for the VGGFace2 dataset [33] and using InceptionResnetV1 [44] for feature extraction. From right to left: precision, accuracy, and F1 score



(شکل-۷) میزان افت دقت در سناریوهای گوناگون برای پایگاه دادگان MS-Celeb-1M [۳۴] و استفاده از InceptionResnetV1 [۴۴] برای استخراج ویژگی. از راست به چپ: صحت، دقت و f1.

(Figure -7) Accuracy drops rates in various scenarios for the MS-Celeb-1M dataset [34] and using InceptionResnetV1 [44] for feature extraction. From right to left: precision, accuracy, and F1 score



(شکل-۸) میزان افت دقت در سناریوهای گوناگون برای پایگاه دادگان Glint360K [۳۱] و استخراج‌گر ویژگی InceptionResnetV1 [۴۴] از راست به چپ: صحت، دقت و f1.

(Figure- 8) Accuracy drops rates in various scenarios for the Glint360K dataset [31] and InceptionResnetV1 [44] for feature extraction. From right to left: precision, accuracy, and F1 score

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

استخراج ویژگی‌ها: در روش پیشنهادی، به جای پردازش مستقیم تصاویر، الگوریتم بر روی ویژگی‌های به‌دست‌آمده از تصاویر عمل می‌کند. این ویژگی‌ها از شبکه‌های کانولوشنی Residual-Networks تحت عنوان InceptionResnetV1 [۴۳, ۴۴] استخراج می‌شوند. این معماری‌ها قابلیت پردازش تصاویر RGB را دارند و آن‌ها را به بردارهای ویژگی ۵۱۲ بعدی تبدیل می‌کنند.

معیارهای ارزیابی: معیارهای ارزیابی استفاده شده دراین مطالعه عبارتند از صحت[1]، بازخوانی[2] (نرخ مثبت صحیح[3]) و امتیاز F1[4].

$$\Pr ecision = \frac{TP}{TP + FP}, \quad (۳)$$

$$\mathrm{Re}call(TPR) = \frac{TP}{TP + FN}, \quad (۴)$$

که منظور از TP نرخ مثبت صحیح؛ FP نرخ مثبت غلط؛ و FN نرخ منفی غلط است.

$$F1 = 2 \times \frac{\Pr ecision \times \mathrm{Re}call}{\Pr ecision + \mathrm{Re}call}. \quad (۵)$$

انتخاب نماینده‌های زیرشبکه‌ها: در ابتدای امر، این گونه به نظر می‌رسد که وقتی خوشه‌بندی را بر روی تمامی نمونه‌های آموزشی إعمال می‌کنیم، همه نمونه‌های آموزشی مربوط به هر دسته، در خوشه مشترکی واقع می‌شوند. اما در عمل، چنین اتفاقی نمی‌افتد و گاهی نمونه‌های یک دسته، در خوشه‌های متعددی توزیع می‌شوند. در این‌جا چند راه حل به نظر می‌رسد:

۱. به جای آن که خوشه‌بندی را بر روی همه نمونه‌های آموزشی إعمال کنیم، بر روی میانگین نمونه‌های هر دسته اعمال کنیم. با این کار، علاوه‌بر آن که چالش یادشده برطرف خواهد شد، پیچیدگی مسئله نیز به‌دلیل کاهش تعداد اعضا در مسئله خوشه‌بندی (هر دسته، یک نمونه)، کاهش می‌یابد.

۲. دسته مورد نظر را به خوشه‌ای نسبت دهیم که تعداد بیشتری از نمونه‌های آن دسته را در خود جای داده باشد. یکی از چالش‌های این رویکرد، جایی است که چند خوشه، حاوی تعداد برابری از تعداد نمونه‌های آن دسته باشد.

۳. دسته مورد نظر را به همه خوشه‌هایی که حاوی نمونه‌هایی از آن دسته هستند، تخصیص دهیم. چالش این رویکرد این است که ممکن است همپوشانی میان خوشه‌ها زیاد شود و در بدترین حالت، خوشه‌ها متمایل به یکی شدن با یکدیگر پیدا کنند. در آن صورت، فایده شکسته‌شدن مسئله به چند زیرمسئله کوچک‌تر زیر سؤال خواهد رفت.

۴. برای آن که مشکل رویکرد سوم برطرف شود، دسته مورد نظر را تنها به خوشه‌هایی تخصیص دهیم که تعداد نمونه‌های موجود در آن خوشه‌ها، بیش از یک مقدار آستانه معین باشند. خوشه‌هایی که حاوی تعداد بسیار اندکی از نمونه‌های آموزشی آن دسته باشند (کمتر از مقدار آستانه) نادیده می‌گیریم و به عبارت دیگر آن نمونه‌های آموزشی را به‌عنوان داده‌های پرت در نظر می‌گیریم.

این کار مشکل هم‌پوشانی فراوان و درنتیجه میل به یکسان‌شدن خوشه‌ها مرتفع خواهد شد. مزیت مهم این رویکرد این است که چنان‌چه داده آزمون ورودی، به‌اشتباه مورد شناسایی قرار گیرد (به‌دلیل ضعف موجود در استخراج‌گر ویژگی)، با احتمال بیشتری دسته اشتباه مورد شناسایی با دسته واقعی آن داده آزمون، هم‌خوشه خواهد شد و درنتیجه به دسته واقعی، شانس بیشتری برای حضور در رقابت و برنده شدن می‌دهیم.

در مقاله حاضر، از رویکرد نخست استفاده شده‌است. نشان داده می‌شود که حتی استفاده از یک الگوریتم ساده خوشه‌بندی بدون استفاده از تکنیک‌های دیگر، می‌تواند منجر به بهبود دقت نسبت به مدل پایه گردد.

استفاده از رویکرد چهارم، می‌تواند در مطالعات بعدی مورد توجه قرار گیرد.

جزئیات پیاده‌سازی: برای خوشه‌بندی از الگوریتم KMeans استفاده شده‌است. با توجه به وابستگی نتایج این الگوریتم به انتخاب مراکز اولیه خوشه، از گونه بهبودیافته آن، KMeans++ استفاده شده‌است. برای هر زیرشبکه، در همه آزمایش‌های انجام‌شده، از معماری واحدی استفاده شده‌است. لایه ورودی، شامل یک بردار ۵۱۲ بعدی است؛ سپس یک نرمال‌سازی سطح ۲ انجام می‌پذیرد. لایه نهایی (لایه دسته‌بند)، حاوی N_i نورون است. که i اندیس زیرشبکه است. تابع زیان مورد استفاده، $ArcFace$ است. برای بهینه‌سازی از تابع $Adam$ با نرخ یادگیری ۰.۰۰۱ و پارامتر بتا۱ برابر ۰.۹ و پارامتر بتا۲ برابر ۰.۹۹ استفاده شده‌است. تعداد اپوک‌ها برای همه زیرشبکه‌ها، حداکثر ۵۰ در نظر گرفته شده‌است. آزمایش‌ها بر روی مسائلی با اندازه دسته‌های انجام شده‌اند. دسته‌ها به‌صورت تصادفی از مجموعه‌دادگان انتخاب شدند.

[1] Precision
[2] Recall
[3] True Positive Rate (TPR)
[4] F1-score

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]

مقایسه: سناریوهای زیر در آزمایش‌ها مورد ارزیابی واقع شده‌اند:

۱. شناسایی چهره بدون توزیع (با ArcFace)؛

۲. شناسایی چهره با توزیع تصادفی و معیار تصمیم‌گیری بیشینه-بیشینه (مدل سافت‌مکس مستقل)؛

۳. شناسایی چهره با روش نزدیک‌ترین همسایه (KNN)

۴. شناسایی چهره با توزیع تصادفی و معیار تصمیم‌گیری نزدیک‌ترین همسایه؛

۵. شناسایی چهره با توزیع هوشمند (خوشه‌بندی) و معیار تصمیم‌گیری بیشینه-بیشینه؛

۶. شناسایی چهره با توزیع هوشمند (خوشه‌بندی) و معیار تصمیم‌گیری نزدیک‌ترین همسایه.

(جدول- ۱) دقت خوشه‌بندی بر اساس روش‌های گوناگون انتخاب نماینده خوشه

(TABLE- 1): CLUSTERING ACCURACY* ACCORDING TO CLUSTER REPRESENTATION TYPE

داده‌های آزمون (درصد)	داده‌های آموزشی (درصد)	تعداد خوشه‌ها	تعداد دسته‌ها	روش	نماینده خوشه
۲۳.۳۵	۲۳.۱۵	۵	۱۰۰۰۰	ISM	میانگین
۹۲.۶۱	۹۴.۸۸				بیشینه-بیشینه
۹۲.۷۱	۹۵.۲۴				نزدیک‌ترین
۸۵.۶۹	۸۶.۲۱			پیشنهادی	میانگین
۹۷.۱۸	۹۷.۸۹				بیشینه-بیشینه
۹۷.۷۵	۹۸.۴۶				نزدیک‌ترین
۱۱.۷۶	۱۱.۹۲	۱۰	۲۰۰۰۰	ISM	میانگین
۹۱.۳۹	۹۴.۱۲				بیشینه-بیشینه
۹۱.۵۹	۹۴.۴۳				نزدیک‌ترین
۷۷.۹۸	۷۸.۸۵			پیشنهادی	میانگین
۹۵.۷۲	۹۶.۸۱				بیشینه-بیشینه
۹۶.۲۵	۹۷.۳۸				نزدیک‌ترین

(جدول- ۲) مقایسه نتایج سناریوهای: الف) مدل بدون توزیع؛ ب) مدل سافت‌مکس مستقل (توزیع تصادفی و معیار انتخاب بیشینه-بیشینه)؛ ج) مدل توزیعی تصادفی بر اساس نزدیک‌ترین همسایه؛ د) مدل KNN؛ هـ) مدل توزیعی با خوشه‌بندی و معیار انتخاب بیشینه-بیشینه؛ و) مدل توزیعی خوشه‌بندی و معیار انتخاب نزدیک‌ترین همسایه (روش پیشنهادی). معیارهای ارزیابی: صحت، بازخوانی و f1. (استخراج‌گر ویژگی در همه سناریوها InceptionResnetV1[44] بوده است).

(Table I): Comparison results of the a) distribution-free network, b) independent softmax model, c) random distribution with nearest neighbor criteria, d) knn model, e) clustering based distribution with max-max criteria, f) clustering based distribution with nearest neighbor criteria (proposed method). Evaluation metrics are Precision, Recall, F1-score (The feature extractor in all scenarios was InceptionResnetV1 [44]

مجموعه‌دادگان		VGG-Face2 [33]					MS-Celeb-1M[34]				Glint360K[31]			
تعداد دسته‌ها		۵۰۰۰	۶۰۰۰	۷۰۰۰	۸۰۰۰	۸۹۰۰	۵۰۰۰	۱۰۰۰۰	۲۰۰۰۰	۵۰۰۰۰	۵۰۰۰	۱۰۰۰۰	۲۰۰۰۰	۵۰۰۰۰
مدل بدون توزیع	صحت	۸۹.۰۲	۸۸.۴۳	۸۷.۷۴	۸۷.۲۱	۸۶.۴۶	۹۳.۹۷	۹۰.۸۳	۸۸.۰۴	۷۴.۵۲	۹۵.۶۵	۹۳.۴۲	۸۹.۳۳	۷۳.۰۰
	بازخوانی	۸۷.۱۶	۸۶.۵۰	۸۵.۷۱	۸۵.۱۰	۸۴.۲۲	۹۲.۸۶	۸۹.۱۰	۸۵.۱۹	۶۵.۴۶	۹۴.۸۲	۹۲.۱۳	۸۶.۵۰	۶۳.۳۶
	F1	۸۶.۸۲	۸۶.۱۸	۸۵.۳۶	۸۴.۷۰	۸۳.۸۳	۹۲.۵۹	۸۸.۷۴	۸۴.۵۳	۶۳.۹۷	۹۴.۷۱	۹۱.۹۳	۸۵.۹۹	۶۱.۷۷
تعداد زیرشبکه‌ها		۲	۳	۳	۴	۴	۲	۵	۱۰	۲۵	۲	۵	۱۰	۲۵
مدل سافت‌مکس مستقل [۳۲]	صحت	۸۹.۳۸	۸۸.۹۳	۸۸.۴۳	۸۸.۱۳	۸۷.۶۶	۹۴.۴۳	۹۲.۴۱	۹۲.۰۱	۹۰.۵۰	۹۵.۹۰	۹۴.۷۷	۹۳.۴۰	۹۰.۸۷
	بازخوانی	۸۷.۶۰	۸۷.۱۳	۸۶.۵۹	۸۶.۲۰	۸۵.۷۰	۹۳.۴۲	۹۱.۰۹	۹۰.۶۵	۸۸.۹۲	۹۵.۱۴	۹۳.۷۹	۹۲.۱۸	۸۹.۲۰
	F1	۸۷.۳۰	۸۶.۸۲	۸۶.۲۴	۸۵.۸۶	۸۵.۳۵	۹۳.۲۱	۹۰.۸۱	۹۰.۳۴	۸۸.۵۲	۹۵.۰۲	۹۳.۶۵	۹۱.۹۹	۸۸.۹۴
مدل توزیعی تصادفی نزدیک‌ترین	صحت	۸۹.۳۷	۸۸.۹۸	۸۸.۵۷	۸۸.۲۶	۸۷.۸۴	۹۴.۴۳	۹۲.۳۷	۹۱.۹۵	۹۰.۴۳	۹۵.۹۰	۹۴.۶۳	۹۳.۴۳	۹۰.۹۹
	بازخوانی	۸۷.۶۲	۸۷.۱۵	۸۶.۶۸	۸۶.۳۱	۸۵.۸۷	۹۳.۴۶	۹۱.۰۶	۹۰.۶۴	۸۸.۹۹	۹۵.۱۶	۹۳.۶۵	۹۲.۲۵	۸۹.۴۳
	F1	۸۷.۳۷	۸۶.۸۹	۸۶.۴۱	۸۶.۰۳	۸۵.۵۷	۹۳.۲۶	۹۰.۷۹	۹۰.۳۳	۸۸.۵۶	۹۵.۰۶	۹۳.۵۰	۹۲.۰۶	۸۹.۱۶
مدل KNN	صحت	۸۹.۳۴	۸۸.۸۹	۸۸.۴۰	۸۸.۱۳	۸۷.۶۶	۹۴.۴۱	۹۲.۳۷	۹۱.۹۷	۹۰.۴۵	۹۵.۷۷	۹۴.۵۶	۹۳.۴۳	۹۱.۰۰
	بازخوانی	۸۷.۶۳	۸۷.۱۵	۸۶.۶۲	۸۶.۲۹	۸۵.۷۹	۹۳.۴۷	۹۱.۰۱	۹۰.۶۳	۸۸.۹۸	۹۴.۹۹	۹۳.۵۷	۹۲.۲۳	۸۹.۴۳
	F1	۸۷.۳۳	۸۶.۸۴	۸۶.۲۸	۸۵.۹۶	۸۵.۴۴	۹۳.۲۸	۹۰.۷۶	۹۰.۳۳	۸۸.۵۶	۹۴.۹۰	۹۳.۴۲	۹۲.۰۵	۸۹.۱۶
مدل توزیعی هوشمند بیش-بیش	صحت	**۸۹.۴۵**	۸۹.۰۳	۸۸.۵۵	**۸۸.۲۷**	۸۷.۸۷	۹۴.۵۲	۹۲.۶۰	۹۲.۱۷	**۹۰.۷۲**	۹۵.۸۲	**۹۴.۸۲**	**۹۳.۶۴**	**۹۱.۲۴**
	بازخوانی	۸۷.۶۷	۸۷.۲۶	۸۶.۷۲	۸۶.۳۵	۸۵.۹۱	۹۳.۵۱	۹۱.۳۲	۹۰.۸۵	۸۹.۱۸	۹۵.۰۶	۹۳.۸۶	۹۲.۴۶	۸۹.۶۵
	F1	۸۷.۳۶	۸۶.۹۳	۸۶.۳۸	۸۶.۰۱	۸۵.۵۶	۹۳.۲۹	۹۱.۰۳	۹۰.۵۳	۸۸.۷۹	۹۴.۹۷	۹۳.۷۲	۹۲.۲۷	**۸۹.۴۰**
مدل توزیعی هوشمند نزدیک‌ترین	صحت	۸۹.۴۳	**۸۹.۰۶**	**۸۸.۵۶**	۸۸.۲۶	**۸۷.۸۸**	**۹۴.۵۷**	**۹۲.۶۲**	**۹۲.۱۸**	۹۰.۶۷	**۹۵.۹۴**	**۹۴.۸۲**	۹۳.۶۳	۹۱.۲۰
	بازخوانی	**۸۷.۷۲**	**۸۷.۳۵**	۸۶.۷۹	۸۶.۴۶	۸۶.۰۲	۹۳.۵۷	۹۱.۳۹	۹۰.۹۱	**۸۹.۲۴**	۹۵.۱۶	۹۳.۸۷	**۹۲.۵۰**	**۸۹.۶۷**
	F1	۸۷.۴۰	۸۷.۰۲	۸۶.۴۴	۸۶.۱۲	۸۵.۶۶	۹۳.۳۶	۹۱.۱۰	۹۰.۵۸	۸۸.۸۱	۹۵.۰۷	۹۳.۷۳	۹۲.۳۲	۸۹.۴۰

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

(جدول- ۳) مقایسه زمان اجرای آموزش برای هر اپوک از تمام زیرشبکه‌ها

مقادیر جدول فوق، بر حسب ثانیه می‌باشند. سامانه مورد استفاده دارای کارت گرافیک NVIDIA GeForce GTX 1660 SUPER با ۶ گیگابایت حافظه، و پردازشگر Intel Core i7-6700K و ۳۲ گیگابایت حافظه رم بوده است.

Table- 3): comparison of training times for each epoch of all toilers*(

The values in the table are expressed in seconds. The system used for model execution includes an NVIDIA GeForce GTX 1660 SUPER GPU with 6GB of memory, an Intel Core i7-6700K CPU, and 32GB of RAM.

نرخ افزایش سرعت با توزیع سریال	توزیع سریال	نرخ افزایش سرعت با توزیع موازی	توزیع موازی	روش بدون توزیع	تعداد زیرشبکه‌ها	تعداد دسته‌ها
۱.۸۷۱	۷.۶۲۱	۳.۷۴۱	۳.۸۱۱	۱۴.۲۵۹	۲	۵۰۰۰
۲.۷۵۷	۱۰.۲۶۱	۸.۰۳۵	۳.۵۲۱	۲۸.۲۹۱	۳	۶۰۰۰
۲.۵۱۷	۱۵.۴۶۱	۷.۷۶۰	۵.۰۱۶	۳۸.۹۲۴	۳	۷۵۰۰
۴.۱۰۲	۱۶.۵۴۰	۱۶.۷۸۷	۴.۰۴۲	۶۷.۸۵۳	۴	۸۹۰۰
۳.۹۰۰	۱۸.۲۰۸	۱۹.۴۴۶	۳.۶۵۲	۷۱.۰۱۷	۵	۱۰۰۰۰
۷.۸۵۰	۳۹.۱۲۸	۷۸.۲۹۸	۳.۹۲۳	۳۰۷.۱۶۳	۱۰	۲۰۰۰۰
۱۷.۹۴۳	۹۱.۸۱۲	۴۵۳.۲	۳.۶۳۵	۱۶۴۷.۳۸۲	۲۵	۵۰۰۰۰

نتایج: نتایج بازشناسی چهره برای همه سناریوها در جدول (۲) خلاصه شده‌اند. برای آن که نتایج، منصفانه باشد و اثرات تصادف به حداقل برسد، آزمایش‌ها برای ۱۰ بار تکرار شدند و میانگین نتایج، در جدول گزارش شده‌است. نتایج متفاوت برای مجموعه‌دادگان مختلف، می‌تواند ناشی از تفاوت در سطح تمیزبودن یا دشواری تصاویر چهره مجموعه‌دادگان باشد.

زمانی که تعداد دسته‌ها اندک است، خطر بیش‌برازش کمتری وجود دارد و بنابراین مدل پیشنهادی و سایر روش‌های مبتنی بر توزیع، لزوماً عملکرد بهتری نسبت به روش بدون توزیع ندارند؛ اما هر چه که تعداد دسته‌ها افزایش می‌یابد، بهینگی روش پیشنهادی بیشتر خود را می‌نمایاند. با توجه به **Error! Reference source not found.**(۶) (مربوط به پایگاه دادگان VGGFace2) و **Error! Reference source not found.** (مربوط به پایگاه دادگان MS-Celeb-1M)، و **Error! Reference source not found.** (مربوط به پایگاه‌دادگان Glint360K)، نرخ دقت با افزایش اندازه در همه سناریوها کاهش می‌یابد، اما روش پیشنهادی کم‌ترین افت دقت را دارد.

زمان اجرا:

جدول مقایسه‌ای از زمان‌های اجرا برای هر اپوک برای همه سناریوها نمایش می‌دهد. نتایج نشان می‌دهند که زمان اجرای روش‌های توزیعی در حالت اجرای سریال، در حدود m برابر سریع‌تر از حالت اجرای بدون توزیع است. به علاوه اجرای موازی روش‌های مبتنی بر توزیع بر روی چند ماشین، حدوداً m^2 برابر سریع‌تر از حالت اجرای بدون توزیع است.

۵- بحث

روش پیشنهادی این مقاله، منابع مختلف خطا در مسائل دسته‌بندی با در نظر گرفتن ویژگی‌های استخراج‌شده از یک استخراج‌گر ویژگی مورد توجه قرار داده و تلاش کرده است که اثرات منفی ناشی از منابع گوناگون خطا را به کمینه برساند. منابع خطا شامل ویژگی‌های غیردقیق، مشکل بیش‌برازش، دسته‌های دشوار، توزیع نامناسب و چالش‌های تصمیم‌گیری هستند. مطالعه بر روی سناریوهای گوناگون دسته‌بندی، از جمله یک شبکه N دسته‌ای بدون توزیع و مدل‌های با توزیع تصادفی و هوشمند انجام گرفته است.

نتایج نشان می‌دهد که خطای ویژگی‌های غیردقیق در همه سناریوها به یکسان تأثیرگذار است. مشکل بیش‌برازش در سناریوی بدون توزیع بیشترین تأثیر را دارد، زیرا پیچیدگی شبکه در اینجا بیشتر است. دسته‌های دشوار در سناریوی توزیع هوشمند بهتر از سناریوی توزیع تصادفی تمایز داده می‌شوند.

توزیع نامناسب در سناریوی تصادفی تأثیر منفی دارد، اما در سناریوی توزیع هوشمند تأثیر کمتری دارد. چالش تصمیم‌گیری در هر دو روش توزیعی وجود دارد، اما در توزیع تصادفی این چالش بیشتر است.

روش پیشنهادی در این مطالعه، که از KMeans برای توزیع دسته‌ها بین خوشه‌ها استفاده می‌کند، در

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

مقایسه با روش‌هایی همچون تجزیه سافت‌مکس[۲۹] و انتخاب پویای دسته‌های فعال[۳۰] و موازی‌سازی در ArcFace [۲۸] دارای مزیت‌هایی است. برای مثال بر خلاف روش‌های به کار رفته در الگوریتم‌های مذکور، که برای موازی‌سازی و امکان‌پذیر شدن پیاده‌سازی در ابعاد بالا، در توابع زیان دست‌کاری می‌شد یا در فرآیند آموزش نیاز به انجام اقداماتی از قبیل انتخاب دسته‌های فعال بود، اما در مدل سافت‌مکس مستقل و نیز روش پیشنهادی مقاله، نیاز به هیچ اقدام اضافی در حین آموزش نیست. بین خوشه‌ها هیچ‌گونه ارتباطی در هنگام آموزش وجود ندارد. با توجه به مشکل موجود در روش مدل سافت‌مکس مستقل که در بخش کارهای مرتبط مقاله مفصل توضیح داده شده‌است، روش پیشنهادی، توزیع دسته‌ها میان خوشه‌ها را با استفاده از الگوریتم‌های خوشه‌بندی انجام می‌دهد و از سوی دیگر برای معیار تصمیم‌گیری، علاوه‌بر این که بیشینه بیشترین مقادیر خروجی مدل‌ها را در نظر می‌گیرد، مفهوم میزان شباهت داده آزمون به خوشه را نیز دخالت می‌دهد تا به حداکثر دقت دست یابد.

یکی از مزایای دیگر روش پیشنهادی این است که مدل و تابع هزینه استفاده شده در این مقاله، یعنی InceptionResnetV1 و ArcFace، می‌تواند با هر مدل و تابع هزینه دیگری جایگزین شود. دلیل استفاده از ArcFace در این مقاله، این بوده است که نسبت به بسیاری از روش‌های رقیب خود، از دقت بالاتری برخوردار است.

شایان ذکر است که هر دو روش پیشنهادی و مدل سافت‌مکس مستقل، پیچیدگی‌های محاسباتی اندکی دارند و قابلیت موازی‌سازی بالایی دارند که در مقایسه با روش‌های غیرتوزیعی، زمان اجرای بسیار سریع‌تری در گام آموزش، نسبت به روش‌های غیرتوزیعی دارند.

۶- نتیجه‌گیری

کار اصلی این مقاله در دو قسمت پیش‌پردازش و پس‌پردازش بوده است. در عملیات آموزش هیچ نوعی از پیچیدگی افزوده نشده‌است. برای حل مشکل افت دقت و چالش سرعت و حافظه در دسته‌بندی مسائل با دسته‌های بالا، رویکرد تجزیه به زیرمسئله‌های کوچک‌تر اتخاذ گردیده است. تجزیه و توزیع دسته‌ها، از طریق الگوریتم خوشه‌بندی KMeans انجام شده‌است. برای ترکیب نتایج زیرشبکه‌ها، معیار جدیدی پیشنهاد شده‌است. این معیار آمیخته‌ای از معیار بیشینه-بیشینه، و نیز شباهت داده آزمون با ویژگی‌های هر دسته است. به عبارت دیگر، دسته برگزیده، دسته‌ای است که برآیند خروجی شبکه و شباهتش با داده آزمون، بیشترین مقدار را داشته باشد.

یکی از اقدامات رو به جلو در آینده، این است که خطای خوشه‌بندی به حداقل مقدار خود برسد تا به همان میزان، دقت مجموع دسته‌بندی برای تعداد دسته‌های بالا، به مقدار بیشتری برسد. همچنین می‌توان مدل پایه (استخراج‌گر ویژگی) را بازآموزی کرد تا ویژگی‌های مقاوم‌تری را در اختیار قرار دهد.

۷- منابع 7-Refrence

[1] Sun, Y., et al., Deep learning face representation by joint identification-verification. Advances in neural information processing systems, 2014. 27.

[2] Schroff, F., D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[3] Boutros, F., et al., Synthetic data for face recognition: Current state and future prospects. Image and Vision Computing, 2023: p. 104688.

[4] Alansari, M., et al., GhostFaceNets: Lightweight Face Recognition Model From Cheap Operations. IEEE Access, 2023. 11: p. 35429-35446.

[5] Ahmadi, MA., Dianat, R., Introducing a method for extracting features from facial images based on applying transformations to features obtained from convolutional neural networks. Signal and Data Processing, 2020. 17(3): p. 141-156.

احمدی، مرتضیعلی، و دیانت، روح الله. (۱۳۹۹). ارائه یک روش استخراج ویژگی از تصاویر چهره مبتنی بر اعمال تبدیل روی ویژگی های به دست آمده از شبکه های عصبی کانولوشن. پردازش علایم و داده ها، ۱۷(۳ (پیاپی ۴۵))، ۱۴۱-۱۵۶.

[6] Yang, S., et al., Faceness-net: Face detection through deep facial part responses. IEEE transactions on pattern analysis and machine intelligence, 2017. 40(8): p. 1845-1859.

[7] Zhang, K., et al., Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 2016. 23(10): p. 1499-1503.

[8] Hu, P. and D. Ramanan. Finding tiny faces. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[9] Mamieva, D., et al., Improved face detection method via learning small faces on hard images based on a deep learning approach. Sensors, 2023. 23(1): p. 502.

[10] Si, T., F. He, and P. Li, Hybrid feature constraint with clustering for unsupervised person re-identification. The Visual Computer, 2022.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]

[11] Zhao, W., et al., Face recognition: A literature survey. ACM computing surveys (CSUR), 2003. 35(4): p. 399-458.
[12] Campadelli, P., R. Lanzarotti, and C. Savazzi. A feature-based face recognition system. in 12th International Conference on Image Analysis and Processing, 2003. Proceedings. 2003. IEEE.
[13] Bern, M. and D. Eppstein, Mesh generation and optimal triangulation, in Computing in Euclidean geometry. 1995, World Scientific. p. 47-123.
[14] Yim, J., et al. Rotating your face using multi-task deep neural network. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[15] Jin, X. and X. Tan, Face alignment in-the-wild: A survey. Computer Vision and Image Understanding, 2017. 162: p. 1-22.
[16] Galbally, J. and S. Marcel. Face anti-spoofing based on general image quality assessment. in 2014 22nd international conference on pattern recognition. 2014. IEEE.
[17] Galbally, J., S. Marcel, and J. Fierrez, Biometric antispoofing methods: A survey in face recognition. IEEE Access, 2014. 2: p. 1530-1552.
[18] Yu, Z., et al. Flexible-modal face anti-spoofing: A benchmark. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
[19] Zhou, E., Z. Cao, and Q. Yin, Naive-deep face recognition: Touching the limit of LFW benchmark or not? arXiv preprint arXiv:1501.04690, 2015.
[20] Jalal, A.S., D.K. Sharma, and B. Sikander, Suspect face retrieval using visual and linguistic information. The Visual Computer, 2022: p. 1-27.
[21] Nasution, M.I.P., et al. Face recognition login authentication for digital payment solution at COVID-19 pandemic. in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE). 2020. IEEE.
[22] Talahua, J.S., et al., Facial recognition system for people with and without face mask in times of the covid-19 pandemic. Sustainability, 2021. 13(12): p. 6900.
[23] Wang, F., et al., NormFace: L2 Hypersphere Embedding for Face Verification, in Proceedings of the 25th ACM international conference on Multimedia. 2017, Association for Computing Machinery: Mountain View, California, USA. p. 1041–1049.
[24] Liu, W., et al., Large-margin softmax loss for convolutional neural networks. arXiv preprint arXiv:1612.02295, 2016.
[25] Liu, W., et al. Sphereface: Deep hypersphere embedding for face recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[26] Wang, F., et al. Normface: L2 hypersphere embedding for face verification. in Proceedings of the 25th ACM international conference on Multimedia. 2017.
[27] Wang, H., et al. Cosface: Large margin cosine loss for deep face recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[28] Deng, J., et al. Arcface: Additive angular margin loss for deep face recognition. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
[29] He, L., et al. Softmax dissection: Towards understanding intra-and inter-class objective for embedding learning. in Proceedings of the AAAI Conference on Artificial Intelligence. 2020.
[30] Zhang, X., et al. Accelerated training for massive classification via dynamic class selection. in Proceedings of the AAAI Conference on Artificial Intelligence. 2018.
[31] An, X., et al. Partial fc: Training 10 million identities on a single machine. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[32] Wu, Y., et al. Deep convolutional neural network with independent softmax for large scale face recognition. in Proceedings of the 24th ACM international conference on Multimedia. 2016.
[33] Cao, Q., et al. Vggface2: A dataset for recognising faces across pose and age. in 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). 2018. IEEE.
[34] Guo, Y., et al. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. 2016. Springer.
[35] Hadsell, R., S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). 2006. IEEE.
[36] Nguyen, B., C. Morell, and B. De Baets, Distance metric learning for ordinal classification based on triplet constraints. Knowledge-Based Systems, 2018. 142: p. 17-28.
[37] Wen, Y., et al. A discriminative feature learning approach for deep face recognition. in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. 2016. Springer.
[38] Wang, K., et al. An efficient training approach for very large scale face recognition. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[39] Liu, Y. and Q. Liu. Convolutional neural networks with large-margin softmax loss function for cognitive load recognition. in

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.61186/jsdp.21.1.53]

2017 36th Chinese control conference (CCC). 2017. IEEE.
[40] Dietterich, T.G. and G. Bakiri, Solving multiclass learning problems via error-correcting output codes. Journal of artificial intelligence research, 1994. 2: p. 263-286.
[41] Zhang, Q., et al. Large scale classification in deep neural network with label mapping. in 2018 IEEE International Conference on Data Mining Workshops (ICDMW). 2018. IEEE.
[42] Xu, Y., et al. High performance large scale face recognition with multi-cognition softmax and feature retrieval. in Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017.
[43] He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[44] Tim, E. Face Recognition Using Pytorch. 2022 2023/11/02; Available from: https://github.com/timesler/facenet-pytorch.



سید محمد احمدی، دانشجوی دکترای رشته مهندسی فناوری اطلاعات دانشگاه قم است. از مهم‌ترین علاقه‌مندی‌های ایشان می‌توان به داده‌کاوی، یادگیری عمیق، پردازش تصویر و چهره اشاره کرد.
نشانی رایانامه ایشان عبارت است از:
sm.ahmadi@stu.qom.ac.ir



روح الله دیانت در حال حاضر، استادیار گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه قم است. از علاقه‌مندی‌های ایشان می‌توان به پردازش سیگنال، پردازش گفتار، پردازش تصویر و بازشناسی الگو اشاره کرد.
نشانی رایانامه ایشان عبارت است از :
rdianat@qom.ac.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.61186/jsdp.21.1.53]