

تشخیص تخلف رانندگان با استفاده از نگاشت/کاهش

شیوا اسدیانفام^۱، محبوبه شمسی^۲، عبدالرضا رسولی کناری^۳

تاریخ دریافت: ۱۳۹۷/۱۱/۱۶

تاریخ پذیرش: ۱۳۹۸/۰۵/۱۲

چکیده

کاهش معضلات حمل و نقل در خصوص ایمنی، ترافیک و به تبع آن تأثیرات انسانی، اجتماعی و اقتصادی یکی از چالشهای اصلی جوامع بشری محسوب می‌شود. افزایش تخلف رانندگی و اثرات مضر آن یکی از مهمترین مسائل اجتماعی در کشورهای مختلف است داده‌های چند رسانه‌ای/ غیرساختیافته در چند سال گذشته رشد شگفت‌انگیزی داشته است. سهم کلان داده و تشخیص رفتار در این رابطه، بیشتر با کنترل مکانیزه تخلف رانندگان مطرح می‌شود. در سیستم کنونی کنترل توسط نیروی انسانی (پلیس یا اپراتور) صورت می‌پذیرد. هدف از این تحقیق کنترل مکانیزه تخلف رانندگان مبتنی بر معماری توزیع شده و مدل برنامه‌نویسی نگاشت/کاهش (Map / Reduce) است که باعث کاهش زمان پردازش با استفاده از ابزار هادوپ می‌باشد. به منظور بررسی و تست سیستم پیشنهادی از اطلاعات دوربین‌های موجود در سطح شهر استفاده شده است که از مرکز کنترل ترافیک راهنمایی و رانندگی شهرستان یزد در ایران بدست آمده است الگوریتم ارائه شده در این مقاله بدون حضور اپراتور ۲۴ ساعته قادر به تشخیص رفتارهای نامناسب رانندگان است. بنابراین در صورت عدم حضور اپراتور نیز رفتار نادرست و تخلف رانندگان دور از چشم نخواهد بود. نتایج بیانگر این نکته است که زمان پردازش برای مجموعه داده‌های بزرگ با استفاده از هادوپ و اجرای برنامه فقط با یک گره پیرو نسبت به اجرای برنامه ترتیبی بیش از ۷۰ درصد کاهش پیدا می‌کند. همچنین کارایی سیستم با افزایش تعداد گره‌های داده بیش از ۷۵ درصد افزایش پیدا می‌کند.

کلمات کلیدی: تشخیص رفتار رانندگان، ایمنی حمل و نقل، کنترل دوربین‌های نظارتی، تخلفات ترافیکی، رفتارهای نایمن، نگاشت/کاهش

^۱ دانشجوی دکتری، گروه کامپیوتر، واحد قم، دانشگاه آزاد اسلامی، قم، ایران sh_asadianfam@yahoo.com

^۲ استادیار، دانشکده برق و کامپیوتر، دانشگاه صنعتی قم، قم، ایران shamsi@qut.ac.ir

^۳ استادیار، دانشکده برق و کامپیوتر، دانشگاه صنعتی قم، قم، ایران rasouli@qut.ac.ir

۱- مقدمه

هر اقدام پژوهشی و تحقیقاتی که در این مسیر انجام شود، برای جامعه مفید است.

کلان داده‌ها، اصطلاحی است که به مجموعه داده‌هایی اطلاق می‌شود که مدیریت، کنترل و پردازش آنها فراتر از توانایی ابزارهای نرم‌افزاری در یک زمان قابل تحمل و مورد انتظار است [۱۱]. از کاربردهای تحلیل کلان داده در زمینه ترافیک و ترابری می‌توان به تحلیل دوربین‌های نظارتی شهری اشاره کرد. به علت وجود اعمال پر حادثه در سطح شهرها و جاده‌ها و لزوم کنترل و بررسی رفتار رانندگان به منظور اتخاذ تدابیر لازم جهت افزایش ایمنی و آسیب‌های ناشی از سوانح رانندگی، و همچنین بدلیل اینکه ویدئوهای دوربین‌های نظارتی داده‌هایی با جریان سریع، حجیم و متنوع تولید می‌کنند می‌توان از آنها بعنوان کلان داده یاد کرد. نگاشت/کاهش یک مدل برنامه نویسی برای پردازش داده در مقیاس‌های بالا است که با پردازش داده بصورت توزیع‌شده باعث کاهش زمان پردازش می‌شود. بدین منظور این مقاله روشی مبتنی بر نگاشت/کاهش برای تشخیص رفتار رانندگان در تصاویر با مقیاس و حجم بزرگ ارائه می‌دهد. هادوپ یک تکنولوژی رایج برای پردازش کلان داده است. از هادوپ و تکنیک نگاشت/کاهش در حوزه‌های مختلفی اعم از پزشکی، هواشناسی، تشخیص رفتار، پایگاه داده، پیاده‌سازی الگوریتم‌های مختلف یادگیری ماشین با تکنیک نگاشت/کاهش برای تجزیه و تحلیل کلان داده‌ها و ... استفاده شده است که برخی از کارهای انجام شده در ذیل مرور می‌شود.

بدلیل اینکه روند تجزیه و تحلیل زیبایی چهره مخصوصا برای تعداد زیادی از تصاویر بسیار پیچیده و زمان گیر است، در مقاله [۱۲] یک سیستم تجزیه و تحلیل زیبایی چهره با استفاده از هادوپ ارائه شده است که با بهره‌گیری از تکنیک نگاشت/کاهش باعث کاهش زمان پردازش تصاویر با پیاده‌سازی چندین slave شده

راننده به عنوان عامل انسانی با اخذ رفتارهای گوناگون باعث وقوع تخلف و در پی آن آسیب می‌شود. برای کنترل تخلف و اعمال رانندگان، مامورین پلیس راهنمایی و رانندگی در مبادی ورودی مستقر می‌گردند و خودروهای متخلف را متوقف می‌نمایند. در این روش بدلیل اینکه کنترل توسط انسان (مامور پلیس) و بصورت مشاهده انجام می‌گیرد، امکان خطا و سوء استفاده وجود دارد که می‌توان به مواردی همچون تبانی- پنهان ماندن از دید پلیس- عدم حضور پلیس- ایجاد ترافیک در مبادی ورودی و ... اشاره نمود. مقالات بسیار زیادی درمورد رفتار راننده ارائه شده است [۱-۵].

خطر پتانسیلی جهت ایجاد جراحت می‌باشد. در یک اصطلاح کلی خطر با شرایط یا فعالیت کنترل نشده که می‌تواند آسیب تولید کند ارتباط دارد. تشخیص خطرات و حذف یا کنترل آنها به ما امکان جلوگیری از وقوع آسیبها را می‌دهد. هزینه اجتماعی جراحت و تلفات سوانح رانندگی هنوز غیرقابل قبول است. رانندگان اغلب مسئول ایجاد سوانح و آسیب‌ها هستند [۶]، بنابراین تغییر رفتار راننده یکی از مهم ترین و چالش برانگیز مسائل در حمل و نقل محسوب می‌شود.

در سال‌های اخیر با افزایش تعداد خودروها در سطح خیابان‌ها و جاده‌ها، مشکلات کنترل ترافیک و اجرای قوانین رانندگی بیش از پیش مطرح می‌گردند. بعضی از کشورها راه کارهایی را اندیشیده اند و سیستمهایی را بوجود آورده اند تا بتوانند بطور اتوماتیک و مکانیزه ترافیک شهری را تحت کنترل خود درآورند [۷-۱۰]. همچنین با توجه به نصب دوربین‌های مختلف در چهارراه‌ها و مسیرهای پر رفت و آمد، وجود سیستم‌های کارآمد تشخیص خودکار رفتار مخاطره‌آمیز رانندگان، بسیار ضروری و مهم به نظر می‌رسند. با توجه به اهمیت موضوع،

تشخیص تخلف رانندگان با استفاده از نگاشت/کاهش است. تجزیه و تحلیل سری زمانی و تجزیه و تحلیل پیش بینی با استفاده از چارچوب MapReduce در مقاله [۱۳] آمده است. در این مقاله [۱۳] مدل سازی داده های هواشناسی بر روی پلت فرم داده های بزرگ با استفاده از رویکرد MapReduce انجام شده است. نتایج تحلیل داده های هواشناسی مبتنی بر تکنیک نگاشت/کاهش بصورت سریع و با سرعت بالا در این مقاله تحقق یافته است. همچنین در [۱۴] نیز الگوریتم نگاشت/کاهش برای پردازش داده های بزرگ در حوزه هواشناسی ارائه شده است.

در مقاله [۱۵] یک الگوریتم خوشه بندی کلان داده مبتنی بر نگاشت/کاهش با استفاده از کولونی زنبور عسل معرفی شده است که نتایج حاکی از آن است که کارایی خوشه بندی مبتنی بر نگاشت/کاهش در مقایسه با خوشه بندی k-means بهبود یافته است. از MapReduce برای خوشه بندی احتمالی موخر و مدل های مرتبط برای توصیه گر ها استفاده شده است [۱۶]. رفتار مقیاس پذیری خوب با اجرای توزیع شده بر روی گره های مختلف در این مقاله به چشم می خورد. یک الگوریتم TFIDF بهبود یافته برای بازیابی اخبار رویترز و استخراج اطلاعات متنی در حجم عظیمی از داده ها با چارچوبی منحصر بفرد به ترتیب در مقاله [۱۷، ۱۸] معرفی شده است. در مقاله [۱۷] بیان شده است که عملکرد TFIDF مبتنی بر نگاشت/کاهش برای طبقه بندی اخبار، وزن دهی به کلمات و خوشه بندی بهبود پیدا کرده است. همچنین، الگوریتم SVM مبتنی بر نگاشت/کاهش برای حاشیه نویسی (برچسب دهی) تصاویر در مقاله [۱۹] وجود دارد. الگوریتم پیشنهادی توزیع شده در مقاله [۱۹] با تقسیم مجموعه داده های آموزشی به زیر مجموعه های کوچکتر و بهینه سازی زیر مجموعه های تقسیم شده در یک گروه از رایانه ها، زمان آموزش را به طور قابل توجهی کاهش می دهد، در حالی که از حفظ سطح بالایی از دقت نیز برخوردار است. محاسبه اعداد مرزی مکعب داده با استفاده از تکنیک

نگاشت/کاهش برای انجام محاسبات با مرتبه کمتر در مقاله [۲۰] بیان شده است. در حوزه پزشکی نیز از تکنیک نگاشت کاهش برای پردازش داده های بزرگ استفاده شده است [۲۱، ۲۲]. یک چارچوب جدید تجزیه و تحلیل داده ها برای شهرهای هوشمند نیز وجود دارد [۲۳] که در آن در مورد رویکردهای کشف دانش در سیستم های سنتی در مقابل سیستم های کلان داده بحث شده است. تشخیص رفتار دانش آموزان با فریم ورک کلان داده نیز در [۲۴] بیان شده است. در مقاله [۲۵] از یک شبکه سنسور بی سیم استفاده شده است و هدف آن نظارت بر سطح آلودگی هوا در شهر با استفاده از کلان داده است. در مقاله [۲۶] از کلان داده ها برای شمارش عابر پیاده استفاده شده است. در مقاله [۲۷] پیش بینی سرعت ترافیک مبتنی بر شبیه سازی مونت کارلو با استفاده از تاریخچه کلان داده ها انجام شده است. با این حال آن در حوزه حمل و نقل و تحلیل تصاویر دوربین های نظارتی استفاده نشده است.

در این مقاله ما سیستم تحلیل رفتار مخاطره آمیز رانندگان حادثه ساز را بر اساس هادوپ و تکنیک نگاشت/کاهش برای ارائه راه حل های موثر بکار گرفته ایم. سیستم پیشنهادی تشخیص رفتار رانندگان برای سرعت بخشیدن به زمان پردازش مجموعه داده های بزرگ و متنوع است. سیستم به گونه ای طراحی شده است که می تواند رفتار متخلفین حادثه ساز در داده های با مقیاس بزرگ را با فریم ورک هادوپ با استفاده از تکنیک توزیع شدگی کشف کند.

در این مقاله در بخش ۲، مطالب پیش زمینه ای را که برای درک روش ارائه شده در مقاله ضروری است توضیح می دهیم. در بخش ۳، روند کلی کار این مقاله به همراه تابع MapReduce مورد استفاده ارائه می شود. مولفه ها و نحوه ی عملکرد هر بخش از کار توضیح داده می شود. در بخش ۴، تنظیمات انجام شده برای اجرا و نتایج حاصل از کار

ارائه شده و در نهایت در بخش ۵، به یک نتیجه گیری کلی از کار انجام گرفته پرداخته می‌شود.

۲- پیش زمینه

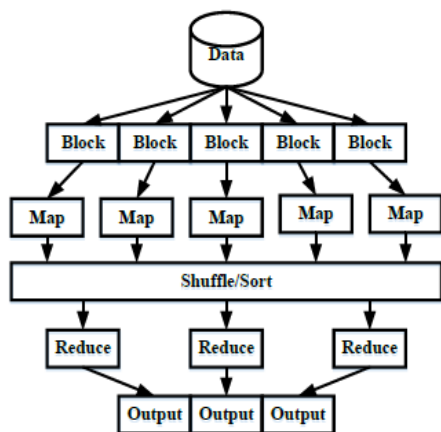
۱-۲ کلان داده

کلان داده‌ها (Big Data)، روش‌ها و فناوری‌های نوینی را جهت جمع‌آوری، ذخیره و آنالیز داده‌های غیرساختیافته بصورت مقیاس‌پذیر معرفی می‌کند [۲۸، ۲۹]. کلان داده‌ها ممکن است از صنعت، کشاورزی، ترافیک و ترابری، مراقبت‌های پزشکی، سازمان‌های عمومی، خانواده‌ها و غیره به دست آمده باشد. در ۲۰ سال گذشته در حوزه‌های مختلف، حجم داده‌ها به مقدار زیادی افزایش یافته است. طبق گزارش موسسه بین‌المللی داده در سال ۲۰۱۱ حجم داده‌های ایجاد شده در سراسر دنیا ۱٫۸ زتا بایت (10^{21} بایت) بود [۳۰] که این مقدار تا سال ۲۰۲۰ به بیش از ۳۵ زتا بایت خواهد رسید [۳۱]. داوگلانی تحلیلگری از متا در یک گزارش تحقیقاتی، فرصت‌ها و چالش‌های ایجاد شده از داده‌های افزایش یافته را با مدل 3V (افزایش حجم- سرعت- تنوع) تعریف کرد [۳۲].

زنجیره‌ی ارزش کلان داده‌ها را می‌توان بطور کلی به چهار مرحله تقسیم کرد: تولید داده‌ها، اکتساب داده‌ها، ذخیره سازی داده‌ها، تحلیل داده‌ها [۱۱]. منظور از فاز تولید کلان داده‌ها، منابع اصلی کلان داده‌هاست که شامل داده‌ها در زمینه‌های مختلف با فناوری‌های متفاوت است. اکتساب کلان داده‌ها شامل جمع‌آوری داده‌ها، انتقال داده‌ها به زیرساخت‌های ذخیره سازی و پیش پردازش داده‌ها است. در مرحله ذخیره‌سازی داده‌ها از مکانیسم‌های مختلفی اعم از ذخیره‌سازی بر روی HDFS و نوع مدل برنامه‌نویسی مانند نگاشت/کاهش استفاده می‌شود. در فاز تحلیل کلان داده‌ها نیز از متدهای مربوط تحلیل کلان داده‌ها همچون بلوم فیلتر، درخت پیشوندی و غیره استفاده می‌شود [۳۳].

۲-۲ هادوپ و تکنیک نگاشت/کاهش

هادوپ این امکان را در اختیار ما قرار می‌دهد تا اپلیکیشن‌هایی را روی سیستم‌هایی مجهز به هزاران گره و حاوی هزاران ترابایت داده به اجرا درآوریم. هادوپ از دو بخش کلی به نام MapReduce و HDFS تشکیل شده است. این سیستم در واقع جهت اجرا بر روی چندین سرور طراحی شده است. بخش Map Reduce بر روی سرور اصلی اجرا می‌شود و بخش HDFS بر روی سرورهای جانبی اجرا می‌شود. سیستم فایل توزیعی هادوپ (HDFS) [۳۴] این امکان را فراهم می‌آورد تا سرعت انتقال داده‌ها در میان گره‌ها افزایش یابد و سیستم بتواند در صورت از کارافتادگی یک گره همچنان بی‌وقفه به کار خود ادامه بدهد. این رویکرد عملاً خطر فاجعه‌بار خرابی سیستم را حتی وقتی تعداد قابل ملاحظه‌ای از گره‌ها غیرفعال می‌شوند، کاهش می‌دهد.



شکل ۱. مدل برنامه‌نویسی نگاشت/کاهش

MapReduce یک چارچوب نرم‌افزاری است که در آن برنامه‌ها به بخش‌های کوچک‌تر تقسیم می‌شوند. هر کدام از این بخش‌ها (که همچنین بخش یا بلوک نامیده می‌شوند) می‌تواند روی هرگره‌ای در یک مجموعه خوشه‌ای از گره‌ها اجرا شود [۳۵]. در واقع MapReduce یک مدل برنامه‌نویسی موازی جهت پردازش داده‌های روی کلاسترها می‌باشد که از دو فاز اصلی شامل فاز نگاشت و فاز کاهش تشکیل شده است. شکل ۱، مدل

تشخیص تخلف رانندگان با استفاده از نگاشت/کاهش نگاشت/کاهش را در حالت کلی نشان می‌دهد. در فاز نگاشت، نود Master ورودی را دریافت می‌کند و آن را به قسمت های کوچکتر تبدیل می‌نماید و آن ها را بین نودهای worker تقسیم می‌کند. نود worker خود ممکن است این مرحله را تکرار کند و در نتیجه یک ساختار درختی به وجود می‌آید. در فاز کاهش، نود Master جواب قسمت های کوچک شده را دریافت می‌کند و آنها را با یکدیگر ترکیب می‌کند تا خروجی مورد نظر تشکیل شود [۳۶].

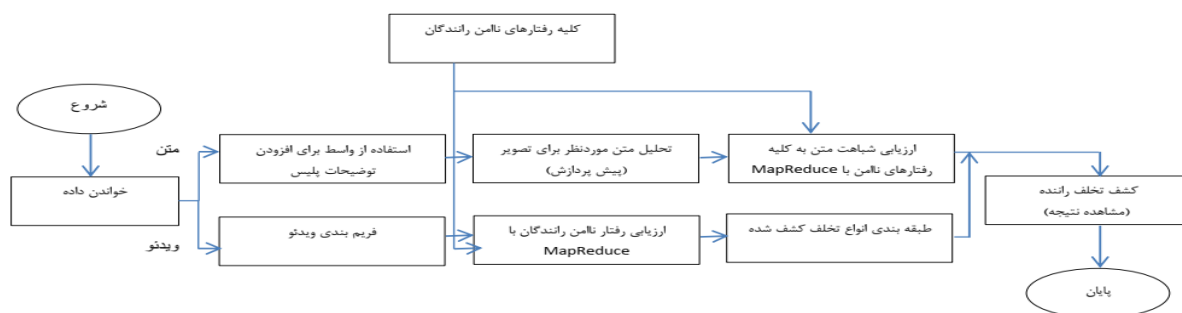
۳. تشخیص رفتار مبتنی برنگاشت/کاهش

ایده اصلی در این مقاله، استفاده از MapReduce برای تشخیص رفتارهای رانندگان است. شکل ۲، پروسه تشخیص رفتار متخلفین حادثه ساز را نشان می‌دهد. داده‌های این سیستم شامل تحلیل داده‌های غیر ساختیافته و تحلیل داده‌های چند رسانه‌ای (شامل فایل‌های تصویری و فیلم ها) است. داده ساختیافته داده‌ای است که دارای یک قالب قابل پیش‌بینی و منظم است. داده ساختیافته شامل رکوردها، فیلدها، کلیدها و ایندکس‌ها است. داده غیر ساختیافته داده‌ای است که غیر قابل پیش‌بینی است و دارای ساختاری نمی‌باشد که قابل تشخیص برای کامپیوتر باشد. دستیابی به داده غیر ساختیافته مشکل است، خصوصاً زمانی که رشته‌های طولانی از داده لازم باشد به طور ترتیبی جستجو شوند (parsing) تا از بطن آن یک واحد داده را پیدا کرد. شکل‌های متفاوتی از داده غیر ساختیافته وجود دارد و شاید متداولترین شکل موجود، تصویر است. در تحلیل داده‌های پلیس می‌توان اینطور بیان کرد که پلیس در صورت مشاهده رفتار نامناسب رانندگان از آن عکس گرفته و با استفاده از واسطی برای تصویر موردنظر برجسبی قرار می‌دهد. این برجسب حاکی از نوع رفتار راننده به‌مراه توضیحات کامل از آن است.

در گام بعدی، عملیات پیش پردازش به منظور استخراج داده‌های مفید انجام می‌شود. رفتار رانندگی را می‌توان به چند بخش تقسیم کرد، تخلفات حرکتی، تخلفات پارک، تخلفات تجهیزاتی، غیره. پس از انجام پیش پردازش، می‌بایست داده‌های مفید استخراج شده با تمامی رفتارهای از پیش تعریف شده‌ی اداره راهنمایی و رانندگی (تخلفات حرکتی، تخلفات پارک و غیره) مقایسه شود. ارزیابی و بررسی شباهت توضیحات پلیس با تمامی رفتارهای ناامن توسط تابع Map و Reduce موجود در شکل ۳ انجام می‌گیرد.

همانطور که در شکل ۳ مشاهده می‌کنید، به منظور ارزیابی شباهت موردنظر از معیار شباهت TF-IDF [۳۷] استفاده شده است (رابطه ۱). الگوریتم با پارامترهای ورودی تمامی رفتارهای از پیش تعریف شده و برجسب متنی پلیس آغاز می‌شود. در تابع Map، بردار فراوانی (تعداد تکرار) عبارات پلیس و داده‌های رفتاری از پیش تعیین شده راهنمایی و رانندگی استخراج شده و معیار کسینوسی (شباهت) برای آن‌ها محاسبه می‌گردد. تابع Reduce، تمامی مقادیر محاسبه شده را دریافت کرده و بزرگترین مقدار را بعنوان نوع رفتار و تخلف راننده اعلام می‌کند. بدین ترتیب خطاهای انسانی از قبیل خطای دید و تشخیص دقیق نوع رفتار تحقق می‌یابد. در فاز بعدی از شکل ۲، اطلاعات ویدئویی که توسط دوربین های نظارت تصویری به مرکز کنترل ترافیک می‌رسند، در گام نخست به تصویر یا فریم تبدیل شده و در گام دوم تشخیص رفتار راننده با استفاده از تابع MapReduce انجام می‌گیرد. شبه کد مربوط به تشخیص مکانیزه رفتار راننده از ویدئو در شکل ۴ توضیح داده شده است.

$$\cos(\theta) = \frac{d_1 \cdot d_2}{|d_1| |d_2|} \quad \text{رابطه (۱)}$$



شکل ۲. فلوجارت تشخیص رفتار مبتنی بر Map/Reduce

الگوریتم ۱. شباهت توضیحات پلیس و رفتارها

```

1: Class Mapper
2:  method Map (doc Behaviours, doc Police-Desc)
3:    for all term t1 ∈ array(line-of- Behaviours) do
4:      If is not Exist in array(line-of- Behaviours)
5:        Frequency[t1] ← 1
6:      else
7:        Frequency[t1]++
8:    end if
9:    pow1 ← 0
10:   pow2 ← 0
11:   sum ← array(Police-Desc)* array(Frequency)
12:   for all term t1 ∈ array(Police-Desc) do
13:     pow1 ← pow1 + pow(t1,2)
14:     for all term t2 ∈ array(Frequency) do
15:       pow2 ← pow2 + pow(t2,2)
16:     distance ← sum/(sqrt(pow1)*sqrt(pow2))
17:     Emit(doc line-of- Behaviours, Count distance)
18:
19: Class Reducer
20:  method Reduce (doc line-of- Behaviours, Count [distance1,distance2,...])
21:    Max ← distance1
22:    for all Count distance ∈ Count [distance2,distance3,...] do
23:      Max ← Find-Max[Max,distance]
24:      Max-S ← Line-Max[distance]
25:    end for
26:    Emit(doc Max-S, Count Max)

```

شکل ۳. شبه کد مربوط به تابع نگاشت/کاهش برای تعیین شباهت توضیحات پلیس و رفتارهای از پیش تعریف شده راهنمایی و رانندگی

الگوریتم ۲. تشخیص مکانیزه رفتار راننده

```

1: Class Mapper
2:  method Map (Initial Images)
3:    x ← pic.height()
4:    y ← pic.width()
5:    Creat-Empty-Image(x,y)
6:    GrayScale (Initial Images)
7:    Lane ← Lane-Detection(pic)
8:    sign ← Traffic-Sign-Detection(pic)
9:    for all Element e ∈ arysign do
10:      If (sign == arysign) then
11:        Text1 ← name-of-TrafficSign

```

```

12:         end if
13:     end for
14:     Light ←Traffic-Light-Detection(pic)
15:     If (Background- of-light == Red) then
16:         Text2← "RED"
17:     else If (Background- of-light == Green) then
18:         Text2← "GREEN"
19:     else If (Background- of-light == Yellow) then
20:         Text2← "YELLOW"
21:     end If
22:     Line ←Pedestrian-Lines(Lane-pic)
23:     Grid ←Grid-Lines(Lane-pic)
24:     If (Text2 == "RED") OR (Text2=="YELLOW") then
25:         Vehicle1←Vehicle-Detection(Line-pic)
26:     end If
27:     if (vehicle1 is not empty ) AND (Text2=="RED") then
28:         Text3← "crossing prohibited places or red lights "
29:     end If
30:     x1←Lane.height()
31:     y1←Lane.width()
32:     Creat-Empty-Image(x1,y1)
33:     Lane1← ((x1)/4,y1)
34:     Vehicle2←Vehicle-Detection(Lane1)
35:     If ((Text1 == "No Stop") OR (Text1==" Carrying by crane ")) AND
    (vehicle2 is not empty) then
36:         Text4← "stopping in prohibited places"
37:     end If
37:     Lane1← ((x1)/2,y1)
38:     Vehicle2←Vehicle-Detection(Lane1)
39:     If (Text1 == "No entry ") AND (vehicle2 is not empty) then
40:         Text5← " No entry "
41:     end If
42:     If (Text1 == "Stop absolutely prohibited") AND (vehicle2 is not empty)
    then
43:         Text6← " Stop absolutely prohibited"
44:     end If
45:     Emit(Final Picture, Text1, Text2, Text3, Text4, Text5, Text6)
1: Class Reducer
2:     method Reduce (Final Picture, Text1, Text2, Text3, Text4, Text5, Text6)
3:         If (Text3 is not empty) then
4:             Emit(Text2+Text3, Final Picture)
5:         end If
6:         If (Text4 is not empty) then
7:             Emit(Text1+Text4, Final Picture)
8:         end If
9:         If (Text5 is not empty) then
10:             Emit(Text1+Text5, Final Picture)
11:         end If
12:         If (Text6 is not empty) then
13:             Emit(Text1+Text6, Final Picture)
14:         end If

```

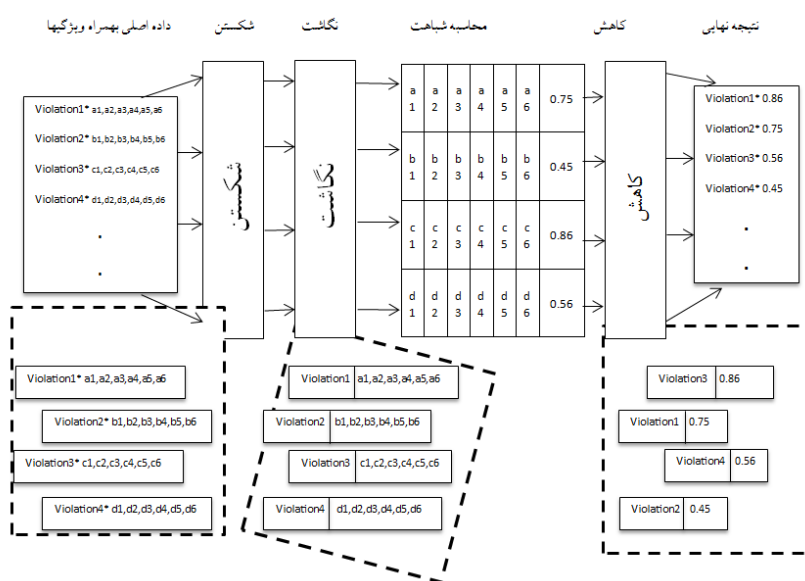
شکل ۴. شبه کد مربوط به تابع نگاشت/کاهش به منظور تشخیص مکانیزه رفتار راننده

علائم راهنمایی و رانندگی [۳۸-۴۰] را بمنظور تعیین نوع
علائم ترافیکی فراخوانی میکند . سپس می‌بایست چراغ

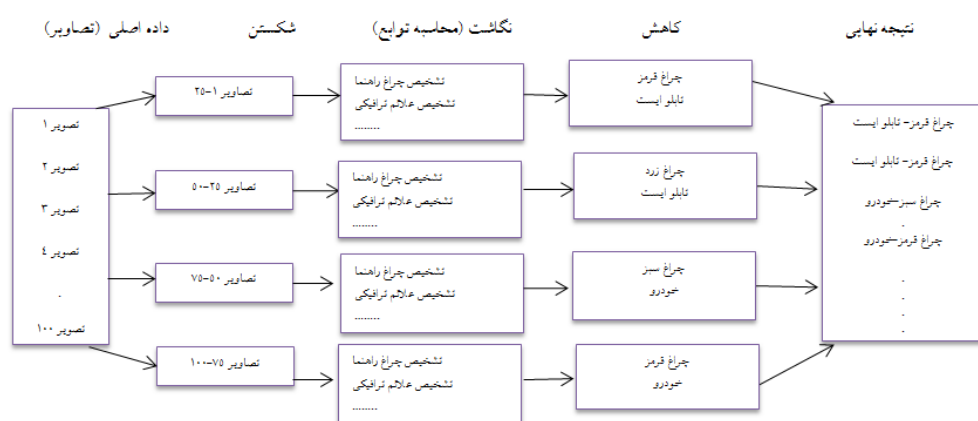
همانطور که در شکل ۴ مشاهده می‌کنید، تابع Map تصویر
یا فریمی از ویدئو را دریافت کرده و سپس تابع شناسایی

چندین شرط متوالی مقادیر خروجی هر تابع را مقایسه کرده و در نهایت نوع تخلف (تخلفات پارک- حرکتی- عبور ممنوع و غیره) و رفتار راننده را برای فریم مورد نظر تعیین کرد. در شکل ۵ فرآیند کلی نگاشت/کاهش تشخیص تخلف رانندگی با جزئیات به تصویر کشیده شده است.

راهنما و رنگ چراغ [۴۱، ۴۲] را با استفاده از تابع مذکور شناسایی کرد. در مراحل بعدی محدوده‌ی خیابان [۴۳]، خطوط عابر پیاده و وسایل نقلیه [۴۰] شناسایی می‌شوند. سپس با استفاده از چندین شرط نوشته شده در شبه کد مورد نظر می‌توان نوع تک‌تک رفتارها را شناسایی کرد. در تابع Reduce، نیز از نتایج بدست‌آمده از هر تابع موجود در تابع Map می‌توان بعنوان ورودی این تابع استفاده کرد و با



(الف)



(ب)

شکل ۵. فرآیند کلی نگاشت/کاهش تشخیص تخلف رانندگی (الف) توضیحات پلیس (ب) تصاویر مراکز کنترل ترافیک

۴. انجام تنظیمات و نتایج

مطالعه تجربی با زبان جاوا به همراه کتابخانه OpenCV پیاده‌سازی شده است. سیستم کامپیوتری تست شده در محیط VMware با دو خوشه پیاده‌سازی شده است. خوشه اول فقط یک گره دارد و خوشه دوم شامل یک گره

در این بخش کارآیی رویکرد پیشنهادی بررسی شده و نتایج حاصل از آن ارائه می‌شود. رویکرد پیشنهادی در این

تشخیص تخلف رانندگان با استفاده از نگاشت/کاهش

master و چهار گره slave است که بر روی تمامی گره های خوشه ها سیستم عامل لینوکس اوبونتو نصب و راه اندازی شده است. جدول ۱ تنظیمات نرم افزاری و سخت افزاری را نشان می دهد. به منظور بررسی و تست سیستم پیشنهادی از اطلاعات دوربین های موجود در سطح شهر استفاده شده است که از مرکز کنترل ترافیک راهنمایی و رانندگی شهرستان یزد در ایران بدست آمده است. تعداد دوربین های ترافیکی استان یزد ۱۰۰ دستگاه هست که شامل ۳۹ دوربین ثبت تخلف سرعت و ۶۱ دوربین نظارت تصویری است. برای این پروژه اطلاعات

چندین دوربین متصل در ارتفاع مشخص جمع آوری شده است که داده ها از طرف مرکز کنترل ترافیک در دسترس قرار گرفته است. برای ارزیابی عملکرد روش پیشنهادی دو مجموعه داده شامل داده های غیر ساخت یافته متنی و تحلیل داده های چند رسانه ای (شامل فایل های تصویری و فیلم ها) در نظر گرفته شده است. در فاز اول از مجموعه داده های متنی که توسط پلیس جمع آوری شده است استفاده می شود. در فاز دوم از اجرای این تحقیق از اطلاعات دوربین های نظارتی مرکز کنترل ترافیک یزد استفاده می شود (جدول ۲).

جدول ۱. مشخصات نرم افزاری و سخت افزاری سیستم

نسخه	سخت افزار /نرم افزار
Linux Ubuntu 14.04 LTS	سیستم عامل
Hadoop 2.7.1	سیستم فایل توزیع شده هادوپ (HDFS)
JDK 1.8.0_152	جاوا
OpenCV_3.2.0	کتابخانه OpenCV
VMware Workstation 12	ابزار مجازی سازی
Intel (R) Core(TM) i7-4710HQ CPU 2.50GHz	CPU
16 GB	Memory

جدول ۲. توضیحات دیتاست ها

تعداد دسته ها	ابعاد	تعداد (رکوردها)	نام دیتاست	دیتاست
-	۶	۱۰۰۰۰	داده های غیرساخت یافته (توضیحات پلیس)	۱
-	-	۱۰۰۰۰	داده های چند رسانه ای (تصاویر مراکز کنترل ترافیک)	۲

۴-۱ مقایسه کارآیی برنامه ترتیبی و خوشه هادوپ با یک گره

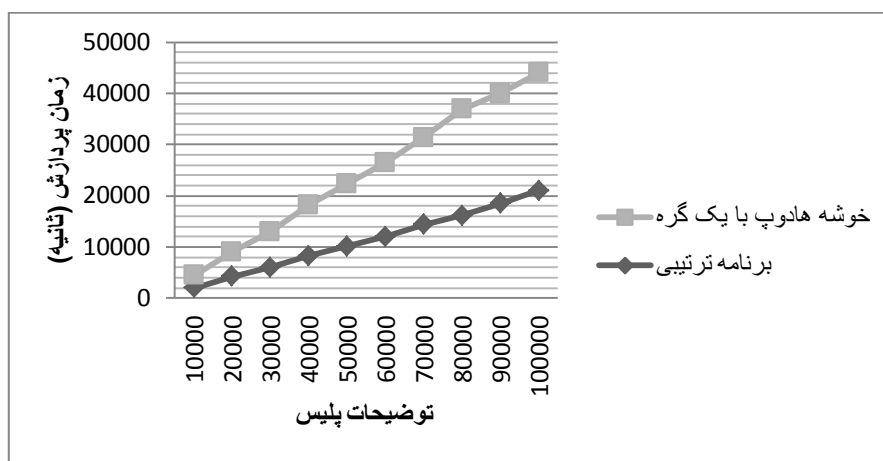
برای مقایسه عملکرد سیستم تشخیص رفتار متخلفین حادثه ساز میان برنامه ترتیبی بدون نگاشت/کاهش و

خوشه هادوپ با یک گره، ما به طور جداگانه زمان پردازش تعداد مختلفی از توضیحات پلیس و تصاویر دوربین های نظارتی را در محدوده ۱۰۰۰۰ تا ۱۰۰۰۰۰ آزمایش می کنیم. زمان پردازش از لحظه ورود توضیحات پلیس و تصاویر تا لحظه ای که نتایج سیستم تشخیص رفتار

رانندگان بدست آید، محاسبه می‌شود. نتایج کارآیی در شکل ۶ و ۷ نشان داده شده است.

در شکل ۶ خط سیاه زمان پردازش برنامه ترتیبی و خط خاکستری زمان پردازش با خوشه هادوپ با یک گره برای تحلیل تعداد متفاوتی از داده‌های غیرساختیافته (توضیحات پلیس) استفاده شده است. واضح است که خوشه هادوپ با یک گره بدلیل وجود سربرار بالا زمان بیشتری را مصرف می‌کند. علاوه بر این، اختلاف زمان

مصرف با افزایش تعداد داده‌های متنی (توضیحات پلیس) افزایش می‌یابد. برای مثال، هنگام پردازش ۱۰۰۰۰ داده متنی، زمان پردازش برنامه موردنظر با خوشه هادوپ با یک گره برابر ۲۴۰۰ است، در حالی که برای پردازش ۱۰۰۰۰۰ داده زمان پردازش به ۲۳۰۰۰ افزایش می‌یابد. بنابراین کارایی برنامه ترتیبی به مراتب بهتر از خوشه هادوپ با فقط یک گره بدلیل وجود سربرار در هنگام لود داده در خوشه هادوپ خواهد بود.

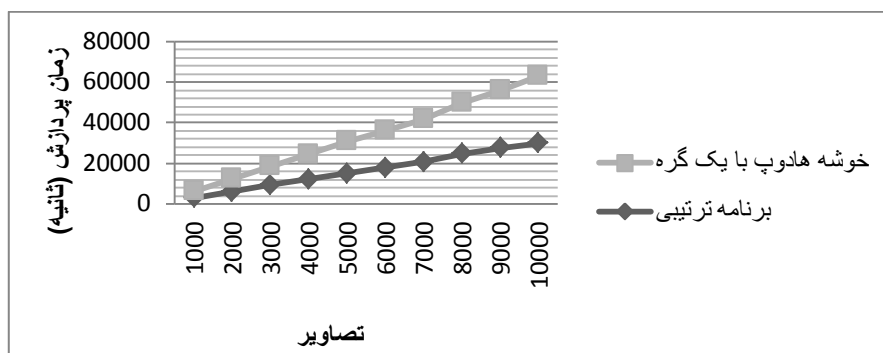


شکل ۶. زمان پردازش حاصل از اجرای الگوریتم تشخیص تخلف رانندگی برای توضیحات پلیس با در نظر گرفتن برنامه ترتیبی و کلاستر هادوپ با یک گره

پردازش ۱۰۰ تصویر، زمان پردازش با خوشه هادوپ با یک گره برابر ۳۳۰۰ است، در حالی که برای پردازش ۱۰۰۰۰ داده تصویری زمان پردازش به ۳۳۰۰۰ افزایش می‌یابد. بنابراین بهره‌وری سیستم تشخیص رفتار رانندگان برای داده‌های متنی با استفاده از خوشه هادوپ با یک گره، بدلیل سربرار قابل توجهی که وجود دارد کاهش می‌یابد.

در شکل ۷ نیز خط سیاه زمان پردازش برنامه ترتیبی و خط خاکستری زمان پردازش با خوشه هادوپ با یک گره برای تحلیل تعداد متفاوتی از داده‌های غیر ساختیافته (تصاویر دوربین‌های نظارتی) استفاده شده است. همانند تحلیل زمان پردازش برای داده‌های غیرساختیافته (متنی)، اختلاف زمان مصرف با افزایش تعداد داده‌های چندرسانه‌ای (تصاویر) افزایش می‌یابد. برای مثال، هنگام

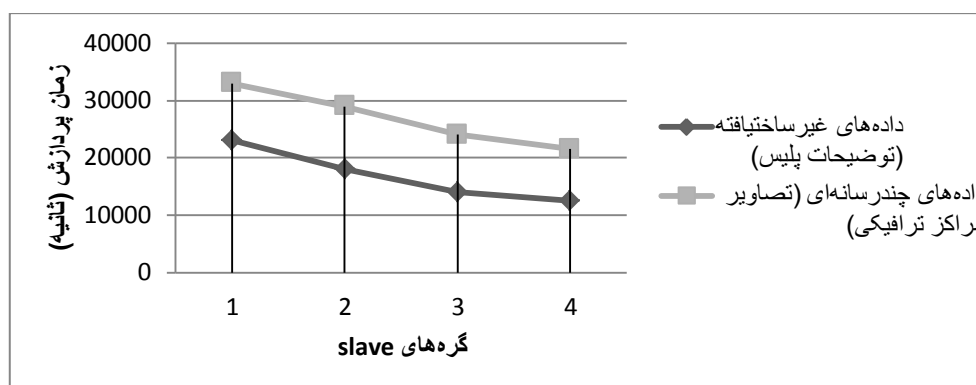
تشخیص تخلف رانندگان با استفاده از نگاشت/کاهش



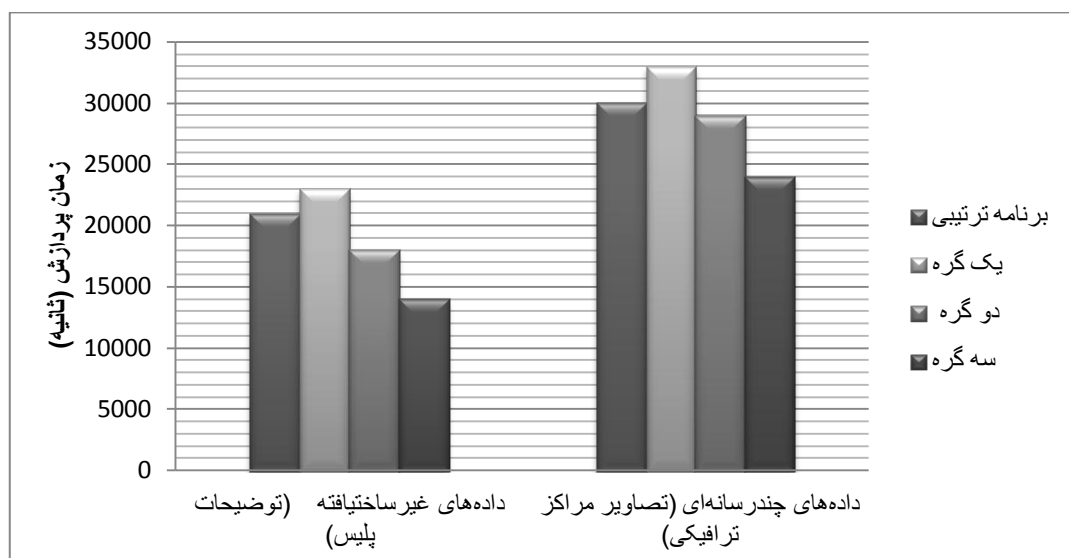
شکل ۷. زمان پردازش حاصل از اجرای الگوریتم تشخیص تخلف رانندگی برای تصاویر برگرفته از دوربین‌های نظارتی با در نظر گرفتن برنامه ترتیبی و کلاستر هادوپ با یک گره

علاوه بر این، به منظور ارزیابی بهبود عملکرد سیستم تشخیص رفتار رانندگان با افزایش گره‌های موجود در یک خوشه هادوپ، زمان پردازش ۱۰۰۰۰۰ داده غیر ساختیافته متنی و ۱۰۰۰ داده چندرسانه‌ای تصویری را با خوشه‌ای از ۱ تا ۴ گره slave در شکل ۸ مقایسه می‌کنیم. همانطور که می‌بینید زمان پردازش با افزایش گره‌های slave (گره داده) کاهش می‌یابد. جالب است بدانید که زمان پردازش عملیات تشخیص رفتار رانندگان برای داده‌های غیر ساختیافته متنی در خوشه هادوپ با ۱ گره slave ۲۳۰۰۰ ثانیه بطول می‌انجامد در حالیکه با اجرای این عملیات بر روی خوشه‌ای با ۴ گره slave حدوداً ۱۲۵۰۰ ثانیه بطول خواهد انجامید. همچنین در

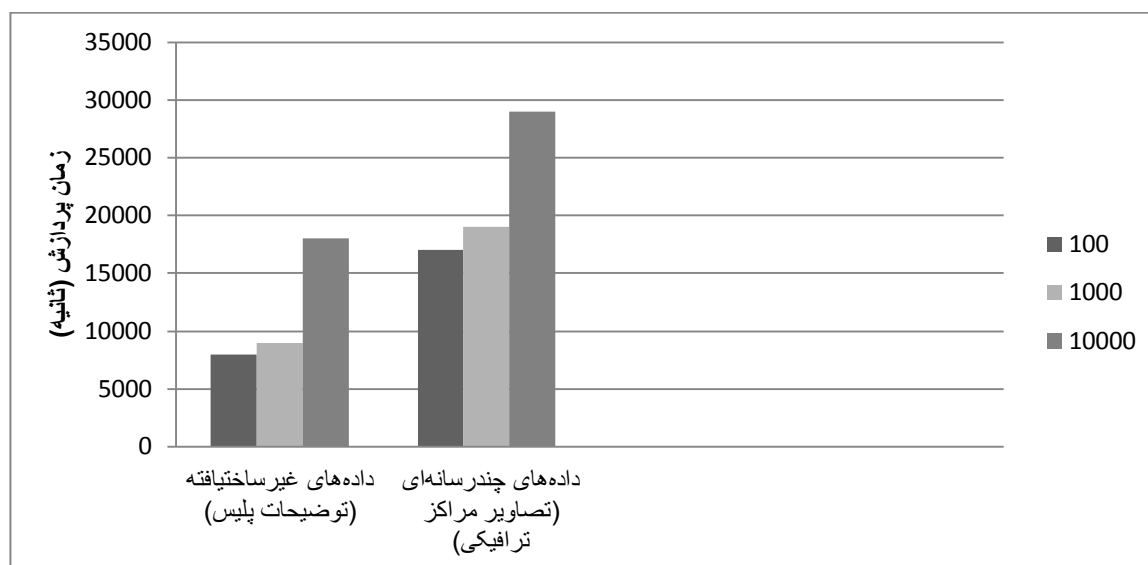
داده‌های غیر ساختیافته در خوشه هادوپ با ۱ گره slave و ۴ گره slave زمان پردازش به ترتیب از ۳۳۰۰۰ به ۲۱۵۰۰ کاهش یافته است. بنابراین داده‌ها اعم از متن و تصویر در سراسر گره‌های داده ایجاد شده توزیع می‌شوند و زمان پردازش را کاهش می‌دهند. به عبارتی دیگر، شکل ۸ نشان می‌دهد که بازدهی سیستم به صورت خطی با افزایش تعداد slave ها در خوشه هادوپ افزایش می‌یابد. بنابراین، با افزایش تعداد گره‌های Slave در تحلیل توضیحات پلیس و تصاویر مراکز کنترل به ترتیب بیش از ۷۵٪ و ۸۲٪ زمان پردازش کاهش می‌یابد.



شکل ۸. زمان پردازش حاصل از اجرای الگوریتم تشخیص تخلف رانندگی برای داده‌های غیرساختیافته متنی و چندرسانه‌ای تصویری با در نظر گرفتن گره‌های slave متفاوت



شکل ۹. زمان پردازش حاصل از اجرای الگوریتم‌های تشخیص تخلف رانندگی با در نظر گرفتن تعداد گره‌ها



شکل ۱۰. زمان پردازش حاصل از اجرای الگوریتم‌های تشخیص تخلف رانندگی با در نظر گرفتن حجم داده‌ها

۲-۴ مقایسه کارآیی و مقیاس پذیری در خوشه

هادوپ با یک گره master و چندین گره slave

به منظور برآورد کارآیی و مقیاس پذیری رویکرد پیشنهادی، دو آزمایش محاسباتی با پیاده سازی الگوریتم های تشخیص تخلف رانندگی انجام شده است. هدف ارزیابی اول، مقایسه مقیاس پذیری پیاده سازی الگوریتم های تشخیص رفتار رانندگان متخلف حادثه ساز با افزایش تعداد گره ها است. در طول این تست ها نیز، دو مجموعه دیتاست موجود در جدول ۲ با استفاده از تعدادی از گره ها پردازش شد.

با توجه به نتایج ارائه شده در شکل ۹، به استثنای خوشه ی با یک گره، سیستم دارای مقیاس پذیری نزدیک به خطی است. با این حال، در تحلیل توضیحات پلیس، با در نظر گرفتن یک گره Slave نسبت به حالتی که سه گره Slave وجود دارد، تقریباً زمان پردازش به ۶۰٫۸۶٪ کاهش می یابد. همچنین در تحلیل تصاویر دوربین های نظارتی نیز زمان پردازش به ۷۲٪ کاهش می یابد همانطور که در بخش ۴-۱ نیز بیان شد، کارآیی ضعیف خوشه با یک گره، در مقایسه با پیاده سازی ترتیبی، ناشی از سربار هادوپ است.

تست دوم با هدف برآورد میزان چگونگی مقیاس سیستم با افزایش حجم داده صورت گرفت. برای این منظور کل خوشه برای پردازش تعداد مختلفی از توضیحات پلیس در خصوص تخلف و تصاویر مراکز کنترل ترافیک مورد استفاده قرار گرفت. نتایج در شکل ۱۰ ارائه شده است. زمان پردازش برای مجموعه داده های کوچک (تعداد داده کمتر)، نیز همان است (به علت سربار خوشه ای). همانطور که انتظار میرفت زمان پردازش برای مجموعه داده های بزرگ (تعداد داده بیشتر)، خطی است. زیرا داده های تصویری حاصل از مرکز کنترل ترافیک به طور مستقل پردازش می شوند.

۵. نتیجه گیری

امروزه کمتر سیستم های حمل و نقلی وجود دارد که بخشی از آن توسط برنامه های هوشمند بینایی ماشین کنترل نشود. خطای بسیار کم، سرعت زیاد، هزینه نگهداری بسیار پایین و خیلی مزایای دیگر باعث شده که صنایع مختلف به سرعت به سمت پردازش تصویر و بینایی ماشین روی بیاورند. سیستم های نظارت تصویری از ابزار اصلی مدیریت سیستم های حمل و نقل بوده و مزیت آن ها فراهم کردن اطلاعات تصویری برای تصمیم گیری است. در این مقاله، کنترل مکانیزه با استفاده از الگوریتمی مبتنی بر نگاهت/کاهش جایگزین نیروی انسانی می گردد؛ بنابراین، اولاً اشتباهات ناشی از خطاهای انسانی از قبیل خطای دید، خستگی و ... از سیستم حذف می گردد؛ ثانیاً حجم نیروی انسانی آموزش دیده مورد نیاز کاهش می یابد و به تبع آن هزینه های وابسته نیز کم می شود. همچنین کارآیی سیستم با افزایش تعداد گره های داده بیش از ۷۵ درصد افزایش پیدا می کند. با تمرکز بر روی پیامد و نتایج ناگوار ناشی از حوادث ترافیکی به نحوی که در جهت تغییر عادت های نادرست رانندگی پیش رویم و هدف از آن نشان دادن تخلفات و سرپیچی از قوانین و مقررات باشد، می تواند بسیاری از رفتارهای غیراجتماعی رانندگان را کنترل و از تصادفات و تلفات شدید جلوگیری کند.

مراجع

- [1] Rahemi, Z., et al., *Sensation-seeking and factors related to dangerous driving behaviors among Iranian drivers*. Personality and Individual Differences, 2017. **116**: p. 314-318.
- [2] Rakotonirainy, A., R. Schroeter, and A. Soro, *Three social car visions to improve driver behaviour*. Pervasive and Mobile Computing, 2014. **14**: p. 147-160.
- [3] Tao, D., R. Zhang, and X. Qu, *The role of personality traits and driving experience in self-reported risky driving behaviors and accident risk among Chinese drivers*. Accident Analysis & Prevention, 2017. **99**: p. 228-235.

- data clustering. Pattern Recognition Letters, 2017. **93**: p. 78-84.
- [16] Valcarce, D., J. Parapar, and Á. Barreiro, *A MapReduce implementation of posterior probability clustering and relevance models for recommendation*. Engineering Applications of Artificial Intelligence, 2018. **75**: p. 114-124.
- [17] Chen, C.-H., *Improved TFIDF in big news retrieval: An empirical study*. Pattern Recognition Letters, 2017. **93**: p. 113-122.
- [18] Abrahams, A.S., et al., *An integrated text analytic framework for product defect discovery*. Production and Operations Management, 2015. **24**(6): p. 975-990.
- [19] Alham, N.K., et al., *A MapReduce-based distributed SVM algorithm for automatic image annotation*. Computers & Mathematics with Applications, 2011. **62**(7): p. 2801-2811.
- [20] Afrati, F.N., et al., *Computing marginals using MapReduce*. Journal of Computer and System Sciences, 2018. **94**: p. 98-117.
- [21] Cattaneo, G., et al., *MapReduce in Computational Biology Via Hadoop and Spark*. Encyclopedia of Bioinformatics and Computational Biology, 2016: p. 1-9.
- [22] Kouanou, A.T., et al., *An optimal big data workflow for biomedical image analysis*. Informatics in Medicine Unlocked, 2018.
- [23] Osman, A.M.S., *A novel big data analytics framework for smart cities*. Future Generation Computer Systems, 2018.
- [24] Cantabella, M., et al., *Analysis of student behavior in learning management systems through a Big Data framework*. Future Generation Computer Systems, 2019. **90**: p. 262-272.
- [25] Rios, L.G. *Big data infrastructure for analyzing data generated by wireless sensor networks*. in *Big Data (BigData Congress), 2014 IEEE International Congress on*. 2014. IEEE.
- [26] Yin, L., et al., *'Big data' for pedestrian volume: Exploring the use of Google Street View images for pedestrian*
- [4] Lotfi, E., *Trajectory Clustering and Behaviour Retrieval from Traffic Surveillance Videos*. Majlesi Journal of Multimedia Processing, 2011. **1**(2).
- [5] Khabiri, M., *Assessing behavioral patterns of motorcyclists based on traffic control device at city intersections by classification tree algorithm*. International Journal of Transportation Engineering, 2018. **5**(4): p. 501-415.
- [6] Moghaddam, A.M. and E. Ayati, *Introducing a risk estimation index for drivers: A case of Iran*. Safety science, 2014. **62**: p. 90-97.
- [7] Park, S.H., et al. *Vision-based traffic surveillance system on the internet*. in *Computational Intelligence and Multimedia Applications, 1999. ICCIMA'99. Proceedings. Third International Conference on*. 1999. IEEE.
- [8] McLauchlan, P., et al. *A real-time computer vision system for measuring traffic parameters*. in *cvpr*. 1997. IEEE.
- [9] Aoyama, K. *Next Generation Universal Traffic Management System (UTMS'21) in Japan*. in *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*. 1997. IEEE.
- [10] Kitchin, R., *The real-time city? Big data and smart urbanism*. GeoJournal, 2014. **79**(1): p. 1-14.
- [11] Chen, M., S. Mao, and Y. Liu, *Big data: A survey*. Mobile networks and applications, 2014. **19**(2): p. 171-209.
- [12] Wang, W., et al., *An efficient image aesthetic analysis system using Hadoop*. Signal Processing: Image Communication, 2015. **39**: p. 499-508.
- [13] Bendre, M. and R. Manthalkar, *Time series decomposition and predictive analytics using MapReduce framework*. Expert Systems with Applications, 2019. **116**: p. 108-120.
- [14] Manogaran, G., D. Lopez, and N. Chilamkurti, *In-Mapper combiner based MapReduce algorithm for processing of big climate data*. Future Generation Computer Systems, 2018. **86**: p. 433-445.
- [15] Banharnsakun, A., *A MapReduce-based artificial bee colony for large-scale*

- identification by deformable models for intelligent vehicles*. IEEE transactions on intelligent transportation systems, 2004. **5**(2): p. 57-68.
- [40] Kasaei, S.H.M. and S.M.M. Kasaei. *Extraction and recognition of the vehicle license plate for passing under outside environment*. in *Intelligence and Security Informatics Conference (EISIC), 2011 European*. 2011. IEEE.
- [41] Sallah, M., et al., *Road sign detection and recognition system for real-time embedded applications*. 2011.
- [42] Elotmani, S. and M. El Hitmy. *A light traffic signs recognition system*. in *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*. 2014. IEEE.
- [43] Krishnan, A., C. Lewis, and D. Day. *Vision system for identifying road signs using triangulation and bundle adjustment*. in *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*. 2009. IEEE.
- counts. *Applied Geography*, 2015. **63**: p. 337-345.
- [27] Jeon, S. and B. Hong, *Monte Carlo simulation-based traffic speed forecasting using historical big data*. *Future generation computer systems*, 2016. **65**: p. 182-195.
- [28] De Mauro, A., et al., *Human resources for Big Data professions: A systematic classification of job roles and required skill sets*. *Information Processing & Management*, 2018. **54**(5): p. 807-817.
- [29] Secundo, G., et al., *Intellectual capital in the age of Big Data: establishing a research agenda*. *Journal of Intellectual Capital*, 2017. **18**(2): p. 242-261.
- [30] Gantz, J. and D. Reinsel, *Extracting value from chaos*. IDC iVIEW, 2011. **1142**(2011): p. 1-12.
- [31] Le, T.M. and S.-Y. Liaw, *Effects of Pros and Cons of Applying Big Data Analytics to Consumers' Responses in an E-Commerce Context*. *Sustainability*, 2017. **9**(5): p. 798.
- [32] Laney, D., *3D data management: Controlling data volume, velocity and variety*. META group research note, 2001. **6**(70): p. 1.
- [33] Chen, M., et al., *Big data: related technologies, challenges and future prospects*. 2014.
- [34] Shvachko, K., et al. *The hadoop distributed file system*. in *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. 2010. Ieee.
- [35] Dean, J. and S. Ghemawat, *MapReduce: simplified data processing on large clusters*. *Communications of the ACM*, 2008. **51**(1): p. 107-113.
- [36] White, T., *Hadoop: The definitive guide*. 2012: "O'Reilly Media, Inc."
- [37] Zhang, W., T. Yoshida, and X. Tang, *A comparative study of TF* IDF, LSI and multi-words for text classification*. *Expert Systems with Applications*, 2011. **38**(3): p. 2758-2765.
- [38] Bui-Minh, T., et al. *Two algorithms for detection of mutually occluding traffic signs*. in *Control, Automation and Information Sciences (ICCAIS), 2012 International Conference on*. 2012. IEEE.
- [39] De La Escalera, A., et al., *Visual sign information extraction and*