



## مدیریت سیستم کنترل ترافیک با استفاده از

### الگوریتم های ساده و سریع و با کمک پردازش تصویر

نجمه طباطبایی  
دانشجوی مهندسی برق - الکترونیک  
گروه برق و کامپیوتر  
دانشکده مهندسی شهید نیکبخت  
[n.tahoa@gmail.com](mailto:n.tahoa@gmail.com)

مهدی ماهرانی  
دانشجوی مهندسی برق - الکترونیک  
گروه برق و کامپیوتر  
دانشکده مهندسی شهید نیکبخت  
[mahdi.maherani@gmail.com](mailto:mahdi.maherani@gmail.com)

معصومه خلیل زاده  
دانشجوی مهندسی برق - الکترونیک  
گروه برق و کامپیوتر  
دانشکده مهندسی شهید نیکبخت  
[freedom.prs@gmail.com](mailto:freedom.prs@gmail.com)

سرور صارمی فر  
دانشجوی مهندسی برق - الکترونیک  
گروه برق و کامپیوتر  
دانشکده مهندسی شهید نیکبخت  
[plz.no.war@gmail.com](mailto:plz.no.war@gmail.com)

**چکیده-** یکی از روشهای شناخت رفتار رانندگان در بزرگراه ها بررسی موقعیت تک تک وسایل نقلیه در طول مسیر است. در این مقاله، الگوریتم های پردازش تصویر برای تعیین پارامترهای مختلف ترافیکی پیشنهاد شده است و اساس این کار بر مبنای روش پنجره است که در آن تنها بخش هایی از تصویر پردازش می شود بدین ترتیب، سرعت پردازش بالا رفته و کارایی بیشتر می شود و برای رسیدن به یک سیستم بلادرنگ سعی شده از الگوریتم های ساده و سریع استفاده شود که توسط یک بورد تصاویر ترافیکی گرفته شده آنها را با تصاویر دیجیتال تبدیل کرده و با استفاده از الگوریتم تصمیم گیری، تصمیم نهایی در مورد پارامتر مد نظر گرفته می شود.

**کلمات کلیدی:** الگوریتم، پردازش تصویر، پنجره، تصاویر ترافیکی، سرعت

۱. تشخیص و شمارش خودرو
۲. اندازه گیری سرعت خودرو
۳. تشخیص و شمارش خودرو در شب
۴. اندازه گیری طول صف در پشت چراغ قرمز
۵. نرخ توزیع خودرو در چهارراه ها

پس از اینکه مقدمات کار آماده شده و سیستم اصلی راه اندازی شد، به بررسی در مورد الگوریتم های کلی برای هر یک از این پارامترها به طور جداگانه می پردازیم. برای انتخاب بهترین الگوریتم برای تشخیص می پردازیم.

#### سیستم موجود در ایران :

از سال ۱۹۷۰ پس از برنامه امریکایی WADS پروژه ها و آزمایش هایی در مورد سیستم های ترافیکی انجام شده از جمله این پروژه ها می توان به سیستم های زیر اشاره کرد ؛ سیستم هایی که در سال ۱۹۸۴ بررسی شدند UMIST در انگلیس که از تفاضل تصویر و به دنبال آن یک حد آستانه دستی استفاده می کند و سیستم ژاپنی TSC60 که تحت شرایط جوی مختلف خودرو ها را تشخیص داده ولی به خودرو های نزدیک حساس است. همچنین سیستم استرالیایی VDO نیز که تصویر را به ۱۶ پنجره تحلیلی تقسیم می کند و از یک پنجره دیگر به عنوان مرجع تفاضل استفاده می کند همچنین در سال ۱۹۹۵ روش فرانسوی TITAN که مربوط

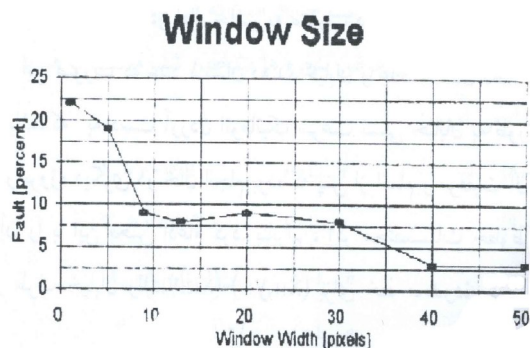
**مقدمه -** برای پردازش تصویر بسته به اطلاعاتی که می خواهیم از تصویر استخراج کنیم روشهای مختلفی وجود دارد. دو روش اصلی برای پردازش، پردازش نقطه ای و پردازش همسایگی است. که در روش نقطه ای برای کار با داده های تصویر مقدار هر پیکسل به تنهایی و بدون مراجعه به پیکسل های جاور در نظر گرفته می شود هرچهار عمل اصلی (تفریق، جمع، ضرب و تقسیم) را می توان بر روی داده های تصویر دیجیتال به صورت پیکسل پیکسل اعمال می کنیم. تفریق دو تصویر یک تبدیل نقطه ای بسیار ساده و موثر را فراهم میکند. در موارد ترافیکی از این روش که تفاضل زمینه نامیده می شود برای تشخیص خودرو در تصویر استفاده می شود. همچنین اگر عمل جمع با تقسیم ترکیب شود تکنیکی برای بر طرف کردن نویز اتفاقی بدست می آید. در روش پردازش همسایگی اختلاف در پیکسل های مجاور مهم است زیرا یک اختلاف کوچک نشاندهنده ناپوستگی در دنیای واقعی و احتمالاً مربوط به نویز است و می توان با در نظر گرفتن مقدار میانگین یک گروه پیکسل نویز را کم کرد و همچنین این روش برای عملیات لبه گیری و هموار سازی به کار می رود.

هدف از ارائه این مقاله، اندازه گیری پارامتر سرعت و چند پارامتر کمی ترافیکی است. این پارامترها عبارتند از :

است به تحلیل ترافیک شهری با استفاده از ریاضیات مورفولوژی و روش زمانی- مکانی برای تحلیل تصویر. مورد آزمایش قرار گرفت

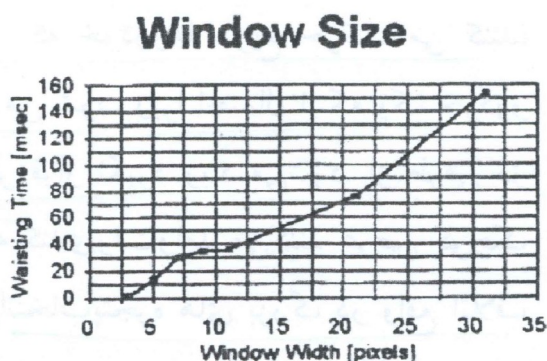
در ایران نیز در سال ۱۳۷۲ در شرکت کنترل ترافیک تهران سیستمی تحت یک پروژه طراحی شد. در این سیستم از یک میکروپروسسور ۸۰۳۸۶ به همراه یک بورد DT2867 از شرکت Data Translation به عنوان بورد Frame Grabber استفاده شده است.

شکل ۲- منحنی خطا بر حسب اندازه پنجره در یک خیابان نمونه



شکل ۲

شکل ۳- منحنی زمان محاسبات بر حسب اندازه پنجره



شکل ۳

## ۱-۲- تعیین آستانه باینری کردن

مقدار کوچکتر این آستانه به سیستم حساستر منجر می شود و نقاط بیشتری ( که ممکن است ناشی از نویز و ....) باشد به عنوان لبه حساب می شود و اگر از حدی بزرگتر شود امکان دارد لبه های خودرو را تشخیص ندهد. البته باید بدانیم که مقدار این عدد برای هر خیابان متفاوت است .

## بیان الگوریتم های اندازه گیری پارامترهای

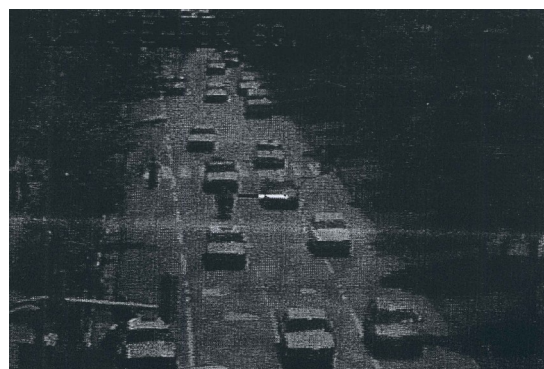
### کمی ترافیک :

#### ۱- شمارش خودرو

##### ۱-۱- تعیین اندازه پنجره

اولین قدم برای این مرحله ، تعیین محل و اندازه مناسب پنجره پردازشی است . پنجره ای افقی که طول آن برابر عرض خط مورد نظر در خیابان باشد. شکل ۱ ، حالت کلی این پنجره را برای یک خط خیابان نشان می دهد. هرچه عرض پنجره بیشتر شود ، دقت بالا می رود البته تاجایی که از حد طول یک خودرو بالاتر نرود.

شکل ۱ - پنجره تشخیص خودرو برای یک خط خیابان



شکل ۱

از طرف دیگر این عرض بستگی به سرعت متوسط خودرو ها در هر خیابانی نیز دارد این عدد به صورت دستی به کامپیوتر وارد می شود . سرعت زیادتر، کارایی پنجره با عرض کوچک را کاهش می دهد ، زیرا احتمال اینکه یک خودرو در هیچ فریمی در پنجره مورد نظر قرار نگیرد زیاد می شود . در تصاویری که با بزرگ نمایی بالا گرفته شده اند اندازه پنجره باید بزرگ تر از تصاویر با بزرگ نمایی پایین باشد. شکل های ۲ و ۳ منحنی های خطا و زمان تلف شده را با اندازه های مختلف پنجره نشان می دهند.

ایجاد لبه های اندکی در پنجره شده باشد ، خودرو در پنجره تشخیص داده می شود. آستانه بزرگ هم ممکن است باعث نادیده گرفتن حضور خودرو شود.

#### ۴-۱- الگوریتم کلی تشخیص خودرو

پس از اینکه محدوده دو طرف خیابان و تعداد خطوطی را که در این محدوده قرار دارند و نیز سرعت خیابان را برای سیستم مشخص کردیم ، اندازه و محل قرار گرفتن پنجره ها محاسبه می شوند. پس از آن سیستم شروع به گرفتن تصویر می کند و آنقدر تصویر می گیرد تا یک تصویر را کاربر به عنوان تصویر زمینه انتخاب کند. بر روی این تصویر ابتدا عملیات حذف نویز را انجام می دهیم. فیلتری که برا این کار در نظر گرفته شده است ، فیلتر میانگین می باشد. پس از این کار ، عمل لبه گیری را انجام می دهیم. برای این کار فیلتر سبل را اجرا می کنیم. حالا این تصویر را برای استفاده در محل بعد در یک بافر ذخیره می کنیم. برای انتخاب آستانه باینری ، یک تصویر را گرفته ، بر روی آن همان عملیات حذف نویز و لبه گیری را انجام می دهیم.

دیده شده که عمل تفاضل ، خود موجب بروز نویزهایی در تصاویر می شود. این نویزها ، به صورت لکه های روشن ، بر روی زمینه تیره دیده می شوند. برای از بین بردن لکه ها تصمیم گرفتیم که از یک فیلتر مورفولوژی باز کردن استفاده کنیم. پس از این کار با استفاده از آستانه به دست آمده ، تصویر را باینری می کنیم. برای این کار ، کلیه نقاط کوچکتر از آستانه را صفر و کلیه نقاط بزرگتر از آنرا بیشترین عدد یعنی ۲۵۵ قرار می دهیم. اکنون می توانیم کلیه نقاط سفید تصویر را شمرده و درصد آن را نسبت به کل نقاط پنجره به دست آوریم. برای این کار ، باز هم از تابع هیستوگرام استفاده می کنیم. اگر این نسبت از ۱۵٪ بیشتر بود ، خودرو در آن پنجره تشخیص داده می شود.

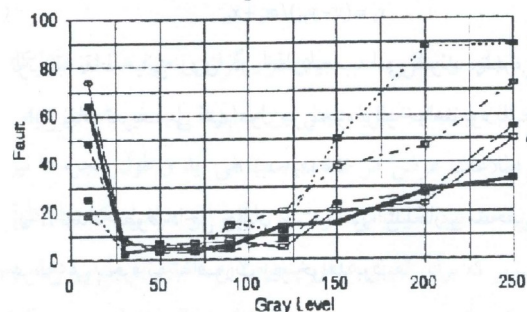
پس از اینکه بر روی تمامی تصاویر مورد نظر این عملیات را انجام دادیم ، برداری به صورت صفر و یک برای هر دنباله از تصاویر به دست می آوریم که برای تصمیم گیری های بعدی به قسمت تصمیم گیری فرستاده می شود.

#### ۵-۱- تصمیم گیری برای تعداد خودرو

حال با استفاده از این بردار ، تعداد خودرو های عبوری از پنجره را تخمین بزنیم. این مسئله ، کاملاً به سرعت متوسط خودروها در هر خیابان و همچنین طول خودروهای عبوری بستگی خواهد داشت گاه اتفاق می افتد که یک خودرو درست در محل پنجره

شکل ۴- نمودار خطا برحسب آستانه های باینری مختلف. برای

#### Binary Threshold



شکل ۴

بنابراین باید الگوریتم ، خود قادر باشد با توجه به تصویر موجود ، این آستانه را استخراج کند.

#### ۳-۱- تعیین آستانه برای نقاط سفید

آستانه بعدی ، تعداد نقاط سفیدی است که پس از باینری شدن ، معرف یک خودرو می باشند. اگر این عدد کوچک انتخاب شود در حالیکه عبور خودرو در خطوط مجاور ، باعث پس از اینکه محدوده دو طرف خیابان و تعداد خطوطی را که در این محدوده قرار دارند و نیز سرعت خیابان را برای سیستم مشخص کردیم ، اندازه و محل قرار گرفتن پنجره محاسبه می شوند. پس از آن سیستم شروع به گرفتن تصویر می کند و آنقدر تصویر می گیرد تا یک تصویر را کاربر به عنوان تصویر زمینه انتخاب کند.

بر روی این تصویر ابتدا عملیات حذف نویز را انجام می دهیم. فیلتری که برا این کار در نظر گرفته شده است ، فیلتر میانگین می باشد. پس از این کار ، عمل لبه گیری را انجام می دهیم. برای این کار فیلتر سبل را اجرا می کنیم. حالا این تصویر را برای استفاده در محل بعد در یک بافر ذخیره می کنیم.

برای انتخاب آستانه باینری ، یک تصویر را گرفته ، بر روی آن همان عملیات حذف نویز و لبه گیری را انجام می دهیم.

سپس تصویر زمینه را از آن کم می کنیم تا لبه های مربوط به زمینه از بین بروند. پس از آن ، از نتیجه یک هیستوگرام می گیریم. بر روی هیستوگرام از نقطه صفر شروع کرده ، پیش می رویم تا به اولین قله محلی برسیم. سپس باز هم کار را ادامه می دهیم تا به اولین دره بعد از آن برسیم. این نقطه به عنوان مرزی برای باینری کردن تصاویر انتخاب می شود. تصاویر ترافیکی را پشت سر هم گرفته ، بر این تصویر کم می کنیم تا لبه های زمینه حذف شوند. البته بعد از انجام چندین آزمایش

### ۳-۲- تصمیم گیری در مورد سرعت خودرو

برای محاسبه سرعت یک خودرو نیاز به دانستن فاصله بین دو پنجره زمان پیمودن آن فاصله توسط خودرو داریم. در مورد زمان پیمودن، مهمترین مسئله، تشخیص این است که خودروی حاضر در پنجره اول، پس از چند فریم به پنجره دوم می رسد. در روش معمولی، یک مقدار حداقل و حداکثر را برای این زمان در نظر می گیریم که برای هر خیابان متفاوت است. به این ترتیب که بر فرض، هیچ خودرویی نمی تواند زودتر از دو یا سه فریم به پنجره دوم برسد و اگر این زمان از ۵ یا ۶ فریم تجاوز کرد، احتمالاً خودرو در این فاصله تغییر مسیر داده است. در ضمن اگر خودرویی، با سرعت آهسته حرکت کرده و در این فاصله یک خودروی دیگر، در مدتی قابل قبول، به پنجره دوم برسد، به عنوان خودروی قبلی در نظر گرفته می شود. پس می بینیم که با این روش، خطا بالا خواهد بود. دقت متوسطی که با استفاده از این روش بدست می آید در حدود ۷۶٪ بود.

### ۳- روشهای تشخیص خودرو در شب

در شب به دلیل نور کم محیط، دقت تشخیص خودرو با لبه گیری بسیار کم است. خودروها تقریباً هم رنگ جاده دیده شده و لبه های کوچک از خود برجای می گذارند. چراغ ها جلوی خودرو لبه های مشخصی ایجاد می کنند ولی باعث بروز خطا می شود. و مشکل دیگر این است که تعدادی از خودروها چراغ ندارند یا کم نورند یا یک چراغ دارند. با توجه به مشکلات فوق، دو روش را برای تشخیص خودرو در شب آزمایش شده است که بقرار زیر است:

۱- لبه گیری

۲- باز شناسی الگو

### ۳-۱- تشخیص خودرو با استفاده از لبه گیری

این روش مانند روش تشخیص خودرو و در روز است ولی آستانه باینری کردن تصویر در شب متفاوت است. در شب بسیاری از لبه ها مربوط به انعکاس نور چراغ ها بر سطح خیابان می باشند. لبه ایجاد شده از این انعکاس از لبه ی مربوط به خود خودرو بزرگتر است و خط ها ی سفید خیابان با برخورد کمتر نوری، انعکاس قابل ملاحظه ای می دهد که با روش تفاضل زمینه قادر به حذف آن نخواهیم بود

تصمیم به تغییر خط می گیرد و بردار را از حالت قبلی خود خارج می سازد. همینطور هنگامی که یک اتوبوس و یا کامیون از پنجره عبور می کنند، تعداد یک های بردار بیشتر خواهد بود در نتیجه الگوریتمی که در اینجا پیاده کردیم به این صورت است که بر روی بردار به دست آمده پیش می رویم. حضور یک یا دو « ۱ » در این بردار نمایانگر یک خودرو فرض می شود. پس از آن، هر بار که تعداد یک های متوالی بردار به عدد فردی رسید، به خودروهای خط مورد نظر یک واحد اضافه می کنیم.

### ۲- سرعت خودرو

اندازه گیری سرعت خودروها را با دو پنجره افقی که در یک خط قرار دارند انجام می دهیم و بر روی هر یک از پنجره ها همان عملیات تشخیص خودرو را انجام می دهیم.

### ۲-۱- اندازه و محل پنجره ها

اندازه این دو پنجره الزاماً یکسان نیست. فاصله بین دو پنجره نیز با توجه به بزرگی تصویر و ضریب سرعت خیابان انتخاب می شود. هر چه تصویر بزرگتر بشد، فاصله دو پنجره هم بیشتر است و برای خیابانی با ضریب سرعت ۲، این فاصله باز هم بیشتر می شود. سعی می کنیم که با توجه به بزرگی تصویر و ضریب سرعت خیابان، فاصله دو پنجره را به گونه ای انتخاب کنیم که یک خودرو برای رسیدن از پنجره اول به پنجره دوم، حداقل دو تا سه فریم زمان لازم داشته باشد. بدین ترتیب، دقت سرعت به دست آمده، بالاتر می رود.

### ۲-۲- الگوریتم کلی اندازه گیری سرعت

ابتدا کاربر محدوده دو طرف خط حورد نظر و ضریب سرعت خیابان را برای سیستم مشخص می کند. پس از محاسبه اندازه و فاصله دو پنجره و مختصات دقیق آنها، تصویر زمینه را مانند بخش قبل ذخیره می کنیم با این تفاوت که اینبار در هیچ یک از دو پنجره نباید خودرویی وجود داشته باشد. سپس آستانه باینری استخراج کرده و عملیات حذف نویز، لبه گیری، تفاضل، باینری کردن تصویر و تشکیل بردار حالت را همانند بخش شمارش ولی این بار با دو پنجره ادامه می دهیم. دو بار حالت به دست آمده را به قسمت تصمیم گیری می فرستیم.

تعداد نقاط از ۷۵٪ بالاتر باشد خودروها متوقف شده اند. باید هر چه که به سمت بالای تصویر پیش می رویم این عدد بزرگتر شود تا جایی که در انتهای تصویر به ۹۵٪ برسد. بنابراین برای به دست آوردن این آستانه از یک تابع خطی بر حسب موقعیت نقطه بر روی خط استفاده می کنیم. اشکال دیگر این بود که با یک خط خطا بالا می رفت. زیرا در بسیاری از مواقع این خط در محلی بین دو خودرو قرار می گیرد. به همین دلیل کار را با افزایش عرض پنجره عمودی دنبال کردیم. به این ترتیب که به جای مقدرا سطح خاکستری یک نقطه، میانگین چند سطح خاکستری را در بردار پروفایل بریزیم

#### ۲-۴- تشخیص وجود خودرو

پس از اینکه عدم حرکت را پشت چراغ قرمز تشخیص دادیم، برای اطمینان از اینکه عدم حرکت به دلیل تشکیل شدن صف است یا عدم وجود خودرو، باید یک الگوریتم تشخیص وجود خودرو را نیز در این مرحله انجام دهیم. برای این کار پنجره هایی در مسیری که پروفایل را گرفته ایم انتخاب کرده همان مرحله ای که برای تشخیص خودرو داشتیم انجام می دهیم. عرض پنجره ها همان است که برای تشخیص عدم حرکت داشتیم. برای طول پنجره نیز باید هر چه به طرف انتهای خیابان پیش می رویم، طول های کوچکتری انتخاب کنیم. نسبت کوچک شدن این طول بستگی به نسبت کوچک شدن اشیا در انتهای تصویر دارد.

#### ۳-۴- محاسبه طول

تاکنون مراحل اصلی تشخیص صف انجام شده است. تنها کار باقی مانده تبدیل این طول صف به ابعاد واقعی است که بر اساس عرض خطوط خیابان می توانیم به راحتی این طول را به دست آوریم. بدین صورت که خیابان را به پنجره هایی به اندازه طول یک خودرو تقسیم می کنیم و به ازای هر پنجره که صف در آن تشخیص داده شده است، طول متوسط یک خودرو، یعنی در حدود ۴/۵ متر را به طول صف اضافه می کنیم. شکل های ۵ و ۶ نمونه هایی از صف های تشخیص داده شده را نشان می دهند.

در این روش بهترین راه حل این است که لبه های بزرگتر را حذف کنیم. حال بهترین آستانه را با روش هیستوگرام می یابیم درصد خطا در شب با استفاده از لبه گیری زیاد است.

#### ۲-۳- تشخیص خودرو با استفاده از بازشناسی الگو

اساس کار در این روش به این صورت است که پس از عملیات حذف نویز، ابتدا تمام اشیای روشن در صفحه را تفکیک کرده و با ویژگی هایی چراغ ها را از سایر نقاط روشن تمیز می دهیم. این ویژگی ها به دو بخش ویژگی های خود شئی و ارتباط آن با سایر اشیا داخل صحنه تقسیم می شود و بر اساس یه گراف رابطه تقسیم می شود.

#### ۴- تشخیص صف

برای اندازه گیری طول صف، ابتدا باید عدم حرکت را در قسمت های جلوی چهارراه تشخیص دهیم. اگر حرکتی مشاهده نشد، تا جایی که حرکت نداریم پیش می رویم. این نقطه، جایبست که انتظار می رود صف تا آنجا کشیده شده باشد. برای آنکه قسمتهای خالی خودرو را به اشتباه به عنوان صف در نظر نگیریم یک الگوریتم تشخیص خودرو هم به این قسمت اضافه می کنیم سپس این طول را بر حسب ابعاد واقعی به دست می آوریم.

#### ۱-۴- تشخیص عدم حرکت

اساس کار بر مبنای تفاضل تصاویر پیاپی است. به این صورت که ابتدا یک خط در وسط محدوده خیابان انتخاب می کنیم. سپس مقادیر این خط را در دو تصویر گرفته شده توسط سیستم، گرفته و آنها را در بردارهای پروفایل ذخیره می کنیم. پس از اینکه دو بردار پروفایل را محاسبه کردیم مقادیر آن دو را از هم کم می کنیم و نتیجه را در بردار دیگری می ریزیم. با آزمایش های مختلف نتیجه گرفته شده که زمانی که حرکت وجود ندارد مقدر این تفاضل از ۲۰ کوچکتر است. پس این عدد به عنوان آستانه برای توقف انتخاب می شود. سپس از ابتدای بردار تفاضل شروع می کنیم و درصد نقاط زیر آستانه را به کل نقاط می سنجیم و تا جایی که مقدار آن از آستانه کمتر است پیش می رویم. سپس درصد نقاط زیر آستانه را به کل نقاط می سنجیم تا جایی که این درصد از یک آستانه دیگر بزرگتر بود، توقف را تشخیص می دهیم. برای این آستانه نیز آزمایش های متعددی انجام شد بود. نتیجه این بود که در نقاط جلوی چهارراه، اگر



گردش به چپ یا راست پنجره های تعریف شده عمود بر پنجره های اصلی می باشند.

شکل ۷- پنجره های اندازه گیری نرخ توزیع خودرو در چهارراه با دو مسیر مجاز



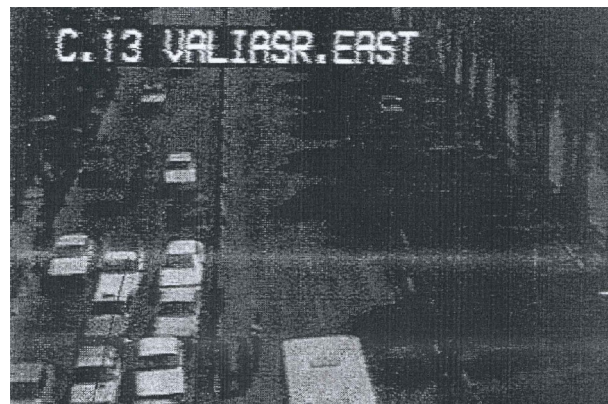
شکل ۷

حال عمل تشخیص خودرو را برای هر پنجره انجام داده و از روی آن می توانیم این نرخ را محاسبه کنیم. نکته مهم در اینجا این است که برای یک خیابان بخصوص ، سیستم باید تنها زمانی فعال شود که چراغ آن خیابان سبز است. در غیر این صورت ، خودروهای عبوری از مسیرهای دیگر را نیز به عنوان مسیر مورد نظر ما می گیرد. روشی که برای مشخص کردن زمان سبز شدن چراغ استفاده کردیم ، روش تشخیص توقف است به این صورت که پنجره ای افقی با عرض ۳ پیکسل را در جایی که خودروها در زمان قرمز بودن چراغ توقف می کنند انتخاب می کنیم. اگر در دو تصویر پیاپی مقدار این پنجره ها را از هم کم کنیم ، باید درصد پیکسل های کوچکتر از حد آستانه نسبت به کل پیکسل ها از یک مقدار مشخص بیشتر باشد. تا زمانی که این شرط برقرار است ، چراغ مربوطه را قرمز در نظر می گیریم.

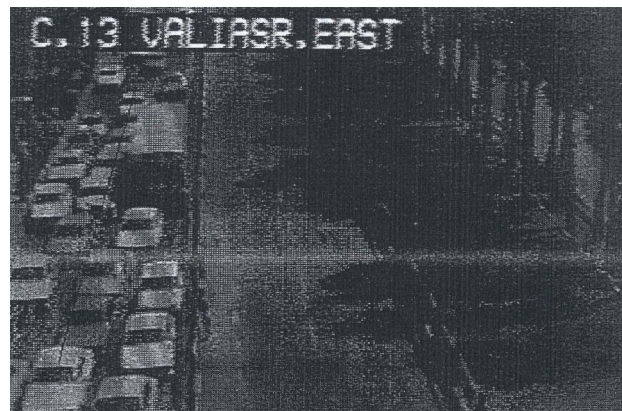
#### ۵-۱- تعمیم سیستم برای کل چهارراه

ابتدا مسیرهای موجود برای همه خیابان هایی که وارد چهارراه می شوند را مشخص می کنیم. سیستم در هر لحظه با توجه به اینکه کدام چراغ سبز است ، پنجره های مربوط به آن خیابان را فعال می کند. اشکالی که در اینجا با آن مواجه می شویم این است که برای هر تصویر باید توقف را در چهار خیابان تشخیص دهد. این مسئله زمان زیادی از دستگاه می گیرد. برای رفع این مشکل بهتر این است که شکل زمان بندی هر چهارراه را در ابتدا

شکل ۵ و ۶ - نمونه هایی از طول صف های تشخیص داده شده



شکل ۵



شکل ۶

#### ۵- نرخ توزیع خودرو در چهارراه ها

هدف اصلی از این بخش این است که نسبت انتخاب هر مسیر را توسط خودروهایی که وارد چهارراه می شوند پیدا کنیم. باز هم روش پردازش پنجره را در پیش می گیریم. باید پنجره های زیادی در مکان های مختلف تعریف کنیم. در ابتدا کاربر کلیه مسیرهای ممکن را برای سیستم مشخص می کند( مثلاً مسیر مستقیم ، گردش به چپ ، گردش به راست و حتی دور زدن ). سپس با توجه به این مسیر ها پنجره های لازم را برای هر یک اختیار می کنیم. سری اول پنجره ها بر روی محل ورودی از خیابان مورد نظر به چهارراه انتخاب می شود. به جز این سری ، باید سری های دیگری به تعداد مسیرهای تعیین شده مشخص شوند. در مورد مسیرهای مستقیم و دور زدن این پنجره ها موازی پنجره های اصلی تعریف می شوند و برای مسیرهای

مشاهد شد که عملی که بیشترین زمان را از سیستم می گرفت نمایش تصویر بود اگر نمایش تصویر را حذف کنیم کلیه عملیات با سرعت خوبی انجام می شد .

#### پیشنهاد :

اگر برای ادامه کار از شبکه عصبی استفاده شود و تعداد عناصر لایه پنهان و تعداد لایه های پنهان را افزایش دهیم و با تغییر تعداد ورودی و خروجی های شبکه عصبی و دادن اطلاعات مناسب می توان اطلاعات بیشتری از پنجره گرفت و حتی شاید شناسایی و ردیابی خودروها نیز امکان پذیر است .

#### مراجع :

از سری مجموعه مقالات اولین کنفرانس ماشین بینایی و پردازش تصویر اسفند ۷۹ - بیرجند

۱- طراحی یک الگوریتم آشکار سازی و ردیابی چندین

هدف متحرک در تصاویر ویدیویی پیاده سازی سریع

آن در یک سیستم مبتنی بر PC

۲- الگوریتمی برای جستجوی بینایی اشیا در تصویر رنگی

با استفاده از شبکه عصبی - صادق دری - سید رضا

افراز

### 3- moving target classification form

Heaydar Mokhtari Farivar

از سری مجموعه مقالات دومین کنفرانس ماشین بینایی و

پردازش تصویر- بهمن ۸۱ - دانشگاه خواجه نصیر

۱- یک روش مبتنی بر پردازش تصویر برای اندازه گیری

پارامتر ترافیکی تعداد خودروها میترا طبایی - منصور

جوزاد

۲- تعیین موقعیت وسائط نقلیه با استفاده از پردازش

تصویر سید محمد سادات حسینی - دکتر محمود

فتحی - دگتر منوچهر وزیری

کاربرد پردازش تصویر اطلاعات تصویری در ترافیک شهری (

گزارش تحقیقاتی شرکت کنترل ترافیک )

از سری مقالات سومین کنفرانس مهندسی برق ایران - دانشگاه

علم . صنعت ۱۳۷۴

به سیستم بدهیم. با این کار ، سیستم هر لحظه تنها یک یا دو خیابان را که در حال حاشر فعال می باشند ، از نظر چراغ کنترل می کند. هر زمان که چراغ آنها قرمز شد سیستم می داند که کدام مسیر را باید پردازش کند.

#### نتیجه گیری

در این پرو ژه الگوریتم هایی برای تعیین پارامترهای کمی ترافیک براساس پردازش تصویر بررسی شد . اساس این الگوریتم ها بر مبنای روش پنجره یعنی پردازش بخش های مهم تصاویر است . از آنجا که سرعت پردازش در این پروژه بسیار مهم بود در کلیه موارد سعی شد از الگوریتم های ساده پردازش استفاده شود . نیز در مورد اندازه پنجره ها مطالعاتی انجام گرفت که در هر مورد با کوچکترین پنجره ممکن و در نتیجه در کوتاهترین زمان بتوان پاسخ مورد نظر را بدست آورد .

#### نتایج :

برای شمارش خودروها با توجه به نوع خیابان و پارامترهای دوربین بهترین اندازه برای پنجره انتخاب شده و محل آن نیز تعیین می شود سپس بر روی پنجره ها فیلترهای لازم اعمال شده و با یک الگوریتم تصمیم گیری تعداد خودروها محاسبه می شود

اگر طول پنجره ها نصف شود می توان در مورد هر پنجره با توجه به پنجره های مجاور آن تصمیم گیری کرد و در نتیجه دقت بالاتر می رود و برای بهبود سیستم در تصمیم گیری باید از یک شبکه عصبی استفاده شود و اشکال آن تنها این است که برای هر خیابان شبکه باید جداگانه آموزش ببیند .

برای اندازه گیری سرعت هم از دو پنجره با فاصله مشخص استفاده گردید و برای تعیین این مساله که آیا خودروی تشخیص داده شده در پنجره دوم همان خودروی است که در پنجره اول بود بهتر است از یک شبکه عصبی استفاده شود .

برای اندازه گیری نرخ توزیع خودرو در چهارراه تعداد پنجره ها را بیشتر کردیم و بر اساس تعداد خودروهای عبوری از هر پنجره نرخ توزیع خودرو بدست می آید

برای اندازه گیری طول صف یک پنجره عمودی در نظر گرفته شد و با تفاضل تصاویر متوالی توقف تشخیص داده می شد و سپس حضور یا عدم حضور بررسی می شد و طول صف بدست می آمد برای تشخیص خودرو در شب نیز ابتدا کلیه اشیا روشن موجود در پنجره جدا می شد .