

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مهندسی برق و ریاتیک

پایان نامه کارشناسی ارشد مهندسی سیستم‌های الکترونیک دیجیتال

**طراحی و پیاده‌سازی سامانه تشخیص پلاک سریع در بستر اندروید**

نگارنده: جواد خاکشور کامه‌علیا

استاد راهنما

دکتر حسین خسروی

بهمن ۱۳۹۵

ب



مدیریت تحصیلات تکمیلی

بسمه تعالی

شماره: ۱۴۵۸/۲۰۱

تاریخ: ۱۳۹۵/۱۱/۱۸

ویرایش: -----

فرم شماره ۷: صورتجلسه دفاع از پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای  
جواد خاکشور کامه علیا به شماره دانشجویی ۹۳۰۶۷۸۴ رشته مهندسی برق گرایش الکترونیک (سیستم) که  
در تاریخ ۹۵/۱۱/۱۸ تحت عنوان:

طراحی و پیاده سازی سامانه تشخیص پلاک سریع در بستر اندروید

با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

|   |                                    |                                |
|---|------------------------------------|--------------------------------|
| قبول (با درجه: <u>بسیار خوب</u> امتیاز: <u>۱۸.۲۷</u> )                            | <input type="checkbox"/> دفاع مجدد | <input type="checkbox"/> مردود |
| نوع تحقیق: نظری <input type="checkbox"/> عملی <input checked="" type="checkbox"/> |                                    |                                |

۱- عالی (۲۰-۱۹)

۲- بسیار خوب (۹۹-۱۸)

۳- خوب (۹۹-۱۷)

۴- قابل قبول (۹۹-۱۵)

۵- نمره کمتر از ۱۴ غیر قابل قبول

| امضاء | مرتبه علمی | نام و نام خانوادگی | عضو هیأت داوران                 |
|-------|------------|--------------------|---------------------------------|
|       | استاد      | صیغ حسینی          | ۱- استاد راهنمای اول            |
| —     | —          | —                  | ۲- استاد راهنمای دوم            |
| —     | —          | —                  | ۳- استاد مشاور                  |
|       | استاد      | عما رابراهمی       | ۴- نماینده شورای تحصیلات تکمیلی |
|       | دانشیار    | عدیه احمدی فرد     | ۵- استاد ممتحن اول              |
|       | دانشیار    | علی براری          | ۶- استاد ممتحن دوم              |

نام و نام خانوادگی رئیس دانشکده:

تاریخ و امضاء و مهر دانشکده:



تقدیم بہ:

پدر، مادر، عزیزم و برادر بزرگوارم

کہ از نگاہشان صلابت

از رفتارشان محبت

و از صبرشان ایستادگی را آموختم

## **تشکر و قدردانی:**

با سپاس فراوان از زحمات استاد محترم و گرانقدر جناب آقای دکتر خسروی که از ابتدای راه و در طی انجام این پژوهش، با راهنمایی‌های خود مرا در نگارش این پایان‌نامه یاری نمودند.

## تعهد نامه

اینجانب **جواد خاکشور کامه‌علیا** دانشجوی دوره کارشناسی ارشد رشته برق - الکترونیک دانشکده برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه طراحی و پیاده‌سازی سامانه تشخیص پلاک سریع در بستر اندروید تحت راهنمایی **دکتر حسین خسروی** متعهد می‌شوم:

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام «دانشگاه صنعتی شاهرود» و یا «Shahrood University of Technology» به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافته‌های آن‌ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

### تاریخ

### امضای دانشجو

### مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

## چکیده

تا قبل از فراگیر شدن گوشی‌های هوشمند، سامانه‌های تشخیص پلاک خودرو، معمولاً مبتنی بر یک دوربین با وضوح بالا، رایانه‌ای قدرتمند و فرستنده‌ای مجزا برای ارسال اطلاعات بودند. در سال‌های اخیر برخی از سازمان‌ها مانند شرکت‌های متولی پارکبان و راهنمایی و رانندگی به منظور ارائه‌ی خدمات توسط گوشی‌های هوشمند، پیشنهاد پیاده‌سازی سامانه<sup>1</sup> ALPR بر روی سیستم‌عامل اندروید را مطرح کرده‌اند. در این پایان‌نامه تلاش کردیم تا این نیاز را برطرف کنیم و سامانه تشخیص پلاک را بر روی گوشی‌های هوشمند پیاده‌سازی کنیم. هرچند در این کاربرد، بلادرنگ بودن خیلی ضروری نیست اما سعی کرده‌ایم تا الگوریتمی سریع برای آشکارسازی و شناسایی پلاک ارائه کنیم. برخلاف روش‌هایی با پیچیدگی محاسباتی بالا، روش ساده و مؤثری را برای این منظور به کار گرفته‌ایم. ابتدا با لبه‌یابی عمودی و سپس افکنش افقی نواحی احتمالی نامزد پلاک مشخص می‌شود. پس از آن با الگوریتمی نویسه‌ها جدا شده و در ادامه ارقام پلاک بازشناسی می‌شود. همچنین برای افزایش سرعت اجرای برنامه از پردازش موازی<sup>2</sup> نیز استفاده کرده‌ایم.

روش پیشنهادی بر روی عکس‌های گرفته‌شده توسط دوربین تلفن همراه مورد ارزیابی قرار گرفته است. این سامانه ابتدا با استفاده از زبان ++C و کتابخانه OpenCV در نرم‌افزار Visual Studio مورد ارزیابی قرار گرفته است، سپس با استفاده از زبان Java و کتابخانه OpenCV بر روی گوشی اندروید پیاده‌سازی شده است. کمترین میانگین زمان پردازش ۰/۶۷s و درصد آشکارسازی نیز ۹۶/۱٪ بوده است.

**کلیدواژه‌ها:** اندروید، پلاک خودرو، افکنش افقی، شبکه عصبی، ALPR، پردازش موازی

---

<sup>1</sup> Automatic License Plate Recognition (ALPR)

<sup>2</sup> Parallel processing

# فهرست مطالب

| صفحه | عنوان   |
|------|---|
| ۱    | <b>فصل اول : مقدمه</b>  |
| ۲    | ۱-۱ مقدمه   |
| ۳    | ۲-۱ ضرورت انجام پژوهش   |
| ۳    | ۱-۲-۱ پارکومتر  |
| ۳    | ۱-۲-۱ سامانه‌ی شناسایی با فرکانس رادیویی                          |
| ۴    | ۳-۲-۱ کارت پارک مبتنی بر NFC                                      |
| ۴    | ۴-۲-۱ کارت پارک مبتنی بر NFC و پلاک خودرو                         |
| ۵    | ۵-۲-۱ کارت پارک مبتنی بر شماره پلاک با استفاده از گوشی تلفن همراه |
| ۶    | ۳-۱ پیش‌نیازهای سامانه تشخیص پلاک خودرو                           |
| ۶    | ۴-۱ کاربردهای سامانه‌ی آشکارسازی و شناسایی پلاک                   |
| ۷    | ۱-۴-۱ اندازه‌گیری سرعت متوسط خودرو بر اساس مسافت طی شده           |
| ۸    | ۲-۴-۱ محاسبه مدت سفر  |
| ۸    | ۳-۴-۱ اخذ عوارض جاده‌ها و بزرگراه‌ها به صورت خودکار               |
| ۹    | ۴-۴-۱ مدیریت پارکینگ  |
| ۹    | ۵-۴-۱ ثبت خودکار تخلفات   |
| ۱۱   | <b>فصل دوم : پیشینه پژوهش</b>                                     |



|    |  |
|----|--|
| ۱۲ | ۱-۲ مروری بر پژوهش‌های پیشین               |
| ۱۲ | ۲-۲ برخی روش‌های آشکارسازی پلاک خودرو      |
| ۱۳ | ۱-۲-۲ لبه‌یابی                             |
| ۱۶ | ۳-۲ ویژگی رنگ                              |
| ۱۹ | ۴-۲ روش‌های مبتنی بر تبدیلات               |
| ۲۱ | ۵-۲ برخی روش‌های دیگر                      |
| ۲۲ | ۶-۲ جدا سازی اعداد و حروف                  |
| ۲۳ | ۱-۶-۲ واکاوی اجزای متصل                    |
| ۲۴ | ۲-۶-۲ افکنش افقی و عمودی                   |
| ۲۵ | ۷-۲ برخی از راهکارهای تشخیص نویسه‌های پلاک |
| ۲۹ | <b>فصل سوم : روش پیشنهادی</b>              |
| ۳۱ | ۱-۳ حذف نویز                               |
| ۳۳ | ۲-۳ لبه یاب سوبل                           |
| ۳۵ | ۳-۳ بکار بردن افکنش                        |
| ۳۷ | ۴-۳ اعمال عملگر مورفولوژی                  |
| ۳۹ | ۵-۳ واکاوی اجزای متصل                      |
| ۴۰ | ۶-۳ تفکیک نویسه‌ها                         |
| ۴۲ | ۷-۳ تشخیص نویسه‌های پلاک                   |

| عنوان  | صفحه |
|--|------|
| ۱-۷-۳ پایگاه داده .....  | ۴۲   |
| ۲-۷-۳ استخراج ویژگی .....  | ۴۴   |
| ۳-۷-۳ آموزش شبکه عصبی .....  | ۴۷   |
| ۸-۳ پردازش موازی .....   | ۴۸   |
| ۱-۸-۳ برنامه‌نویسی موازی .....   | ۴۹   |
| ۱-۱-۸-۳ برنامه نویسی با استفاده از CUDA .....  | ۵۰   |
| ۲-۱-۸-۳ برنامه نویسی با استفاده از OpenCL .....  | ۵۰   |
| ۳-۱-۸-۳ افزایش سرعت اجرای برنامه با الگوریتم‌های چند ریسمانی .....                         | ۵۱   |
| <b>فصل چهارم : نتایج تجربی</b> .....   | ۵۵   |
| ۱-۴ نتایج تجربی .....  | ۵۶   |
| <b>فصل پنجم : جمع بنی و پیشنهادات</b> .....  | ۶۹   |
| ۱-۵ جمع بندی .....   | ۷۰   |
| ۲-۵ پیشنهادها .....  | ۷۱   |
| <b>پیوست الف : نحوه ایجاد یک پروژه و اضافه کردن کتابخانه OpenCV در StudioAndroid</b> ..... | ۷۳   |
| <b>مراجع</b> .....   | ۹۵   |

# فهرست شکل‌ها

صفحه

عنوان

- شکل ۱-۱ روش محاسبه سرعت متوسط خودرو ..... ۷
- شکل ۱-۲ تشخیص محل پلاک از روی تراکم چگالی لبه ..... ۱۴
- شکل ۲-۲ انواع پلاک‌های مورد استفاده در ایران ..... ۱۶
- شکل ۳-۲ نوار آبی رنگ پلاک ملی ..... ۱۷
- شکل ۴-۲ پیمایش ستونی تصویر برای جستجوی حاشیه آبی رنگ پلاک ..... ۱۸
- شکل ۵-۲ اثر آستانه‌گیری تطبیقی ..... ۲۲
- شکل ۶-۲ جداد سازی اعداد و حروف با استفاده از روش واکاوی اجزای متصل (CCA). الف) تصویر پلاک، ب) تصویر دو سطحی، ج) جداسازی نویسه‌ها. .... ۲۳
- شکل ۷-۲ جداسازی حروف بر اساس روش هیستوگرام ..... ۲۴
- شکل ۱-۳ ساختار کلی یک سیستم تشخیص خودکار پلاک خودرو ..... ۳۰
- کل ۲-۳ تصاویر به‌دست آمده قبل از اعمال فیلتر میانه ..... ۳۲
- کل ۳-۳ تصاویر به‌دست آمده بعد از اعمال فیلتر میانه ..... ۳۲
- شکل ۴-۳ تاثیرات لبه یاب عمودی و آستانه‌گیری از تصویر لبه یابی شده ..... ۳۴
- شکل ۵-۳ منحنی حاصل روش افکنش افقی ..... ۳۵
- شکل ۶-۳ منحنی حاصل روش افکنش افقی ..... ۳۶
- شکل ۷-۳ تاثیر اعمال افکنش به تصویری که پلاک در آن نسبت به افق زاویه ندارد ..... ۳۷
- شکل ۸-۳ تاثیر اعمال افکنش به تصویری که پلاک در آن نسبت به افق زاویه دارد ..... ۳۷
- شکل ۹-۳ تاثیر اعمال عملگر مورفولوژی گسترش بر تصویر ..... ۳۸

## عنوان

## صفحه

- شکل ۳-۱۰ نامزدهای درست و نادرست محل پلاک ..... ۳۹
- شکل ۳-۱۱ مراحل روش تفکیک نویسه‌ها ..... ۴۰
- شکل ۳-۱۲ نویسه‌هایی که به درستی استخراج شده ..... ۴۱
- شکل ۳-۱۳ نمونه‌هایی که به اشتباه به جای نویسه‌ها استخراج شده است ..... ۴۱
- شکل ۳-۱۴ مراحل تشخیص نویسه‌های پلاک ..... ۴۲
- شکل ۳-۱۵ برخی از نمونه‌های نامعتبری ..... ۴۴
- شکل ۳-۱۶ برخی از تصاویر پایگاه داده حروف و اعداد فارسی ..... ۴۴
- شکل ۳-۱۷ تصویر حرف "ل"، "د" و "م" و ویژگی‌های استخراج شده‌ی توسط بلوک  $4 \times 4$  و  $10 \times 10$  ..... ۴۵
- شکل ۳-۱۸ الف) عملکرد برنامه چندریسمانی، ب) عملکرد برنامه تک‌ریسمانی ..... ۵۲
- شکل ۳-۱۹ زمان اجرای برنامه چندریسمانی بر روی CPU الف) چندهسته‌ای و ب) تک‌هسته‌ای ..... ۵۳
- شکل ۴-۱ روند افزایش و سپس کاهش سرعت بر اساس انتخاب تعداد ریسمان‌ها ..... ۵۸
- شکل ۴-۲ خروجی برنامه برای تصاویری با وضوح  $1080 \times 1920$  در فاصله ۴ تا ۸ متری از دوربین ..... ۶۳
- شکل ۴-۳ خروجی برنامه برای تصاویری با وضوح  $1080 \times 1920$  در فاصله ۲ تا ۴ متری از دوربین ..... ۶۴
- شکل ۴-۴ خروجی برنامه برای تصاویری با وضوح  $540 \times 920$  در فاصله ۱ تا ۲ متری از دوربین ..... ۶۵
- شکل ۴-۵ خروجی برنامه برای تصاویری با وضوح  $240 \times 320$  در فاصله ۱ متری از دوربین ..... ۶۶
- شکل ۴-۶ خروجی‌های نادرست ..... ۶۷

# فهرست جدول‌ها

| عنوان  | صفحه |
|--|------|
| جدول ۱-۳ برخی از نمونه‌های آموزش.....  | ۴۲   |
| جدول ۱-۴ نتایج خروجی نرم‌افزار برای الگوریتم ۱ تا ۵ ریسمانه برای تصاویری با وضوح مختلف .....             | ۵۵   |
| جدول ۲-۴ نتایج زمانی خروجی نرم‌افزار برای ۱ تا ۱۷ ریسمان .....   | ۵۷   |
| جدول ۳-۴ بررسی خروجی نرم‌افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح $۱۰۸۰ \times ۱۹۲۰$ ..... | ۵۹   |
| جدول ۴-۴ بررسی خروجی نرم‌افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح $۹۲۰ \times ۵۴۰$ .....   | ۶۱   |
| جدول ۵-۴ بررسی خروجی نرم‌افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح $۳۲۰ \times ۲۴۰$ .....   | ۶۲   |

# فصل اول

مقدمه

## ۱-۱ مقدمه

مطالعه و بررسی تشخیص خودکار پلاک خودرو، اواخر دهه ۱۹۷۰ میلادی صورت پذیرفت. این سامانه، در سال ۱۹۷۶ برای اولین بار مورد آزمایش قرار گرفت و نمونه اولیه آن در سال ۱۹۷۹ به کار گرفته شد. رشد سریع فن‌آوری و توسعه سامانه‌های حمل‌ونقل هوشمند<sup>۱</sup>، ITS، زندگی راحت‌تری را برای جوامع بشری فراهم آورد. در این زمینه تشخیص خودکار پلاک خودرو نقش مهمی در پیشبرد آسان بسیاری از امور از جمله کشف خودروهای سرقتی، مدیریت پارکینگ‌ها، کنترل ترافیک و اعمال قانون رانندگان متخلف ایفا می‌کند. در روش‌های سنتی تشخیص پلاک معمولاً نیازمند یک دوربین با وضوح بالا برای گرفتن تصاویر با کیفیت خوب، یک رایانه قدرتمند برای پردازش الگوریتم‌های پیچیده و فرستنده‌ای مجزا برای ارسال اطلاعات هستیم.

با پیشرفت‌های عظیم در عرصه فن‌آوری ساخت تلفن‌های همراه، امروزه گوشی‌های هوشمند با پردازنده‌های قوی و دوربین‌های با وضوح بالا، وجود دارد که می‌توانند نیازهای یک سامانه ALPR را برآورده سازند. مزیت این گوشی‌ها، قیمت ارزان، امکانات متنوع از جمله سامانه پیامک، اینترنت پرسرعت با استفاده از شبکه 4G<sup>۲</sup> و داشتن انواع حس‌گرها و نیز حمل آسان نسبت به سایر سامانه‌های تشخیص پلاک خودرو است. در این پایان نامه هدف ارائه نرم افزار برای سیستم عامل اندروید است که توانایی خواندن پلاک را داشته باشد. از سامانه ارائه‌شده در این پایان نامه می‌توان به جای سامانه‌های متداول

---

<sup>۱</sup> Intelligent Transport Systems

<sup>۲</sup> Fourth Generation

پارکبان، دستگاه‌های ثبت تخلفات افسران راهنمایی و رانندگی و همچنین در پارکینگ‌های عمومی استفاده نمود.

## ۲-۱ ضرورت انجام پژوهش

امروزه روند کنترل ترافیک با افزایش وسایل نقلیه و توقف آن‌ها در خیابان‌ها و پارکینگ‌های عمومی، کار دشواری شده است. داده‌های ترافیکی ممکن است توسط حس‌گر یا نیروی انسانی ثبت شوند که زمان‌بر و پرهزینه است. در این راستا تا کنون فن‌آوری‌هایی ارائه شده است. در ذیل به برخی از این فن‌آوری‌ها اشاره می‌کنیم و در پایان راهکار خود را بیان می‌کنیم.

### ۱-۲-۱ پارکومتر

در این شیوه صاحب خودرو پس از توقف باید به دستگاه پارکومتر نصب شده در آن محدوده مراجعه کند و هزینه را به صورت الکترونیکی پرداخت نماید. از معایب این روش نیاز به آموزش همگانی، سرگردانی و اتلاف وقت رانندگان، لزوم همراه داشتن کارت هوشمند شهروندی، هزینه قابل توجه نصب و نگهداری دستگاه‌های پارکومتر، اتکای زیاد به نیروی پلیس به دلیل هوشمند نبودن آن‌ها، کاهش اشتغال زایی و عدم مسئولیت‌پذیری مردم نسبت به پرداخت هزینه است.

### ۲-۲-۱ سامانه‌ی شناسایی با فرکانس رادیویی

سامانه‌ی شناسایی با فرکانس رادیویی<sup>۱</sup>، RFID، با چسباندن یک برچسب که اطلاعات خودرو بر روی حافظه آن ذخیره شده است، خودرو شناسایی می‌شود. یک گیرنده رادیویی ثابت یا متحرک وظیفه

<sup>۱</sup> Radio-frequency identification



دریافت اطلاعات خودرو را برعهده دارد. این سامانه در ورودی بزرگراه‌ها و پارکینگ‌ها مورد استفاده قرار می‌گیرد. سامانه ذکر شده دارای مزایایی چون دقت بالا، پیچیدگی اندک و معایبی چون هزینه‌ای اضافی برای صاحبان خودرو و عوامل کنترل‌کننده، مجهز نبودن همه وسایل نقلیه به آن و تداخل فرکانسی است.

### ۱-۲-۳ کارت پارک مبتنی بر NFC<sup>۱</sup>

روش کار با این سامانه مشابه دستگاه‌های مبتنی بر RFID است. در این روش نیز باید برچسب مخصوص NFC بر روی خودرو نصب شود. سپس مأموران پارک حاشیه‌ای، گوشی تلفن همراه یا دستگاه‌هایی که مجهز به تکنولوژی NFC هستند را به برچسب نزدیک می‌کنند تا هزینه پارک محاسبه شود. مزیت آن نسبت به RFID این است که به سخت‌افزار خاصی نیاز ندارد و می‌توان از برخی تلفن‌های همراه که فن‌آوری NFC در آن‌ها وجود دارد به عنوان دستگاه پارکبان استفاده کرد. مشکل تداخل فرکانسی نیز به خاطر برد کم این فن‌آوری اتفاق نمی‌افتد. معایب آن نیز تحمیل هزینه اضافی برای صاحب خودرو و ثبت نشدن پلاک خودرو است.

### ۱-۲-۴ کارت پارک مبتنی بر NFC و پلاک خودرو

تمامی مراحل این روش مشابه ۱-۲-۳ است. با این تفاوت که در این شیوه پلاک خودرو نیز ثبت می‌شود. اما با توجه به کثرت خرید و فروش خودروها قطعاً تعویض پلاک‌های زیادی صورت می‌پذیرد

که در صورت عدم مراجعه اشخاص جهت تغییر اطلاعات ثبت شده مغایرت‌های زیادی در مشخصات پلاک و برچسب NFC به وجود خواهد آمد که پس از گذشت زمان تعداد این مغایرت‌ها افزایش می‌یابد. بنابراین با تعویض پلاک خودرو باید برچسب NFC نیز تعویض شود.

## ۱-۲-۵ کارت پارک مبتنی بر شماره پلاک با استفاده از گوشی تلفن همراه.

راهکار ارائه شده در این سامانه مبتنی بر وارد کردن شماره پلاک به صورت دستی در تلفن همراه است. در سامانه ارائه شده ایرادات سامانه پارکومتر، کارت پارک مبتنی بر RFID و NFC برطرف شده است. اما هنوز معایبی وجود دارد که می‌توان به وارد کردن شماره پلاک به صورت دستی، بالارفتن احتمال خطای انسانی، زمان‌بر بودن ثبت پلاک و ایجاد خستگی برای متصدی پارکبان اشاره نمود. با توجه به مطالب ارائه شده، استفاده از سامانه‌ای خودکار برای ثبت پلاک خودرو توسط تلفن‌های هوشمند امری ضروری به نظر می‌رسد تا از صرف هزینه اضافی از نظر سخت‌افزاری و نرم‌افزاری جلوگیری شود. در این راستا می‌توان نرم‌افزارهایی را نیز تولید نمود که علاوه بر ثبت پلاک خودرو و بررسی آنلاین مشخصات آن از سرقتی بودن خودرو نیز اطمینان حاصل نمود همچنین می‌توان توسط موقعیت‌یاب<sup>۱</sup> GPS مشخصات مکان توقف خودرو را ثبت کرد و از آن در جمع‌آوری اطلاعات آماری و پیشبرد برخی امور استفاده نمود.

---

<sup>۱</sup> Global Positioning System

## ۳-۱ پیش‌نیازهای سامانه تشخیص پلاک خودرو

به‌طور کلی یک سامانه تشخیص خودکار پلاک خودرو شامل یک دوربین، نرم‌افزار وابسته به آن و یک رایانه است. این سامانه بهتر است موارد زیر را پوشش دهد.

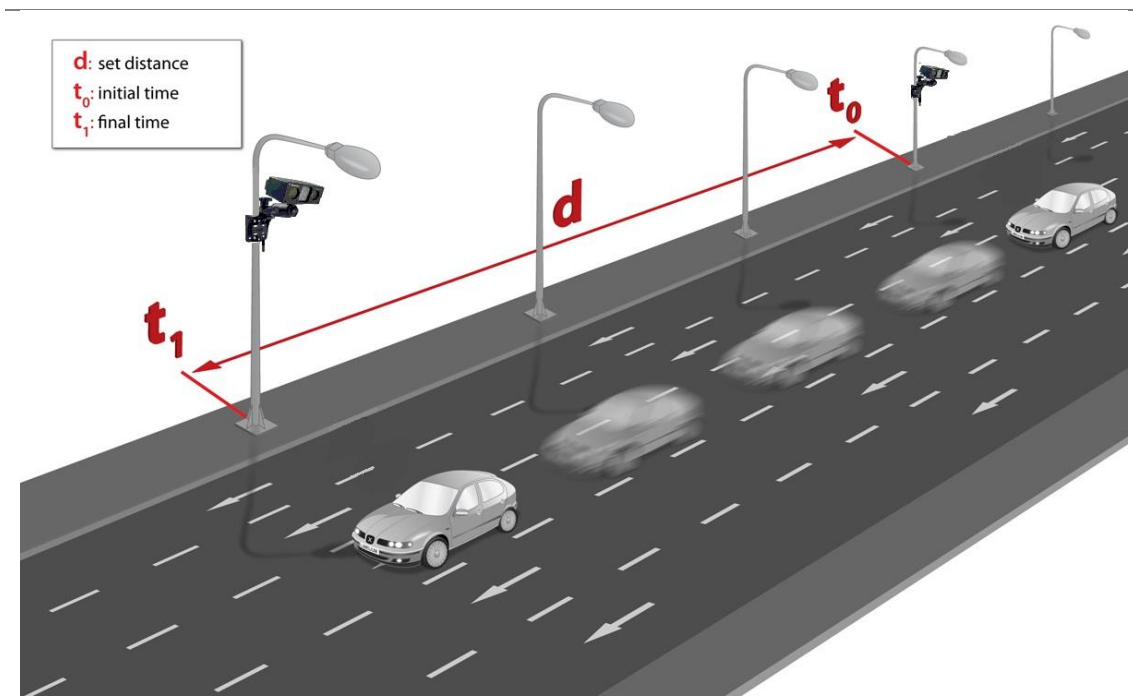
در فضای باز و بسته به درستی عمل کند، در شرایط نوری مختلف به خوبی کار کند، نسبت به عوامل خارجی همچون نویز مقاوم باشد پاسخ نسبتاً سریعی ارائه کند و در شرایط آب و هوایی مختلف پاسخ مناسبی داشته باشد. هرچند مشکلات دیگری نیز مانند تنوع پلاک در برخی مناطق، تغییرات در بافت پلاک به دلیل فرسودگی یا آلودگی و برچسب‌های نصب شده توسط صاحبان خودرو وجود دارد که گاه اجتناب‌ناپذیر است.

## ۴-۱ کاربردهای سامانه‌ی آشکارسازی و شناسایی پلاک

از جمله کاربردهای سامانه تشخیص پلاک می‌توان به کنترل ترافیک و بررسی مسائل امنیتی در حوزه حمل‌ونقل جاده‌ای و شهری [۱]، پرداخت خودکار عوارض جاده‌ای، مدیریت و پرداخت خودکار هزینه پارکینگ‌های عمومی، رهگیری و شناسایی خودروهای سرقتی، کنترل و ثبت بلادرنگ سرعت وسایل نقلیه، کنترل سرعت بر اساس طی مسافت خاص، تخمین مدت سفر [۲]، ثبت خودکار تخلفات، کنترل عبور و مرور خودروهای ترانزیت در مرزها، سامانه‌های متحرک شناسایی پلاک خودرو پلیس، شناسایی پلاک خودروها متخلف در پمپ‌بنزین، کنترل ورودی و خروجی در مناطق حفاظت‌شده اشاره کرد. در ادامه به برخی از این کاربردها می‌پردازیم.

## ۱-۴-۱ اندازه‌گیری سرعت متوسط خودرو بر اساس مسافت طی شده

در این روش طبق شکل ۱-۱ با ثبت پلاک خودرو در ابتدا و انتهای مسیری که در آن دوربین نصب شده است، بر اساس مسافت تعیین شده بین این دو دوربین و مدت زمانی که طول کشیده است تا خودروی مورد نظر فاصله بین این دو دوربین را طی کند می‌توان سرعت متوسط خودرو را محاسبه کرد. این روش برای رانندگان متخلفی استفاده می‌شود که از محل دقیق دوربین‌های نصب شده در معابر جاده‌ای اطلاع داشته و به محض نزدیک شدن به دوربین‌های کنترل سرعت، سرعت خود را کاهش می‌دهند و در سایر مکان‌ها دیگر با سرعت غیرمجاز اقدام به رانندگی می‌کنند. این روش در مقایسه با روش‌های مبتنی بر سرعت سنجی نقطه‌ای برتری دارد.



شکل ۱-۱ روش محاسبه سرعت متوسط خودرو

## ۱-۴-۲ محاسبه مدت سفر

یکی از کاربردهای مهم سامانه‌های ترافیک هوشمند تخمین مدت زمان سفر است. در این راه کار، سامانه‌ی تشخیص پلاک خودرو در نقاط مختلف یک جاده نصب می‌شود (برای مثال در مبدأ و مقصد). بنابراین مدت زمان سفر را به صورت تفکیک شده در زمان‌های مختلف برای هر خودرو محاسبه می‌کند. با تحلیل دقیق آماری مدت زمان طی شده برای تمامی خودروها می‌توان اطلاعات مفیدی را برای تصمیم‌گیری در اختیار عموم قرار داد. در این کاربرد، مسافران می‌توانند پیش از سفر به آمارها و اطلاعات مربوطه مراجعه کنند و تخمینی از مدت زمان سفر میان مبدأ و مقصد خود داشته باشند. اطلاعات دریافتی از این روش معمولاً در نرم‌افزارهای مسیریابی مانند نقشه گوگل<sup>۱</sup> و سایر نرم‌افزارهای مسیریاب استفاده می‌شود.

## ۱-۴-۳ اخذ عوارض جاده‌ها و بزرگراه‌ها به صورت خودکار

از آنجایی که وجود موانع بر سر راه خودروها در عوارضی‌ها باعث کند شدن حرکت، بروز تصادف، ایجاد ترافیک و همچنین آلودگی محیط زیست می‌شود، راه‌های مختلفی برای رفع این مشکل ارائه شده است. یکی از این راه‌ها استفاده از سامانه‌ی تشخیص خودکار پلاک خودرو است. در این روش، خودروها بدون نیاز به توقف از عوارضی‌ها عبور می‌کنند و سامانه‌ی تشخیص پلاک خودرو شماره پلاک آن‌ها را ثبت می‌کند. بر اساس شماره پلاک، عوارض مربوطه محاسبه می‌شود و راننده ملزم به پرداخت عوارض در

---

<sup>1</sup> Google Mape

زمان مشخصی خواهد بود. در صورت عدم پرداخت عوارض در زمان مقرر، خودرو طبق قانون جریمه خواهد شد.

#### ۴-۴-۱ مدیریت پارکینگ

یکی دیگر از کاربردهای مهم سامانه تشخیص پلاک، اجازه تردد به خودروهای مجاز است. این سیستم را می‌توان در پارکینگ‌های خصوصی، پارکینگ مجتمع‌های مسکونی و پارکینگ‌های ادارات، بیمارستان‌ها و مناطق امنیتی به کار گرفت. در این روش فقط خودروهایی اجازه عبور خواهند داشت که قبلاً شماره پلاک آن‌ها در پایگاه داده مربوط به آن مکان ثبت شده باشند.

#### ۴-۵-۱ ثبت خودکار تخلفات

ثبت خودکار تخلفات همان‌طور که از نامش پیداست زمانی عمل خواهد کرد که تخلفی صورت گرفته باشد مانند حرکات غیرمتعارف، سرعت بیش از حد مجاز، عبور از چراغ قرمز، ورود به طرح ترافیک. در تمامی موارد ذکر شده بر اساس الگوریتم‌های مختلف پردازش تصویر مانند ردیابی جسم متحرک پس از احراز تخلف صورت گرفته، پلاک خودرو متخلف به صورت خودکار ثبت شده و اعمال قانون خواهد شد.

ادامه‌ی ساختار پایان‌نامه بدین شرح است: در فصل دوم به برخی پژوهش‌های صورت گرفته در حوزه تشخیص خودکار پلاک خودرو از جمله تعیین محل پلاک، جداسازی نویسه‌ها و تشخیص نویسه‌ها اشاره خواهیم کرد. فصل سوم به روش‌های پیشنهادی مورد استفاده در پایان‌نامه خواهیم پرداخت. فصل

چهارم نتایج به دست آمده را نشان خواهیم داد و در نهایت جمع بندی و پیشنهادهای پایان نامه را در

فصل پنجم ارائه می کنیم.

# فصل دوم

## پیشینه پژوهش



## ۱-۲ مروری بر پژوهش‌های پیشین

سامانه‌ی تشخیص خودکار پلاک خودرو شامل سه بخش اساسی از جمله دوربین، واحد پردازش و نرم‌افزار مرتبط با آن است. در اینجا در مورد بخش نرم‌افزاری آن بحث خواهد شد. در ادامه قسمت‌ها مختلف آن را بیان کرده و عملکرد هر بخش را بر اساس روش‌های پیشنهادی شرح خواهیم داد.

هر سیستم نرم‌افزاری تشخیص پلاک خودرو شامل سه بخش اساسی است.

❖ تشخیص محل پلاک

❖ جداسازی اعداد و حروف

❖ تشخیص اعداد و حروف

از آنجایی که صحت عملکرد مراحل دوم و سوم در گرو مرحله اول است؛ بنابراین دقت سامانه تشخیص خودکار پلاک وابسته به مرحله اول خواهد بود. از این رو یکی از مهم‌ترین بخش‌ها، تشخیص صحیح محل پلاک خودرو است. در ادامه به برخی از روش‌های موجود که بر اساس سه مرحله فوق طراحی شده‌اند اشاره خواهیم کرد.

## ۲-۲ برخی روش‌های آشکارسازی پلاک خودرو

روش‌های گوناگونی برای مراحل مختلف شناسایی پلاک خودرو در مراجع مختلف ارائه شده است. اصول کلی برخی از روش‌های پرکاربرد که معمولاً پلاک با توجه به اطلاعات حاصل از آن‌ها شناسایی

می‌شود عبارت‌اند از استفاده از اطلاعات لبه<sup>۱</sup>، بافت تصاویر سطح خاکستری<sup>۲</sup> و استفاده از اطلاعات رنگی تصاویر. در ادامه به برخی از راهکارهای ارائه‌شده خواهیم پرداخت.

## ۱-۲-۲ لبه‌یابی

یکی از روش‌های ارائه‌شده در مقالات، استفاده از فیلتر سوبل برای استخراج محل پلاک است که در [۳] به آن اشاره‌شده است. در این شیوه میزان حجم پراکنش<sup>۳</sup> و گرادیان<sup>۴</sup> محلی در تصویر محاسبه می‌شود. کلیت این ایده بر اساس اختلاف میزان روشنایی بین نویسه‌های پلاک و زمینه پلاک است که به دلیل وجود لبه‌های زیاد ایجادشده توسط نویسه‌های پلاک مورد استفاده قرار می‌گیرد. در [۴] روش دیگری بر مبنای لبه‌یابی ارائه شده است که پس از بهبود کیفیت تصویر مورد نظر، لبه‌های عمودی با استفاده از لبه‌یاب عمودی سوبل به دست می‌آیند. در ادامه لبه‌های اضافی و نویز پس‌زمینه حذف شده و در نهایت به کمک یک پنجره با اندازه‌ی مشخص به عنوان جستجوگر پلاک و کانالو آن با تصویر حاصل از لبه‌یابی با در نظر گرفتن بیشترین چگالی، محل پلاک کشف خواهد شد. این روش نسبت به اندازه‌ی پلاک حساس است؛ یعنی این که در صورت تغییر اندازه‌ی پلاک در تصویر، احتمال یافتن مکان آن کاهش خواهد یافت. شکل ۱-۲ عملکرد این روش نشان می‌دهد.

• نجاتیان و حسن‌پور در [۵] ابتدا کیفیت تصویر مورد نظر را بهبود بخشیده و عملگر سوبل عمودی را به

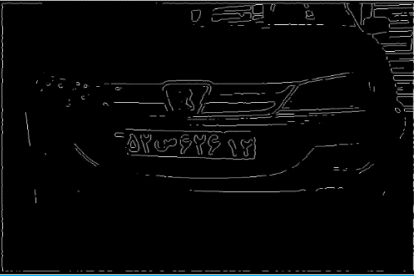


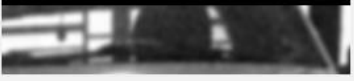

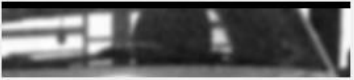
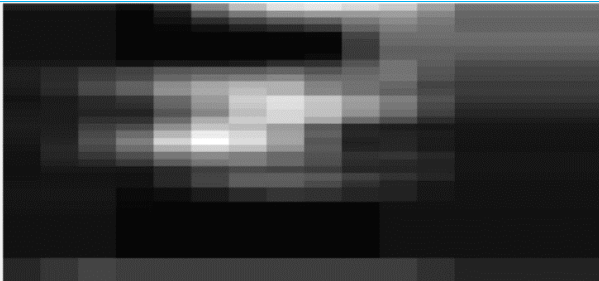
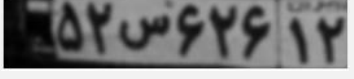
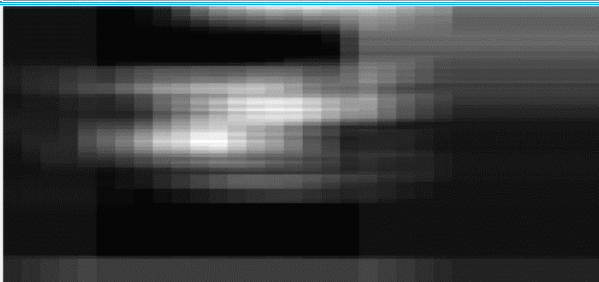
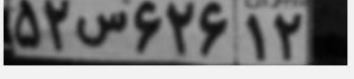
---

<sup>1</sup> Info Edge

<sup>2</sup> Grayscale

<sup>3</sup> Variance

<sup>4</sup> Gradient

|   |   |
|---|---|
|    |   |
|    | <p>پنجره‌بندی با همپوشانی 0.00%<br/>تصویر خروجی استخراج‌شده:</p>    |
|   | <p>پنجره‌بندی با همپوشانی 0.50%<br/>تصویر خروجی استخراج‌شده:</p>   |
|  | <p>پنجره‌بندی با همپوشانی 0.75%<br/>تصویر خروجی استخراج‌شده:</p>  |
|  | <p>پنجره‌بندی با همپوشانی 0.87%<br/>تصویر خروجی استخراج‌شده:</p>  |

شکل ۱-۲ تشخیص محل پلاک از روی تراکم چگالی لبه

تصویر اعمال می‌کند. سپس تصویر را به پنج بخش مشخص تقسیم می‌نمایند. در ادامه با در نظر گرفتن

این موضوع که پلاک معمولاً در وسط تصویر قرار دارد، بخش‌های ابتدا و انتهای تصویر را مورد بررسی

قرار نمی‌دهد و سه بخش باقی مانده را به صورت سطر به سطر بررسی نموده و با یک عنصر مربعی به کمک عملیات گسترش، لبه‌های موجود را به هم متصل می‌کند.

مکانی که دارای بیشترین چگالی است، به عنوان نامزد محل پلاک انتخاب می‌کند. در این مقاله عنوان شده است به دلیل پردازش‌های اولیه‌ای که در جهت بهبود تصویر انجام می‌دهد برای تصاویری که دارای زمینه پیچیده، شدت روشنایی کم، کیفیت پایین و تصاویر تار شده است، عملکرد بهتری دارد. عیب این روش فرض قرار گرفتن محل پلاک در وسط تصویر و حذف بخشی از تصویر قبل از انجام پردازش است. زیرا ممکن است در برخی از تصاویر محل پلاک در وسط تصویر نباشد.

در روشی دیگر با پیمایش چند ردیف در تصویر به تعیین محل پلاک می‌پردازد [۶]. در این شیوه برای مشخص نمودن محل پلاک، تصویر به وسیله ردیف‌هایی با فاصله‌ی مشخص پیمایش می‌شود و میزان لبه‌های هر ردیف شمارش خواهد شد. اگر تعداد نقاط شمارش شده حاصل از لبه یابی بزرگ‌تر از یک آستانه معین شود. وجود پلاک در آن محل محرز می‌شود. در غیر این صورت الگوریتم با کاهش میزان فواصل بین هر ردیف و در نتیجه افزایش میزان ردیف‌ها، به جستجوی دوباره محل پلاک می‌پردازد. این روش عملکرد سریعی دارد به دلیل این‌که در هر بار جستجوی محل پلاک، تعداد ردیف کمتری را مورد بررسی قرار می‌دهد. در [۷] ابتدا با اعمال لبه یاب به تصویر سطح خاکستری، تصویر لبه یابی شده حاصل می‌شود. سپس با در نظر گرفتن محلی از تصویر که دارای بیشترین چگالی لبه است، مکان پلاک

مشخص خواهد شد. در این روش با فرض این که تصویربرداری تحت تنش‌های محیطی مانند لرزش و شرایط متفاوت آب و هوایی مانند هوای مه‌آلود گرفته شده است، در پیش‌پردازش از تیز کردن لبه استفاده شده است که باعث بهبود عملکرد لبه یابی می‌شود.

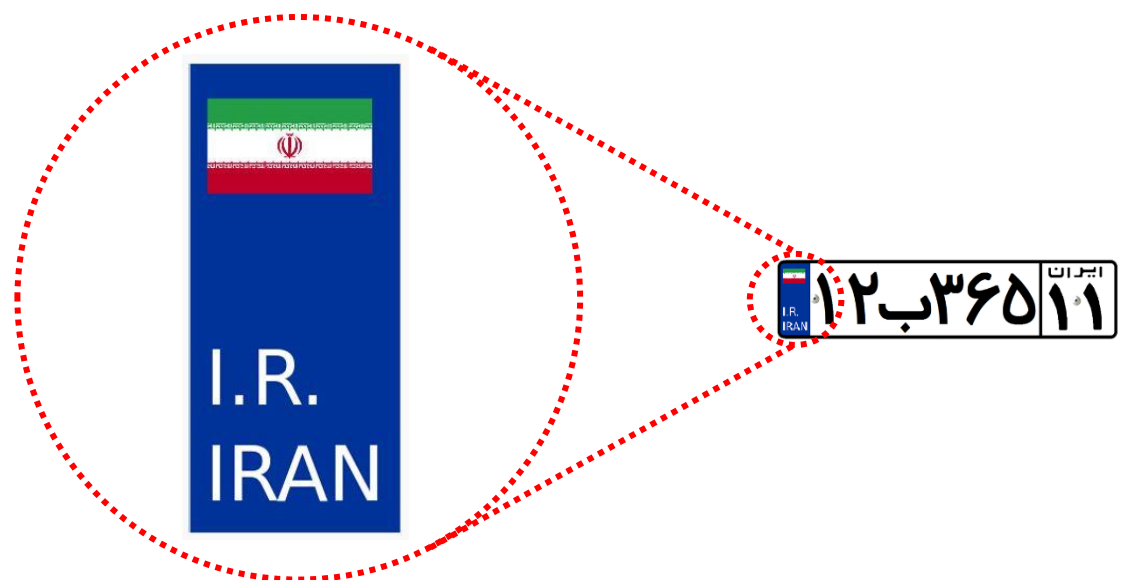
## ۳-۲ ویژگی رنگ

ایده اصلی [۸] بر اساس تغییر رنگ در بین نویسه‌ها و زمینه پلاک و ایجاد لبه رنگی بین آن‌ها شکل گرفته است؛ به عبارت دیگر این مقاله یک آشکارسازی بر اساس لبه رنگی را بیان نموده است. به علت رنگ مختلف پلاک‌های عمومی، اداری و شخصی، آشکارساز ارائه‌شده به سه نوع لبه زرد-مشکی، قرمز-سفید و سفید-مشکی حساس است. شکل ۲-۲ انواع پلاک‌های ایران را نشان می‌دهد. در این روش ابتدا به جستجوی نوار آبی‌رنگ پرداخته تا محل پلاک مشخص شود و سپس با استفاده از لبه‌یاب رنگی ارائه شده نوع پلاک نیز مشخص می‌شود.



شکل ۲-۲ انواع پلاک‌های مورد استفاده در ایران

اشتر و نوردین [۹] روشی را ارائه کرده‌اند که در قیدوبند اندازه پلاک نبوده و آن را در هر ابعاد و شرایطی خواهد یافت. این راهکار، برای تشخیص پلاک در یک تصویر به یک ویژگی ثابت نیاز دارد تا به عنوان الگوی مرجع در نظر گرفته شود. ویژگی در نظر گرفته شده در پلاک‌های ملی جدید که مطابق با استانداردهای اروپا تهیه شده‌اند و جود دارد و آن قسمت آبی حاشیه‌ی چپ پلاک است. شکل ۲-۳ نمونه‌ای از پلاک ملی و حاشیه‌ی آبی‌رنگ پلاک را نمایش می‌دهد.



شکل ۲-۳ نوار آبی‌رنگ پلاک ملی

با توجه به وجود ویژگی مذکور جهت شناسایی مکان پلاک خودرو، برخلاف روش‌های متداول که نیاز به انتقال تصویر به سطح خاکستری یا باینری بود، از اطلاعات رنگی استفاده شده است. تشخیص و پردازش، در محیطی رنگی و بر اساس ویژگی‌های رنگی پلاک، انجام می‌شود. برای این منظور کافی است تصویر را به صورت دوره‌ای<sup>۱</sup> از بالا به پایین مورد پیمایش قرار داد. شکل ۲-۴ پیمایش ستونی را

<sup>1</sup> Periodic

نمایش می‌دهد. طبق شکل ۲-۴ هنگام پیمایش ستونی زمانی که چستوجوگر عمودی ابتدا به رنگ آبی سپس سبز، سفید، قرمز و مجدداً آبی برسد و محدوده نقاط رنگی سبز، سفید و قرمز را با اندازه یکسان بیابد آن قسمت را به عنوان نامزد احتمالی مکان پلاک معرفی می‌کند. نسبت حجم رنگ آبی به سایر رنگ‌ها ۱۱ به ۳ است [۱۰].



شکل ۲-۴ پیمایش ستونی تصویر برای جستجوی حاشیه آبی رنگ پلاک

معمولاً نسبت طول به عرض پلاک در تصاویر مختلف ثابت است، بنابراین تنها با داشتن طول و یا عرض نوار آبی رنگ حاشیه پلاک می‌توان طول و عرض کل پلاک را به دست آورد و پلاک را از درون تصویر استخراج نمود در مقاله [۱۱] ابتدا به جستجوی قسمت های آبی رنگ تصویر می‌پردازد. پس از یافتن نقاط آبی رنگ با در نظر گرفتن بیشترین ارزش برای نقاط آبی رنگ تصویر و کمترین ارزش به سایر نقاط رنگی، تصویر تک کانال جدیدی را ایجاد می‌کند که آن یک تصویر سطح خاکستری خواهد

بود. در مرحله بعدی با اعمال یک آستانه به تصویر سطح خاکستری یک تصویر دو سطحی بدست می‌آورد و قسمت‌هایی از تصویر که دارای مقدار یک است را به عنوان قسمت سمت چپ پلاک در نظر می‌گیرد. سپس بر اساس ابعاد قسمت سفید رنگ، پلاک را استخراج می‌کند. این روش در مواردی که خودرو به رنگ آبی باشد برای یافتن محل پلاک با مشکل روبرو می‌شود. عیب روش‌های مبتنی بر رنگ این است که در شرایط نوری مختلف عملکرد مناسبی از خود نشان نمی‌دهند، زیرا در شرایط مختلف نوری رنگ پایداری و اصالت خود را از دست خواهد داد.

## ۴-۲ روش‌های مبتنی بر تبدیلات

روحانی و عامری [۱۲] از تبدیل هاف<sup>۱</sup> به منظور پیدا کردن محل پلاک با استفاده از خطوط موازی حاشیه‌های اطراف پلاک استفاده کرده‌اند. در این روش ابتدا لبه‌های تصویر موردنظر با استفاده از روش لبه‌یابی به دست می‌آید. پس از آن به منظور کشف محل پلاک از تبدیل هاف استفاده شده است. عیب این روش در آن است که شکل‌های شبیه پلاک که دارای خطوط موازی می‌باشند، نیز در نظر گرفته می‌شود؛ و از سرعت قابل قبولی هم برخوردار نیست. یکی از مزیت‌های این نوع تبدیل حل مشکل زاویه‌دار بودن پلاک در تصاویر است. به این صورت که پس از به دست آوردن خطوط اطراف پلاک و محاسبه میزان انحراف و شیب آن، می‌توان این مشکل را برطرف نمود. مزیت انجام این روش در مرحله تشخیص نویسه‌ها خود را نشان خواهد داد. هرچند اگر زاویه پلاک بیش از حد نباشد و شبکه عصبی

---

<sup>1</sup> Hough Transform



که موظف به تشخیص نویسه‌ها است را با حروف دارای اعوجاج و زوایای مختلف، آموزش دهیم؛ دیگر انجام مرحله رفع اعوجاج اهمیت چندانی پیدا نخواهد کرد. در روشی دیگر به منظور پیدا کردن محل پلاک از تبدیل فوریه [۱۳] با عنوان تحلیل طیفی، استفاده شده است. در این روش، دلیل بکار بردن تبدیل فوریه را وجود حروف در تصویر و چینش افقی و تناوب خاص حاصل از شدت نور ایجادشده بین حروف و زمینه پلاک، عنوان شده است.

تبدیل ارائه شده دیگر  $^1$ IFT است [۱۴]. در این راهکار ابتدا لبه‌های عمودی تصویر به دست می‌آیند. مکان‌هایی از تصویر که بیشترین چگالی حاصل از لبه یابی را داشته باشند به عنوان محل پلاک در نظر گرفته می‌شوند. این مهم به وسیله شمارش لبه‌های موجود در همسایگی محلی هر نقطه از تصویر لبه یابی شده به دست می‌آید. معیار ارتباط پیکسل‌ها همسایگی چهارتایی، هشت‌تایی و غیره است. بدین ترتیب، ماتریسی با ابعاد تصویر ورودی ساخته می‌شود که ارزش هر پیکسل آن چگالی لبه‌ها در محل متناظر در تصویر اولیه است. این ماتریس را می‌توان یک ماتریس سطح خاکستری در نظر گرفت. در نهایت محلی که بیشترین چگالی را داشته باشد به عنوان مکان پلاک در نظر گرفته می‌شود. در [۱۵] برای تعیین محل پلاک از نوعی تبدیل موجک گسسته با عنوان DWT استفاده شده است. در این مقاله

---

<sup>1</sup> Image Foresting Transform

عنوان شده است که می‌تواند محل پلاک را در تصاویری با کیفیت متفاوت و پیچیدگی بالا تعیین کند.

هرچند این مقاله بیان کرده است، در تصاویری که رنگ پلاک خیلی نزدیک به رنگ پس‌زمینه باشد

ممکن است تعیین محل آن با مشکل روبرو شود.

## ۵-۲ برخی روش‌های دیگر

رنگ‌های به کاررفته در پلاک به گونه‌ای هستند که بتوان اجزای مختلف پلاک را به راحتی تفکیک

نمود. این تفاوت رنگ سبب می‌شود که تضاد رنگ در این ناحیه زیاد شود. برای یافتن گوشه‌های موجود

در تصویر ابتدا باید لبه‌های تصویر یافته شود. گوشه را می‌توان اشتراک بین دو لبه یا دو ناحیه‌ی مختلف

از تصویر تعریف نمود. چگالی لبه از رابطه‌ی ۱-۲ به دست می‌آید:

$$\rho_E = \frac{n_E}{A} \quad (1-2)$$

در این رابطه  $A$ ،  $n_E$  و  $\rho_E$  به ترتیب مساحت ناحیه محصور، تعداد گوشه‌ها و چگالی گوشه می‌باشند.

در طی فرایند آستانه‌گیری، مقادیر پیکسل‌های موجود در تصویر به دو کلاس سیاه و سفید تقسیم

می‌شود. به صورت کلی روش آستانه‌گیری را می‌توان به صورت رابطه ۲-۲ تعریف نمود [۱۶].

$$\begin{cases} \text{if } inImage(x,y) \geq T & outImage(x,y) = 255 \\ \text{otherwise} & outImage(x,y) = 0.00 \end{cases} \quad (2-2)$$

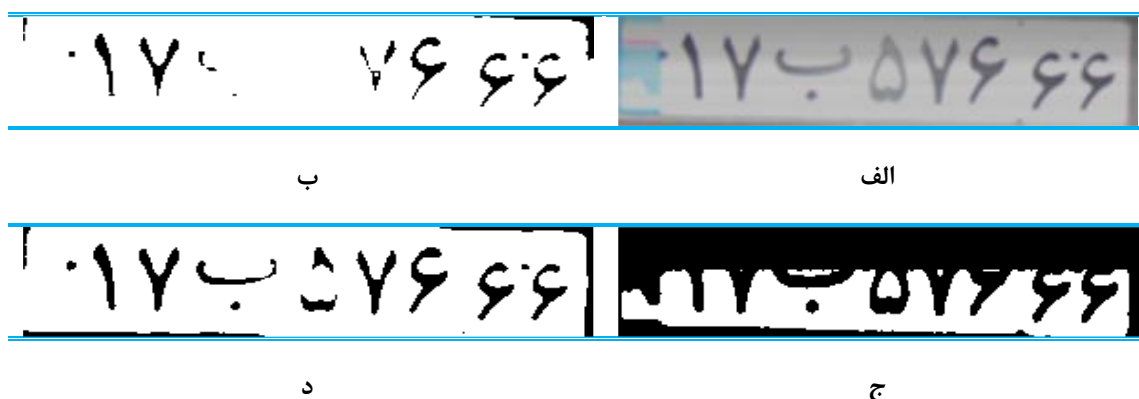
در رابطه ۲-۲  $T$  آستانه،  $inImage$  تصویر ورودی و  $outImage$  تصویر خروجی است.

استفاده از روش ساده‌ی عنوان‌شده باعث می‌شود که تشخیص پلاک در هنگام تغییر روشنایی

تصویر و یا افتادن سایه بر روی پلاک، در عمل امکان‌پذیر نباشد. برای حل این مشکل از آستانه‌گیری

تطبیقی استفاده می‌شود شکل زیر مقایسه‌ای از روش آستانه‌گیری با حد آستانه‌ی ثابت و آستانه‌گیری

تطبیقی را نشان می‌دهد:



شکل ۲-۵ اثر آستانه‌گیری تطبیقی [۱۶]

در شکل بالا قسمت (الف) تصویر اصلی یک پلاک که بر روی آن سایه افتاده است را نشان می‌دهد

و قسمت‌های (ب)، (ج) و (د) به ترتیب، تصویر حاصل از آستانه‌گیری با حد بالا، تصویر آستانه‌گیری با

حد پایین و آستانه‌گیری تطبیقی را نشان می‌دهند..

## ۲-۶ جداسازی اعداد و حروف

دومین بخش در فرایند خواندن پلاک، جداسازی نویسه‌ها است. پس از پیدا کردن محل پلاک

باید نویسه‌ها از تصویر پلاک استخراج شوند تا در مرحله تشخیص نویسه‌ها بتوانیم نویسه‌ها را شناسایی

کنیم. دو روش متداول که در اکثر مقالات به آن‌ها اشاره شده است در ۲-۶-۱ و ۲-۶-۲ بیان شده‌اند.

## ۲-۶-۱ واکاوی اجزای متصل

در این روش پس از پیدا کردن محل پلاک و جداسازی آن و انجام پردازش‌هایی مانند لبه‌یابی،

آستانه‌گذاری و عملیات مرفولوژی بر روی آن، با برچسب زنی قسمت‌های متصل بر اساس همسایگی‌های

۸ تایی و ۴ تایی و قطری می‌توان حروف و اعداد درون پلاک را استخراج کرد. شکل ۲-۶ یک نمونه

از فرایند جداسازی نویسه نشان می‌دهد. در این بخش پس از به‌دست آوردن تصویر دوسطحی<sup>۱</sup> پلاک

الگوریتم واکاوی اجزای متصل را به آن اعمال می‌کنیم. سپس با برچسب‌زنی هر قسمت در هر مرحله

نویسه‌ها پیدا می‌شود.



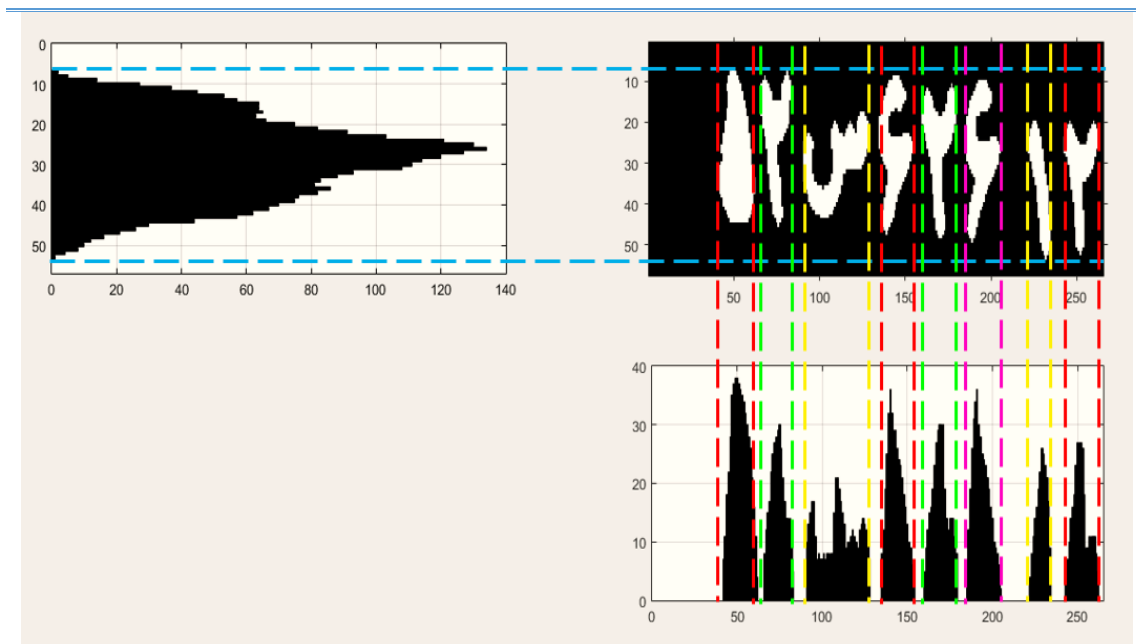
شکل ۲-۶ جداسازی اعداد و حروف با استفاده از روش واکاوی اجزای متصل (CCA). الف) تصویر پلاک، ب) تصویر

دوسطحی، ج) جداسازی نویسه‌ها.

<sup>1</sup> Two-Level Image or Binary image

## ۲-۶-۲ افکنش افقی و عمودی

در این روش بر اساس جستجوی عمودی و افقی تصویر و شمارش تعداد یک‌های موجود در سطرها و ستون‌های تصویر باینری و از تلاقی این دو می‌توان اعداد و حروف را استخراج کرد که همه این‌ها در شکل ۷-۲ آمده است. همان‌طور که در شکل ۷-۲ مشهود است. مزیت این روش نسبت به روش ۱-۶-۲ این است که هر یک از اعداد و حروف در مکان اصلی خود قرار گرفته‌اند و این باعث می‌شود که فرایند شناسایی و نمایش حروف راحت‌تر باشد. اما در روش واکاوی اجزای متصل ممکن است بعضی اعداد و حروف به ترتیب قرارگیری آن‌ها در پلاک شناسایی نشوند. برای حل این مشکل باید از اطلاعات مختصات آن‌ها در تصویر اصلی استفاده کرد [۱۷].



شکل ۷-۲ جداسازی حروف بر اساس روش هیستوگرام

## ۷-۲ برخی از راهکارهای تشخیص نویسه‌های پلاک

برای تشخیص نویسه در سامانه<sup>۱</sup> LPR، می‌توان از الگوریتم مبتنی بر تطبیق الگو و یادگیری با استفاده از شبکه عصبی کمک گرفت. داده‌های به‌دست‌آمده از این روش‌ها نشان می‌دهد که با افزایش داده‌های ورودی شبکه عصبی می‌توان پاسخ بسیار مناسبی را دریافت نمود.

در [۱۸] برای تشخیص نویسه‌های فارسی و در [۱۹] برای تشخیص نویسه‌های لاتین از شبکه عصبی چندلایه پرسپترون<sup>۲</sup> MLP بهره برده‌اند. در این شیوه برای آموزش شبکه عصبی از الگوریتم پس‌انتشار خطا<sup>۳</sup> استفاده شده است. همچنین تعداد نرون‌های لایه مخفی با آزمایش و خطا در مرحله یادگیری تعیین می‌شوند. در [۲۰] به علت وجود حروفی از ترکیب دو زبان مختلف چینی و انگلیسی به‌طور هم‌زمان از چند شبکه عصبی که هر یک وظیفه تشخیص حروف لاتین و چینی را برعهده گرفته‌اند استفاده شده است. در روش بعدی برای تشخیص نویسه‌ها از شبکه عصبی به شیوه دیگری بهره برده است [۲۱]. راهکار ارائه‌شده در این روش، استفاده از دو نوع شبکه پس‌انتشار خطای سه لایه با تابع انتقال تانژانت سیگموئید و شبکه کوانتیزاسیون بردار یادگیری<sup>۴</sup> برای تشخیص نویسه‌ها است. معیار مقایسه بین این دو روش در این مقاله بر مبنای میزان دقت بازشناسی نویسه‌ها عنوان شده است. راهکار ارائه‌شده بعدی استفاده از شبکه عصبی احتمالی<sup>۵</sup> PNN، است. در این روش از دو شبکه احتمالی، یکی

---

<sup>1</sup> License Plate Recognition

<sup>2</sup> Multi Layer Perceptron

<sup>۳</sup> Backpropagation Algorithm

<sup>4</sup> Learning Vector Quantization

<sup>5</sup> Probabilistic Neural Network

برای تشخیص حروف و دیگری برای تشخیص اعداد استفاده شده است [۲۲]. در [۲۳] به همین شیوه رفتار شده است با این تفاوت که این بار از یک شبکه عصبی برای تشخیص حروف و اعداد استفاده شده است. مرحله آموزش در PNN بسیار سریع‌تر از سایر شبکه‌های عصبی است؛ اما نیاز به حافظه بیشتر و زمان اجرای کندتری نسبت به سایر شبکه‌های عصبی متداول دارند. مقاله [۲۴] از ماشین بردار پشتیبان، SVM<sup>۱</sup>، برای تشخیص نویسه‌ها استفاده کرده است. در راهکار ارائه‌شده از چهار طبقه بند مبتنی بر SVM برای آشکارسازی نویسه‌ها استفاده شده است. در هرکدام از این طبقه بندها به ازای تعداد کلاس‌های آن طبقه بند، SVM در نظر گرفته شده است.

روش‌های ذکرشده همگی بر روی پایگاه داده‌ای استاندارد از نویسه‌ها آموزش می‌بینند؛ به عبارت دیگر به دنبال راهکاری هستند که قادر باشد؛ برای کل ورودی‌ها در فضای مسئله جواب قابل قبولی داشته باشد.

ترکیب طبقه‌بندها رویکردی است جهت بهبود عملکرد طبقه‌بندی خصوصاً برای مسائل پیچیده‌ای از قبیل مسائلی که تعداد الگوها زیاد است یا بُعد ویژگی بزرگ است یا کلاس‌ها همپوشانی دارند. در روش ترکیب طبقه‌بندها، ثابت شده است که قاعده تقسیم و غلبه در بسیاری از این وضعیت‌های پیچیده مؤثر است. این ترکیب می‌تواند در دو ساختار پویا<sup>۲</sup> و ایستا<sup>۳</sup> انجام گیرد [۲۵]. در حالت دینامیک،

---

<sup>۱</sup> Support Vector Machine

<sup>۲</sup> dynamic

<sup>۳</sup> ecstatic

سیگنال ورودی طبقه‌بندها به‌طور مستقیم در سازوکار ترکیب خروجی‌ها دخالت دارند. اختلاط خبره‌ها یکی از شناخته‌شده‌ترین روش‌ها است که بر اساس قاعده تقسیم و غلبه پایه‌گذاری شده است [۲۶]. در این روش فضای مسئله بین تعدادی از شبکه‌های عصبی با نظارت یک شبکه میانجی تقسیم می‌شود.

در این فصل به برخی از روش‌های پرکاربرد که بر اساس اطلاعات لبه، بافت تصویر و اطلاعات رنگی تصویر پیاده‌سازی شده‌اند اشاره کردیم. همچنین به راهکارهای ارائه‌شده در زمینه انجام مراحل مختلف تشخیص خودکار پلاک خودرو از جمله تشخیص محل پلاک، جداسازی نویسه‌ها و تشخیص نویسه‌ها اشاره کردیم.

در فصل بعد به روش‌های استفاده شده در این تحقیق برای به‌دست آوردن محل پلاک، جداسازی نویسه‌ها و تشخیص آن‌ها می‌پردازیم. همچنین برخی از نتایج به‌دست آمده در مراحل مختلف را نشان می‌دهیم.

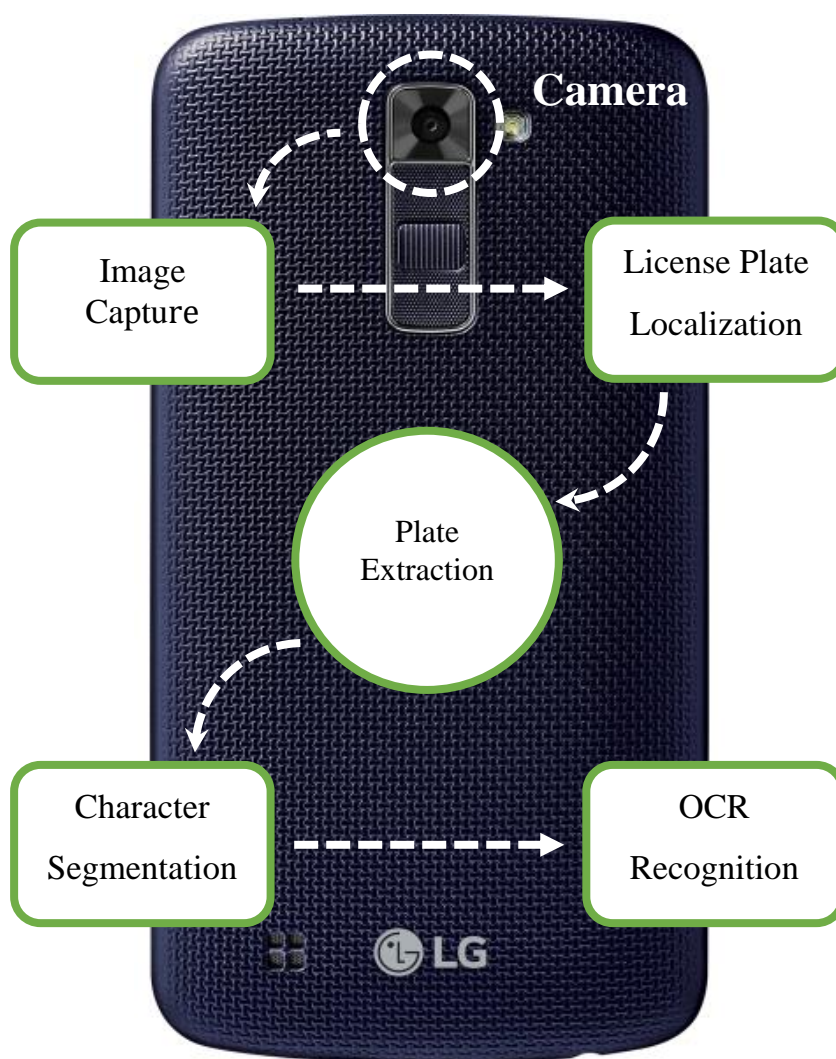




# فصل سوم

## روش پیشنهادی

به طور معمول یک سامانه تشخیص پلاک خودرو از چهار بخش اصلی تشکیل شده است. این بخش‌ها عبارت‌اند از تهیه عکس، آشکارساز پلاک، آشکارساز نویسه‌ها و تشخیص نویسه‌ها. تمامی این قسمت‌ها با اهمیت هستند؛ اما تعیین محل پلاک خودرو و پیدا کردن نویسه‌ها از اهمیت بسزایی برخوردار است؛ زیرا در صورت نقص هر کدام از این مراحل، بخش‌های بعدی تحت تأثیر قرار می‌گیرند. شکل ۱-۳ ساختار کلی یک سیستم تشخیص خودکار پلاک خودرو را نشان می‌دهد.



شکل ۱-۳ ساختار کلی یک سیستم تشخیص خودکار پلاک خودرو

در الگوریتم پیشنهادی پس از ثبت عکس توسط دوربین و انجام پیش‌پردازش‌های لازم بر روی

عکس، ابتدا محل پلاک را پیدا کرده و سپس نویسه‌ها را جدا می‌کنیم و در آخر نویسه‌ها را تشخیص می‌دهیم.

### ۱-۳ حذف نویز

ابتدا تصویر را تحت فیلتر میانه<sup>۱</sup> قرار می‌دهیم تا در صورت وجود نویزهای تصادفی در تصویر، آن‌ها را برطرف سازیم. این فیلتر بجای مقدار یک پیکسل، میانه سطوح همسایگی آن پیکسل را قرار می‌دهد. فیلتر میانه از نوع مرتبه آماری<sup>۲</sup> است. فیلترهای مرتبه آماری، فیلترهای مکانی غیرخطی‌اند که پاسخ آن‌ها مبتنی بر رتبه‌بندی پیکسل‌های موجود در ناحیه‌ای از تصویر است که در حیطه فیلتر قرار می‌گیرند و سپس مقدار پیکسل مرکزی با مقدار تعیین‌شده توسط نتیجه رتبه‌بندی، جایگزین می‌شوند [۲۷]. این فیلتر نسبت به فیلترهای خطی هم‌اندازه تصویر را کمتر مات می‌کند. رابطه فیلتر میانه در ۲-۳ آمده است.

$$\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s, t)\} \quad (2-3)$$

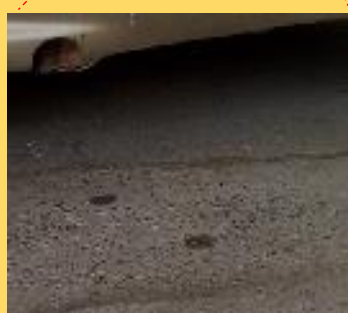
در اینجا  $x$  و  $y$  مختصات پیکسل تصویر،  $s$  و  $t$  مختصات پیکسل داخل پنجره فیلتر،  $S_{xy}$  پنجره

در نظر گرفته شده برای فیلتر است.

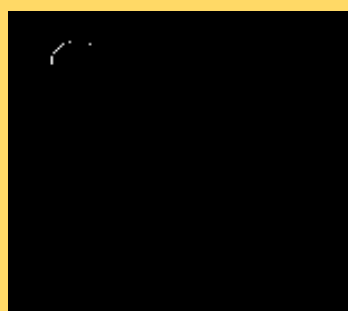
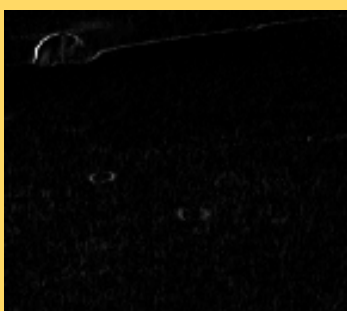
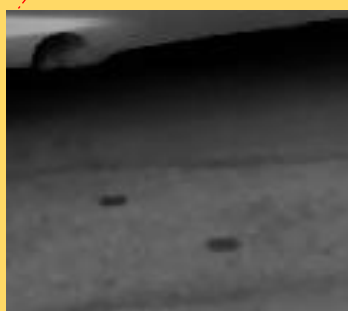
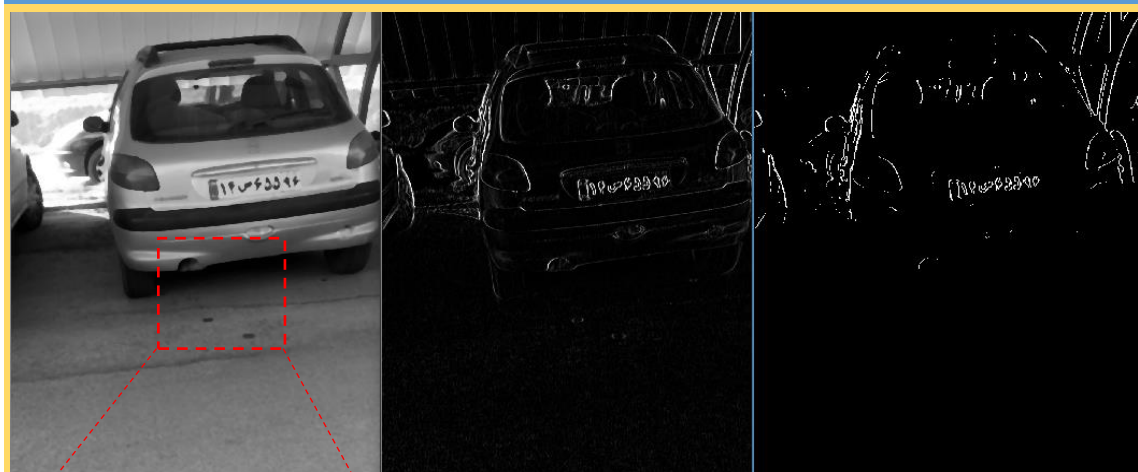
---

<sup>1</sup> Median Filter

<sup>2</sup> Order Statistic Filter



کل ۲-۳ تصاویر به دست آمده قبل از اعمال فیلتر میانه



کل ۳-۳ تصاویر به دست آمده بعد از اعمال فیلتر میانه

شکل ۲-۳ با فرض این که فیلتر میانه به تصویر اعمال نشده باشد را نشان می‌دهد. در این تصویر ریگ‌های موجود در کف پارکینگ و لبه‌های ایجادشده توسط آن‌ها نویز فلفل نمکی را در ذهن تداعی می‌کند که باعث شد از فیلتر میانه استفاده کنیم. شکل ۳-۳ اعمال فیلتر میانه و تأثیر عملکرد آن بر بهبود تصویر حاصل از لبه یابی در مرحله بعد را نشان می‌دهد.

### ۲-۳ لبه یاب سوبل<sup>۱</sup>

وجود لبه‌های عمودی زیاد ناشی از نویسه‌های پلاک خودرو یک ویژگی منحصربه‌فرد برای تعیین محل پلاک است. از این خصوصیت می‌توان برای یافتن مکان دقیق پلاک استفاده کرد. راهکارها و الگوریتم‌های متفاوتی برای پیدا کردن لبه در پردازش تصویر مطرح است. در این بین عملگر سوبل به دلیل سرعت بالا و حجم پردازش کم در مقایسه با دیگر روش‌ها بیشتر استفاده می‌شود. لبه یاب سوبل در واقع گرادینان یک تصویر را محاسبه می‌کند. عملگر سوبل در حقیقت یک عملگر دیفرانسیلی گسسته است که از طریق کانوالو آن با تصویر اصلی و محاسبه مشتق مرتبه اول تصویر می‌توان لبه‌های موجود در تصویر که حاصل اختلاف شدت روشنایی است را محاسبه نمود. لبه یابی به روش سوبل به ترتیب دارای دو ماسک تشخیص لبه افقی<sup>۲</sup> و عمودی<sup>۳</sup> طبق رابطه ۳-۳ و ۴-۳ است.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3-3)$$

<sup>1</sup> Sobel Edge Detection

<sup>2</sup> Horizontal Edge

<sup>3</sup> Vertical Edge

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4-3)$$

در روش پیشنهادی چون اعداد و حروف دارای لبه‌های عمودی زیادی می‌باشند فقط از ماسک لبه‌یاب عمودی استفاده می‌شود. با این فرض که تصویر ورودی، چرخش بیش‌ازحد نداشته باشد. تصویر حاصل از فرایند لبه‌یابی یک تصویر سطح خاکستری است که پیکسل‌های مربوط به لبه‌ها دارای بیشترین مقدار و سایر نقاط دارای کمترین مقدار است. پس از این مرحله با اعمال یک آستانه به تصویر، لبه‌ها را برجسته می‌کنیم تا در مرحله بعد مورد استفاده قرار دهیم. شکل ۴-۳ نتیجه عمل لبه‌یابی و آستانه‌گذاری را نشان می‌دهد.



تصویر سطح خاکستری

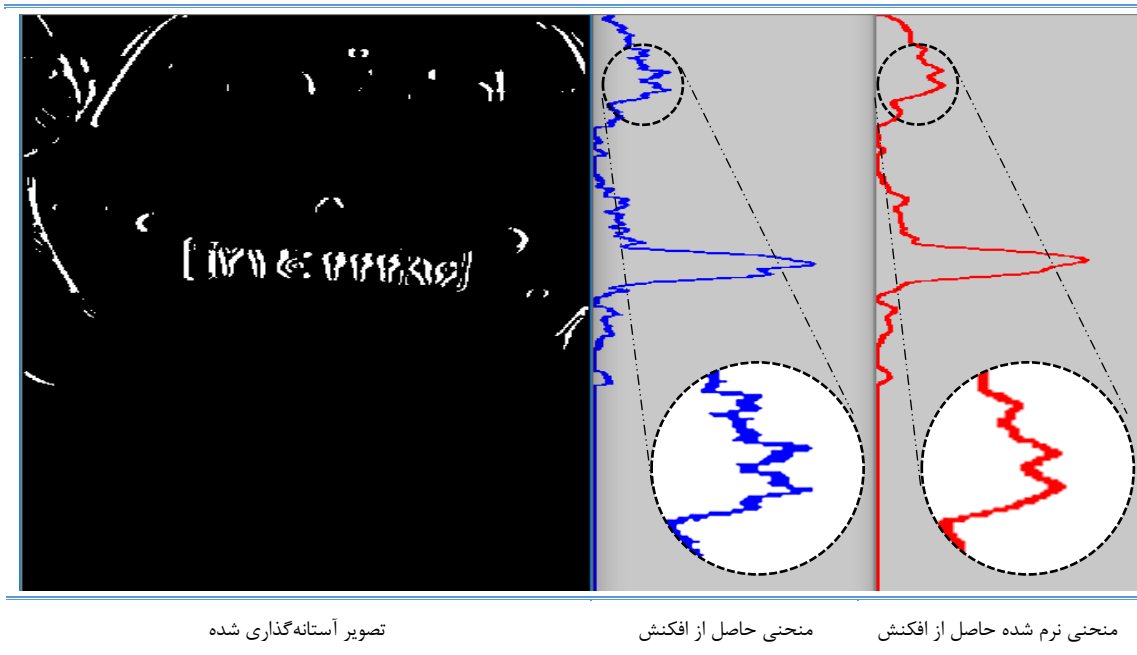
تصویر حاصل از لبه‌یاب عمودی

تصویر آستانه‌گذاری شده

شکل ۴-۳ تأثیرات لبه‌یاب عمودی و آستانه‌گیری از تصویر لبه‌یابی شده

### ۳-۳ بکار بردن افکنش<sup>۱</sup>

بعد از عملیات لبه‌یابی و آستانه‌گذاری نوبت به تعیین محل احتمالی پلاک می‌رسد. همان‌طور که در مرحله قبل توضیح دادیم نویسه‌های پلاک دارای بیشترین لبه‌های عمودی می‌باشند؛ بنابراین اگر بتوانیم در تصویر باینری حاصل از مرحله قبل به شمارش پیکسل‌های سفید پردازیم می‌توانیم محل احتمالی پلاک را مشخص کنیم. اما جهت شمارش پیکسل‌ها بسیار مهم است. در روش افکنش باید پیکسل‌های سفید یا برتر را در جهت افقی، بررسی و شمارش کنیم که حاصل این شمارش پس از جستجوی تمامی سطریهای یک منحنی با قله‌ها و دره‌های بسیار تیز خواهد بود که مطلوب ما نیست و احتمال خطا در مرحله بعد را افزایش خواهد داد. از این‌رو با عبور این منحنی از یک فیلتر پایین‌گذر می‌توان نقاط تیز آن را نرم کرد. در شکل ۳-۵ تصویر حاصل از اعمال افکنش افقی را مشاهده می‌کنیم.



شکل ۳-۵ منحنی حاصل روش افکنش افقی

<sup>۱</sup> projection

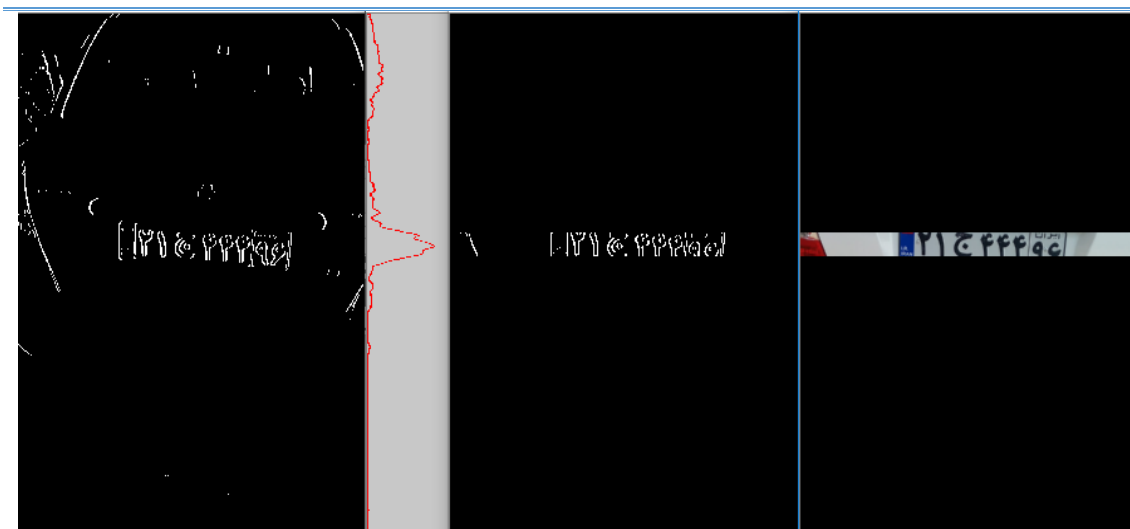


پس از انجام این روش و به دست آوردن نمودار حاصل از افکنش با کسب بیشینه مقدار شمارش

لبه‌ها و با فرض این که آن مکان پلاک است؛ با اعمال یک آستانه که معمولاً تجربی به دست می‌آید

می‌توان مکان احتمال پلاک را به دست آورد. آستانه در اینجا  $0/6$  مقدار بیشینه در نظر گرفته شده است.

شکل ۳-۶ خروجی این مرحله را نشان می‌دهد.



شکل ۳-۶ منحنی حاصل روش افکنش افقی

در برخی روش‌ها، افکنش را در جهت عمود نیز اعمال می‌کنند و از برآیند افکنش عمودی و افقی

محل پلاک را تعیین می‌کنند؛ اما این ایده به خاطر تمرکز آن بر روی ویژگی لبه، همیشه کارساز نخواهد

بود. از آن گذشته چه روش افکنش افقی و چه ترکیب آن با افکنش عمودی به دلیل حساسیت به زاویه‌دار

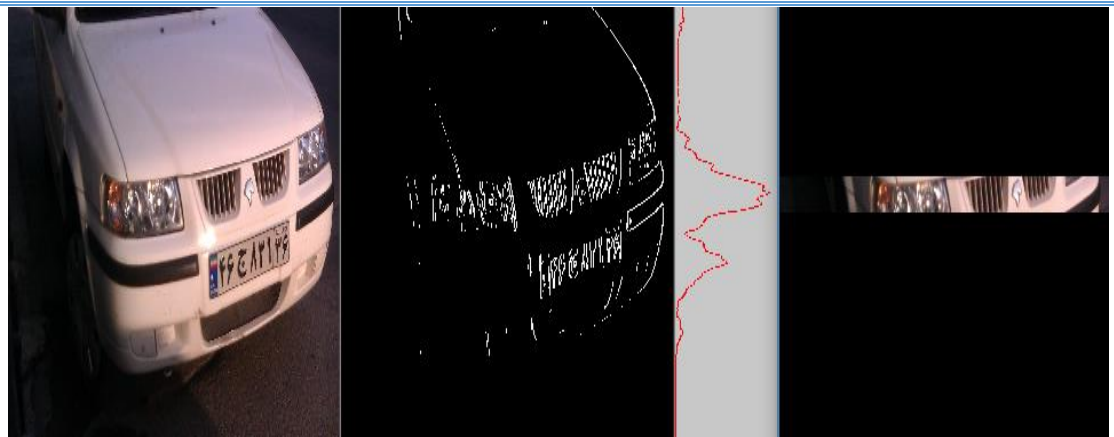
بودن پلاک روش‌های چندان مناسبی نیست؛ اما از سرعت خوبی برخوردارند و در زمانی که زاویه دوربین

به خودرو صحیح باشد روش قابل قبولی است. شکل ۳-۷ و ۳-۸ خطای حاصل از زاویه‌دار بودن پلاک

را نشان می‌دهد.



شکل ۷-۳ تأثیر اعمال افکنش به تصویری که پلاک در آن نسبت به افق زاویه ندارد



شکل ۸-۳ تأثیر اعمال افکنش به تصویری که پلاک در آن نسبت به افق زاویه دارد

### ۴-۳ اعمال عملگر مورفولوژی<sup>۱</sup>

عملگرهای مورفولوژی، عملگرهایی هستند که تصاویر را بر مبنای عنصر ساختاری که برای آن‌ها تعریف

شده است پردازش می‌کنند. از ساده‌ترین عملگرهای مورفولوژی عملگرهای فرسایش<sup>۲</sup> و گسترش<sup>۳</sup>

هستند که در این قسمت برای اتصال لبه‌های عمودی پلاک به یکدیگر برای تعیین محل پلاک از عملگر

گسترش استفاده می‌کنیم.

<sup>1</sup> Morphology

<sup>2</sup> Erosion

<sup>3</sup> Dilation

همان طور که از نام عملگر پیداست، این عملگر باعث گسترش نقاط سفید در تصویر می‌شود. معمولاً به پنجره مورد استفاده در این روش عنصر ساختاری می‌گویند که مقادیر عنصر ساختاری، می‌تواند ۱ یا صفر باشد. المان ساختاری در این مرحله دارای شکلی مستطیلی است و تمامی مقادیر آن ۱ است. گسترش تصویر I با عنصر ساختاری W در مجموعه‌ی Z طبق رابطه ۵-۳ تعریف می‌شود:

$$I \oplus W = \{Z | reflection(W) \cap I \neq NULL\} \quad (5-3)$$

این معادله بر اساس بازتاب دادن W حول مبدأش و انتقال دادن ای بازتاب با z است؛ به عبارت دیگر گسترش I با عنصر ساختاری W بدین معنی است که اگر عنصر ساختاری W را بر روی پیکسل‌های I حرکت دهیم و در هر بار حرکت اشتراک عنصر ساختاری با محدوده زیر عنصر ساختاری در تصویر I تهی نباشد، مقدار پیکسل مرکزی که عنصر ساختاری بر روی آن قرار گرفته است، برابر ۱ خواهد شد. شکل ۹-۳ حاصل استفاده از این روش است.



شکل ۹-۳ تأثیر اعمال عملگر مورفولوژی گسترش بر تصویر

### ۵-۳ واکاوی اجزای متصل<sup>۱</sup>

پس از گسترش لبه‌های عمودی پلاک، با استفاده از واکاوی اجزای متصل به جستجوی نامزدهای احتمالی پلاک می‌پردازیم و با تعیین یک سری آستانه‌ی عرضی و طولی، از انتخاب نامزدهای غیرمحمتمل جلوگیری و نامزدهای محتمل را برای مرحله بعدی انتخاب می‌نماییم. حذف نمونه‌های غیرمحمتمل سرعت پردازش مراحل بعدی را افزایش خواهد داد. انتخاب آستانه عرضی و طولی بر اساس نسبت طول به عرض پلاک در نظر گرفته می‌شود. این نسبت  $\frac{4}{5}$  به  $1$  است یعنی طول پلاک  $\frac{4}{5}$  برابر عرض آن در نظر گرفته شده است. شکل ۳-۱۰ برخی از نامزدهایی که به عنوان محل پلاک در نظر گرفته شده‌اند را نشان می‌دهد. این نامزدها در مراحل بعدی بررسی می‌شوند، اگر وجود نویسه‌ها در آن‌ها تایید شود به عنوان پلاک انتخاب می‌شوند.

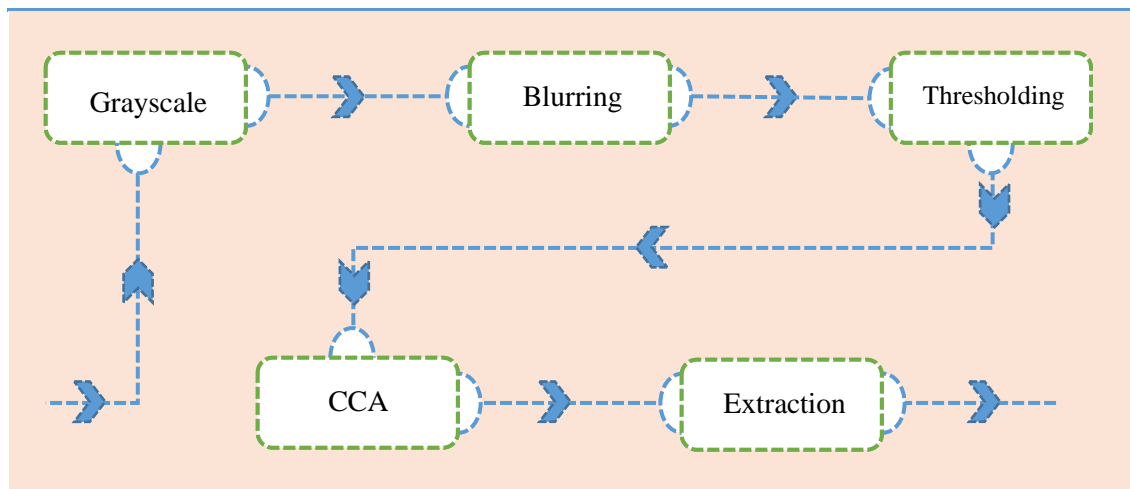


شکل ۳-۱۰ نامزدهای درست و نادرست محل پلاک

<sup>1</sup> Connected Components Analysis (CCA)

### ۳-۶ تفکیک نویسه‌ها

در قسمت قبل همراه نامزدهای جداسازی پلاک، توسط الگوریتم واکاوی اجزای متصل مختصات مکانی آنها نیز ثبت می‌شود. حال از روی مختصات ثبت شده در مرحله ۳-۵، نامزدهای محتمل از تصویر اصلی برش خواهند خورد. سپس مطابق شکل ۳-۱۱ مراحل ذکر شده اعمال خواهد شد.



شکل ۳-۱۱ مراحل روش تفکیک نویسه‌ها

ابتدا تصویر رنگی پلاک استخراج شده را به تصویر سطح خاکستری تبدیل می‌کنیم. سپس با اعمال یک فیلتر میانگین‌گیر تصویر به دست آمده را مات می‌کنیم تا نویز احتمالی تصویر برطرف شود. حال طبق الگوریتم ۳-۱ یک آستانه‌گذاری افقی به تصویر مات شده اعمال می‌کنیم. نتیجه یک تصویر

---

---

الگوریتم ۳-۱: آستانه‌گذاری افقی

---

---

اگر  $\text{Blur}(x, y) \times 0.95$  بزرگ‌تر از  $\text{Gray}(x, y)$  باشد

مقدار  $\text{Out}(x, y)$  را برابر 1.0 قرار بده

در غیر این صورت

مقدار  $\text{Out}(x, y)$  را برابر 0.0 قرار بده

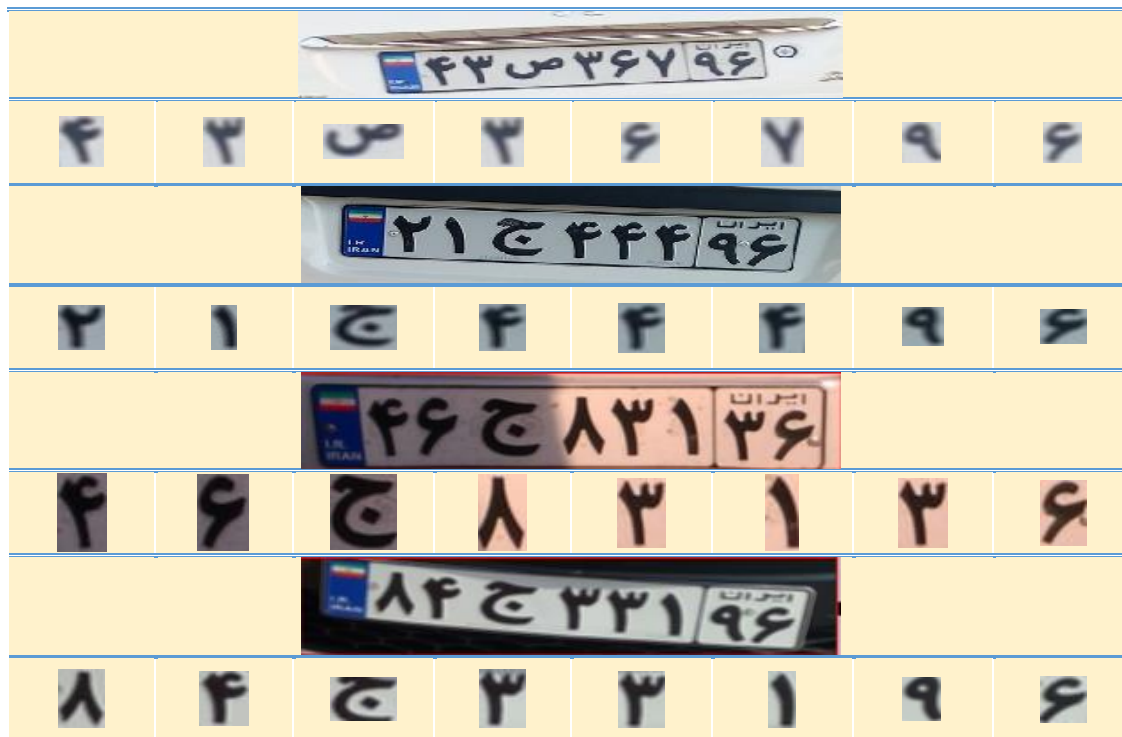
---

---

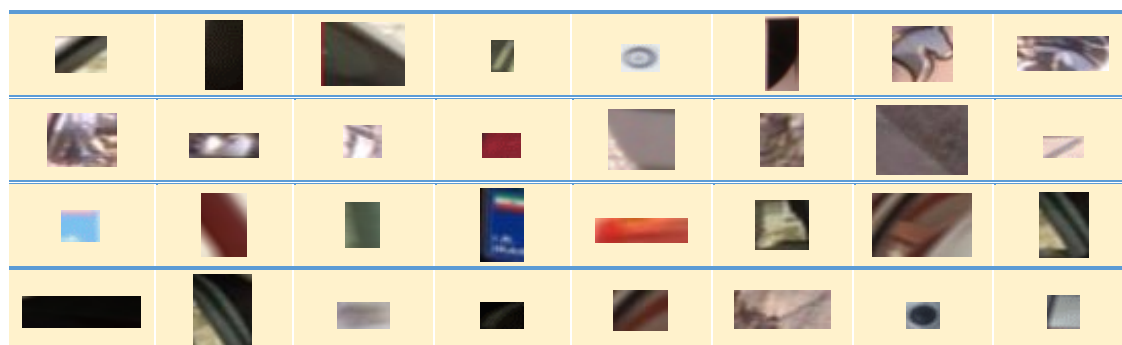
باینری خواهد بود. در مرحله بعدی با استفاده از الگوریتم CCA و آستانه‌گذاری طولی و عرضی بر اساس

اندازه نویسه‌های پلاک، نامزدهای احتمالی را برای مرحله تشخیص نویسه‌ها انتخاب می‌کنیم. شکل ۳-۳-

۱۲ نویسه‌هایی که به درستی استخراج شده و ۳-۱۳ نمونه‌های اشتباه را نشان می‌دهد.



شکل ۳-۱۲ نویسه‌هایی که به درستی استخراج شده



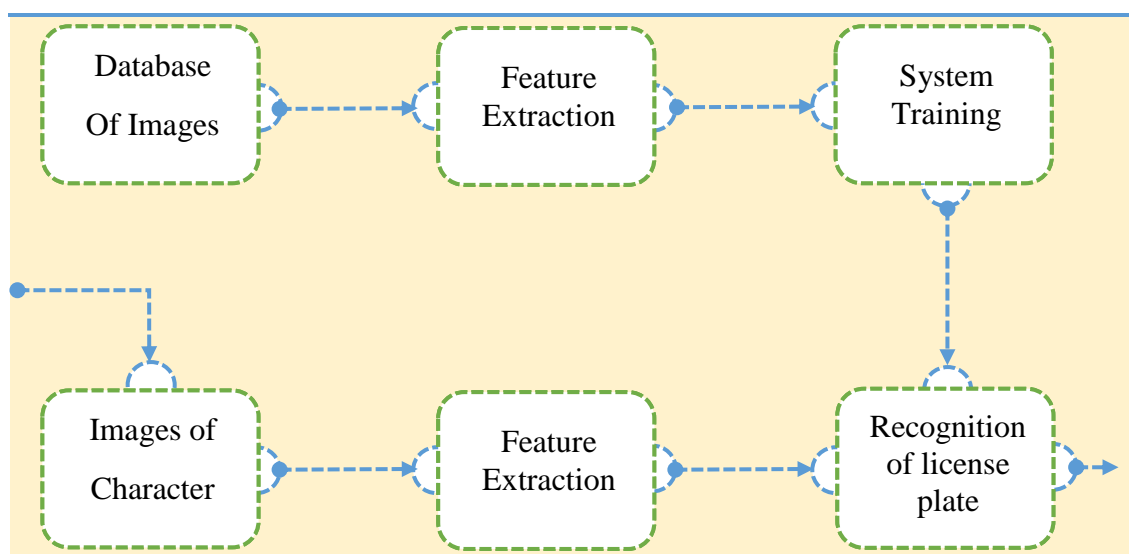
شکل ۳-۱۳ نمونه‌هایی که به اشتباه به جای نویسه‌ها استخراج شده است

### ۷-۳ تشخیص نویسه‌های پلاک

این مرحله جز یکی از مهم‌ترین قسمت‌های سامانه تشخیص پلاک است. بخش‌های این مرحله طبق

شکل ۱۴-۳ عبارت‌اند از:

- ❖ پایگاه داده‌ای متشکل از تصاویر حروف و اعداد تحت شرایطی خاص (مانند تصاویر زاویه‌دار، نویزی، برش خورده و با اشکال متنوع)
- ❖ استخراج ویژگی از تصاویر پایگاه داده
- ❖ آموزش سامانه تشخیص پلاک بر اساس ویژگی‌های استخراج‌شده،
- ❖ آزمایش سامانه تشخیص پلاک بر اساس تصاویر پلاک ورودی به سیستم



شکل ۱۴-۳ مراحل تشخیص نویسه‌های پلاک

### ۱-۷-۳ پایگاه داده

این بخش شامل حدود ۳۰۰۰۰ نمونه از تصاویر اعداد و حروف است که تحت شرایط گوناگون

تهیه شده‌اند. این نمونه‌ها در ۳۰ کلاس مختلف شامل نمونه‌های نامعتبر، ۱۸ حرف از حروف الفبا فارسی،

۹ رقم از اعداد ۱ تا ۹ و علامت معلولان است. در جدول ۱-۳ برخی از نمونه‌های آموزش آورده شده‌اند.

جدول ۱-۳ برخی از نمونه‌های آموزش

| مثال‌ها |    | تعداد | کلاس | مثال‌ها |     | تعداد | کلاس | مثال‌ها |   | تعداد | کلاس    |
|---------|----|-------|------|---------|-----|-------|------|---------|---|-------|---------|
| ق       | اق | Gh    | ۲۰   | الف     | الف | A     | ۱۰   |         | * |       | نامعتبر |
| ک       | ک  | k     | ۲۱   | ب       | ب   | B     | ۱۱   |         | 1 | ۱     |         |
| ل       | ل  | L     | ۲۲   | پ       | پ   | P     | ۱۲   |         | 2 | ۲     |         |
| م       | م  | M     | ۲۳   | ت       | ت   | T     | ۱۳   |         | 3 | ۳     |         |
| ن       | ن  | N     | ۲۴   | د       | د   | D     | ۱۴   |         | 4 | ۴     |         |
| و       | و  | V     | ۲۵   | ح       | ح   | J     | ۱۵   |         | 5 | ۵     |         |
| ه       | ه  | H     | ۲۶   | س       | س   | C     | ۱۶   |         | 6 | ۶     |         |
| ی       | ی  | E     | ۲۷   | ص       | ص   | S     | ۱۷   |         | 7 | ۷     |         |
| یر      | یر | Y     | ۲۸   | ط       | ط   | Ta    | ۱۸   |         | 8 | ۸     |         |
| ی       | ی  | Ch    | ۲۹   | ع       | ع   | Aa    | ۱۹   |         | 9 | ۹     |         |

در جدول ۱-۳ حرف ت مخصوص پلاک تاکسی، حرف ع خودروهای عمومی، حرف پ خودرو پلیس،

حرف الف خودرو اداری، حرف ک خودرو کشاورزی و علامت مخصوص معلولان است. پلاک

خودروهای ایران شامل ۷ رقم و ۱ حرف هست. نسبت طول به عرض پلاک نیز تقریباً ۴/۵ به ۱ است.

نمونه‌های نامعتبر که در جدول ۱-۳ به آن‌ها اشاره شد، شامل اشکال یا حروفی است که در

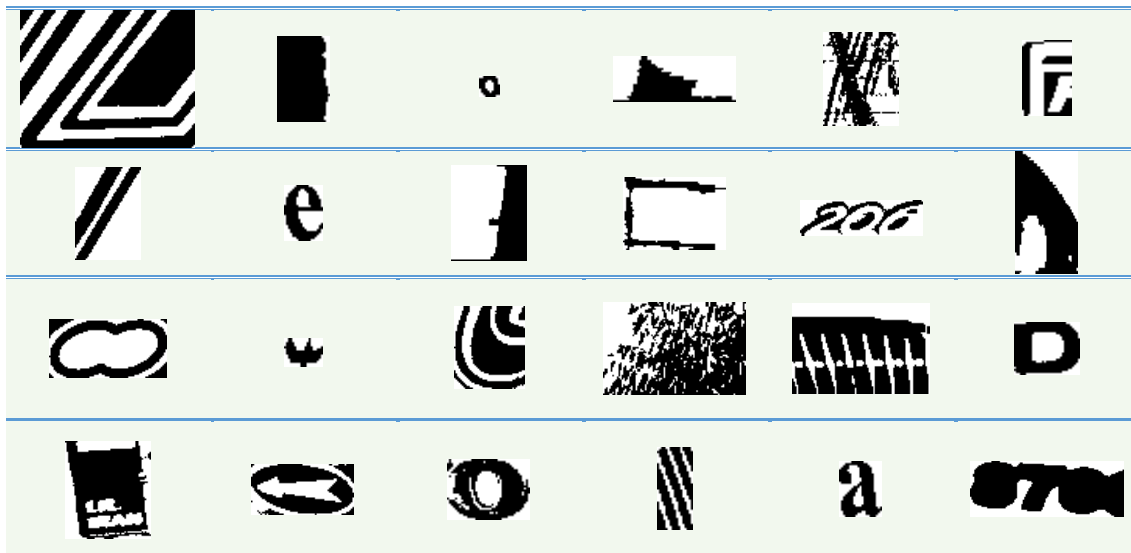
مجموعه‌ی نویسه‌های تعریف‌شده برای پلاک نمی‌باشند و تعریف این کلاس بدین خاطر است که اگر

نمونه نامعتبری در مرحله تشخیص حروف و اعداد به شبکه عصبی داده شود، به اشتباه به عنوان یکی



از حروف یا اعداد پلاک در نظر گرفته نشود. شکل ۳-۱۵ برخی از نمونه‌های نامعتبری که در

پایگاه داده حروف و اعداد این مجموعه قرار داده‌ایم را نشان می‌دهد.



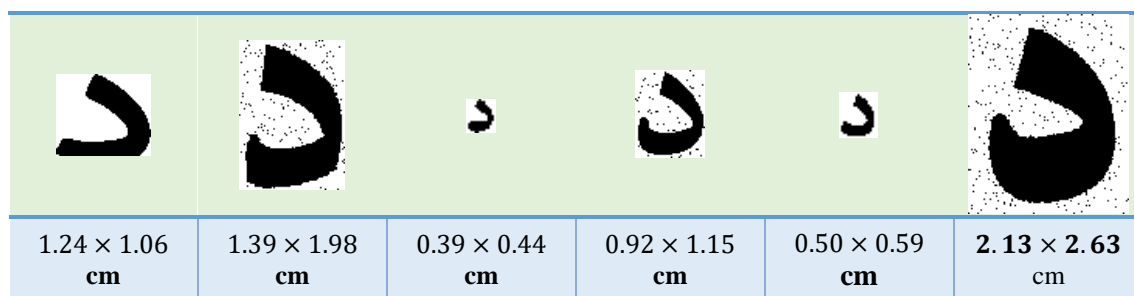
شکل ۳-۱۵ برخی از نمونه‌های نامعتبری

### ۳-۷-۲ استخراج ویژگی

اگر بخواهیم یک شبکه عصبی را آموزش دهیم باید برای هر نویسه یک ویژگی منحصر به فرد ایجاد

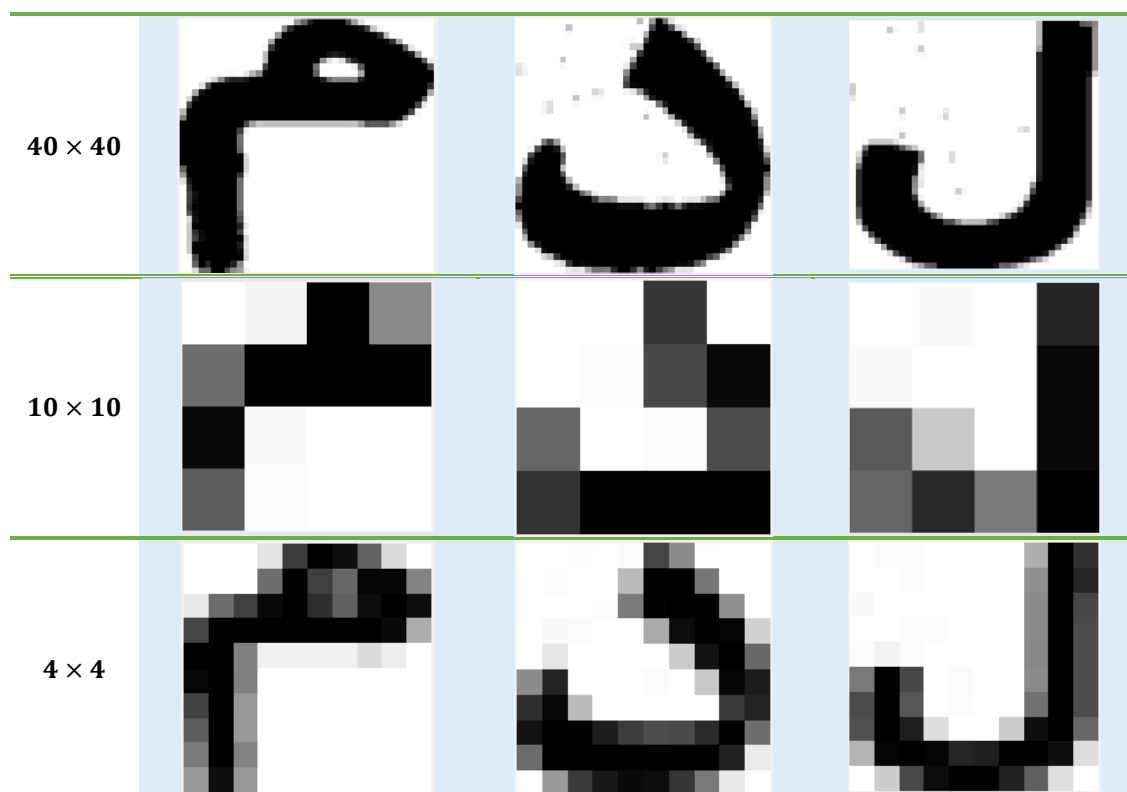
کنیم تا بتوانیم به راحتی شبکه عصبی را آموزش دهیم. بهترین راه حل تولید یک بردار ویژگی برای هر

نویسه است. همان طور که در شکل ۳-۱۶ مشاهده می‌کنید تصاویری که از هر نویسه در پایگاه



شکل ۳-۱۶ برخی از تصاویر پایگاه داده حروف و اعداد فارسی

داده وجود دارد، در اندازه‌های متفاوتی هستند. بنابراین در صورت استخراج ویژگی از این تصاویر، طول بردارهای ویژگی به دست آمده برای هر نویسه متفاوت خواهد بود. برای حل این مشکل اندازه تمامی تصاویر پایگاه داده را یکسان می‌کنیم. برای این منظور، اندازه هر تصویر  $40 \times 40$  در نظر گرفته شده است. روش اول برای استخراج ویژگی این است که تصویر را به  $m \times n$  بلوک تقسیم کنیم و متوسط مقدار پیکسل‌های هر بلوک را به عنوان یک ویژگی در نظر بگیریم. در نهایت  $m \times n$  ویژگی داریم. این شیوه به شدت روشنایی تصویر حساس است. شکل ۳-۱۷ تصویر حرف "ل"، "د" و "م" و ویژگی استخراج شده‌ی آن‌ها را برای بلوک  $4 \times 4$  و  $10 \times 10$  نشان می‌دهد؛ در این مورد، به ترتیب ۱۶ و ۱۰۰ ویژگی خواهیم



شکل ۳-۱۷ تصویر حرف "ل"، "د" و "م" و ویژگی‌های استخراج شده، توسط بلوک  $4 \times 4$  و  $10 \times 10$

داشت. ویژگی‌های به دست آمده باید قبل از اعمال به شبکه عصبی نرمالیزه شوند. روش دوم استفاده از هیستوگرام گرادیان جهت‌دار<sup>۱</sup> است. این روش حساس به تغییرات شدت روشنایی نیست. در این روش نیز ابتدا تصویر نویسه‌ها را به اندازه  $40 \times 40$  تبدیل و سپس آن را به  $16$  بلوک  $10 \times 10$  تقسیم می‌کنیم. از هر بلوک  $16$  تایی،  $16$  ویژگی استخراج خواهد در ادامه به توضیح استخراج ویژگی در این روش می‌پردازیم.

در این الگوریتم در ازای هر نقطه از تصویر  $40 \times 40$  یک زاویه و اندازه حاصل می‌گردد. طبق

رابطه  $6-3$  و  $7-3$  ابتدا کرنل سوبل عمودی و افقی به تصویر اعمال می‌شود.

$$G_x = I * S_x \quad (6-3)$$

$$G_y = I * S_y \quad (7-3)$$

در رابطه  $6-3$  و  $7-3$  علامت  $*$  عمل کانولوشن<sup>۲</sup>،  $G_x$  و  $G_y$  گرادیان تصویر،  $S_x$  و  $S_y$  کرنل سوبل

در راستای  $x, y$  و تصویر مورد نظر با  $I$  نشان داده شده است. سپس اندازه و جهت گرادیان توسط روابط

$8-3$  و  $9-3$  حاصل می‌شود.

$$|G(i, j)| = \sqrt{G_x(i, j)^2 + G_y(i, j)^2} \quad (8-3)$$

$$\theta_G(i, j) = \tan^{-1} \left[ \frac{G_y(i, j)}{G_x(i, j)} \right] \quad (9-3)$$

<sup>1</sup> Histogram of Oriented Gradients (HOG)

<sup>2</sup> Convolution

در ۳-۸ و ۳-۹،  $i$  و  $j$  شماره سطر و ستون تصویر،  $|G(i,j)|$  اندازه گرادیان و  $\theta_G(i,j)$  جهت گرادیان است. به دلیل این که گرادیان را علامت‌دار فرض کردیم، زوایای بین  $0^\circ - 360^\circ$  را خواهیم داشت. این بازه را به ۱۶ قسمت مساوی تقسیم می‌کنیم که ۱۶ تعداد جهت‌های گرادیان یا همان بین‌های هیستوگرام را نشان می‌دهد و هر کدام از این قسمت‌ها یک کانال هیستوگرام را تشکیل می‌دهند [۲۸]. ۱۶ بلوک داریم و از هر بلوک ۱۶ ویژگی استخراج خواهد شد بنابراین تعداد ویژگی‌های استخراج‌شده ۲۵۶ ویژگی خواهد بود. ویژگی استخراج‌شده در هر کدام از بلوک‌ها توسط رابطه ۳-۱۰ به دست خواهد آمد.

$$A = |G(i,j)| \quad (۳-۱۰)$$

$$f_\theta = \sum_k A_k \quad (۳-۱۱)$$

$k$  بیانگر شماره اندازه‌ها در هر بلوک هست. بنابراین طبق رابطه ۳-۱۱ هر ویژگی استخراج‌شده در هر بلوک از مجموع اندازه‌های زاویه  $\theta$  در بلوک جاری به دست می‌آید. همان‌طور که قبلاً نیز گفتیم قبل از اعمال نمونه‌های آموزش باید ویژگی‌های استخراج‌شده نرمال شوند. برای نرمال‌سازی ویژگی‌ها، تک‌تک عناصر بردار ویژگی را بر اندازه بردار ویژگی تقسیم می‌کنیم که اندازه بردار ویژگی به ۱ نرمال شود.

### ۳-۷-۳ آموزش شبکه عصبی

حال که بردارهای ویژگی را تولید کردیم به دلیل این که نمونه‌ها با نظم خواستی چیده شده‌اند

باید نظم آن‌ها را بر هم زد تا شبکه پاسخ مناسب و خوبی را به ما بدهد. ابتدا به اندازه کل نمونه‌ها، یک بردار با اعداد تصادفی بدون تکرار تولید می‌کنیم و سپس آرایش بردارهای ویژگی را بر اساس تعداد کل بردارها به صورت تصادفی قرار می‌دهیم. شبکه عصبی استفاده‌شده از نوع چندلایه پرسپترون<sup>۱</sup> است.

در این شبکه تعداد نرون‌های ورودی به تعداد ویژگی‌ها یعنی ۲۵۶، تعداد نرون‌های لایه مخفی را به‌طور تجربی برابر ۵۰ و تعداد لایه خروجی را بر اساس تعداد کلاس‌های موجود برابر ۳۰ قرار دادیم. خروجی برای روش میانگین بلوکی برابر ۹۲/۶٪ و برای روش هیستوگرام گرادیان ۹۸/۵٪ به‌دست آوردیم. بنابراین با توجه به خروجی‌های دریافت شده روش دوم را مورد استفاده قرار دادیم.

### ۳-۸ پردازش موازی

یکی از روش‌های متداول برای بهبود سرعت اجرای برنامه‌ها، استفاده از پردازش موازی است. کاهش زمان محاسبه، امکان حل مسائل بزرگ با بار محاسباتی زیاد، از جمله برتری‌های پردازش موازی نسبت به پردازش سری است.

پردازش موازی در واقع اجرای چند فرایند به صورت هم‌زمان است. این روش برای افزایش کارایی و به‌دست آوردن جواب در کوتاه‌ترین زمان است. گاهی اوقات اشتراک زمان بین چندین فرایند و انجام محاسبات بر روی یک پردازنده تک‌هسته‌ای به اشتباه پردازش موازی تلقی می‌شود. به همین دلیل

---

<sup>1</sup> Multilayer Perceptron (MLP)

برای انجام پردازش موازی بر روی یک سیستم باید به اطلاعات سخت‌افزاری آن توجه شود. زیرا پردازش موازی تنها روی پردازنده‌های چند هسته‌ای، معنا دارد. ایده‌ی این کار، بر این مبنا است که هر مسئله به صورت معمول قابل تقسیم به چندین مسئله با اندازه‌ی کوچک‌تر است که این مسئله‌های کوچک می‌توانند به صورت هم‌زمان حل شوند و در نهایت با جمع‌بندی جواب مسئله‌های کوچک‌تر نتیجه نهایی سریع‌تر به دست آید.

### ۳-۸-۱ برنامه‌نویسی موازی

برنامه‌نویسی<sup>۱</sup> موازی به منظور استفاده بهینه از منابع سخت‌افزاری و افزایش سرعت و کارایی برنامه بر روی پردازنده‌ها ایجاد شد. در این شیوه برنامه‌نویسی، قسمت‌هایی از برنامه اصلی که استعداد اجرای هم‌زمان<sup>۲</sup> را دارند به چند زیربرنامه تقسیم می‌شوند. سپس به صورت هم‌زمان روی چند پردازنده اجرا می‌شوند. بخشی از برنامه هم که قابلیت اجرای موازی را ندارد به صورت سری بر روی یک پردازنده اجرا می‌شود. در حقیقت فرق اصلی برنامه‌سازی سری و موازی همین مسئله است، اما در پی آن مفاهیم فراوانی عنوان می‌شود که بیشتر اوقات در برنامه‌نویسی معمولی مطرح نبوده یا حقیقت آن‌ها با مفهوم متناظر آن در برنامه‌نویسی معمولی کاملاً متمایز است.

برنامه‌نویسی به کمک سکوه‌های OpenCL، CUDA یا با استفاده از Multithreading روش‌های

متداول برنامه‌نویسی موازی است که در ذیل به آن‌ها اشاره می‌کنیم.

---

<sup>1</sup> Parallel Programming

<sup>2</sup> Concurrent

### ۳-۸-۱ برنامه‌نویسی با استفاده از CUDA

CUDA<sup>۱</sup> جز اولین سکوه‌های ارائه‌شده پردازش موازی است که توسط شرکت NVIDIA ابداع شده است. CUDA موتور پردازشی در GPUهای NVIDIA است که توسط برنامه‌نویسان و در زبان برنامه‌نویسی خاص خود قابل دسترسی است. CUDA مبتنی بر نسخه خاصی از زبان قدرتمند C است. هر برنامه کودا در واقع برنامه سریالی است که شامل کرنل موازی است. کد زنجیره<sup>۲</sup> زبان C در درون هسته‌های پردازشی میزبان یا همان ریسمان‌های CPU<sup>۳</sup> اجرا می‌شود و هسته موازی کد C درون تعداد زیادی از هسته‌های پردازشی واحد کودا یا همان هسته‌های پردازشی GPU اجرا می‌شوند.

این شیوه فقط بر روی پردازنده‌های گرافیکی شرکت NVIDIA قابل اجرا است و فعلاً برای گوشی‌های هوشمند ارائه نشده است.

### ۳-۸-۲ برنامه‌نویسی با استفاده از OpenCL

OpenCL<sup>۴</sup> به معنای زبان محاسباتی آزاد است. زبان OpenCL را می‌توان روی انواع پردازنده‌ها شامل CPU و GPU<sup>۵</sup> اجرا کرد. OpenCL برخلاف CUDA بر اساس استاندارد عمومی تعریف شده است که کاربران غیرتخصصی نیز می‌توانند از آن استفاده کنند و روی تمامی ابزارهای مورد نیازشان آن را توسعه دهند. در برنامه‌نویسی موازی به روش OpenCL، وظایف مختلف

---

<sup>۱</sup> Compute Unified Device Architecture

<sup>۲</sup> serial

<sup>۳</sup> Central processing unit

<sup>۴</sup> Open Computing Language

<sup>۵</sup> Graphics processing unit

محاسباتی به چند بخش پردازشی که به طور همزمان باهم کار می کنند اختصاص داده می شود. این وظیفه ها کرنل نامیده می شوند که کد کرنل باید به وسیله یک یا چند پردازنده سازگار اجرا شود.

این فن آوری هنوز به صورت جدی برای پردازنده های گوشی های هوشمند ارائه نشده است. و تنها برخی سخت افزارهای مرتبط با گوشی هوشمند را پشتیبانی می کند و معمولاً شامل گوشی های پرچم دار شرکت های سازنده تلفن همراه می شود.

### ۳-۱-۸-۳ افزایش سرعت اجرای برنامه با الگوریتم های چند ریسمانی<sup>۱</sup>

در برنامه نویسی های متداول وقتی برنامه ای را اجرا می کنیم به صورت دستور به دستور در طی فرایندی یکنواخت یا در یک ریسمان مجزا اجرا می شود. یعنی در این شیوه ی برنامه نویسی، در هر زمان فقط یک دستور می تواند اجرا شود و تا دستور خط فعلی به پایان نرسد خط بعدی برنامه اجرا نخواهد شد. همچنین در این روش معمولاً CPU های چند هسته ای تنها از یک هسته خود برای پردازش استفاده می کنند. بنابراین از توان هسته های دیگر برای پردازش استفاده نمی شود و سرعت هم افزایش پیدا نخواهد کرد.

یکی از روش های افزایش بخشیدن به سرعت اجرای برنامه استفاده از الگوریتم های چند ریسمانی است. الگوریتم های چند ریسمانی برای انجام کارهای موازی بر روی رایانه های چند پردازنده اجرا می شوند

---

<sup>1</sup> Multithreading



. این الگوریتم به پردازنده این اجازه را می‌دهد تا برنامه‌ها به صورت موازی اجرا شوند. یعنی به پردازنده

مجوز اجرای چندین دستورالعمل به صورت هم‌زمان را می‌دهد.

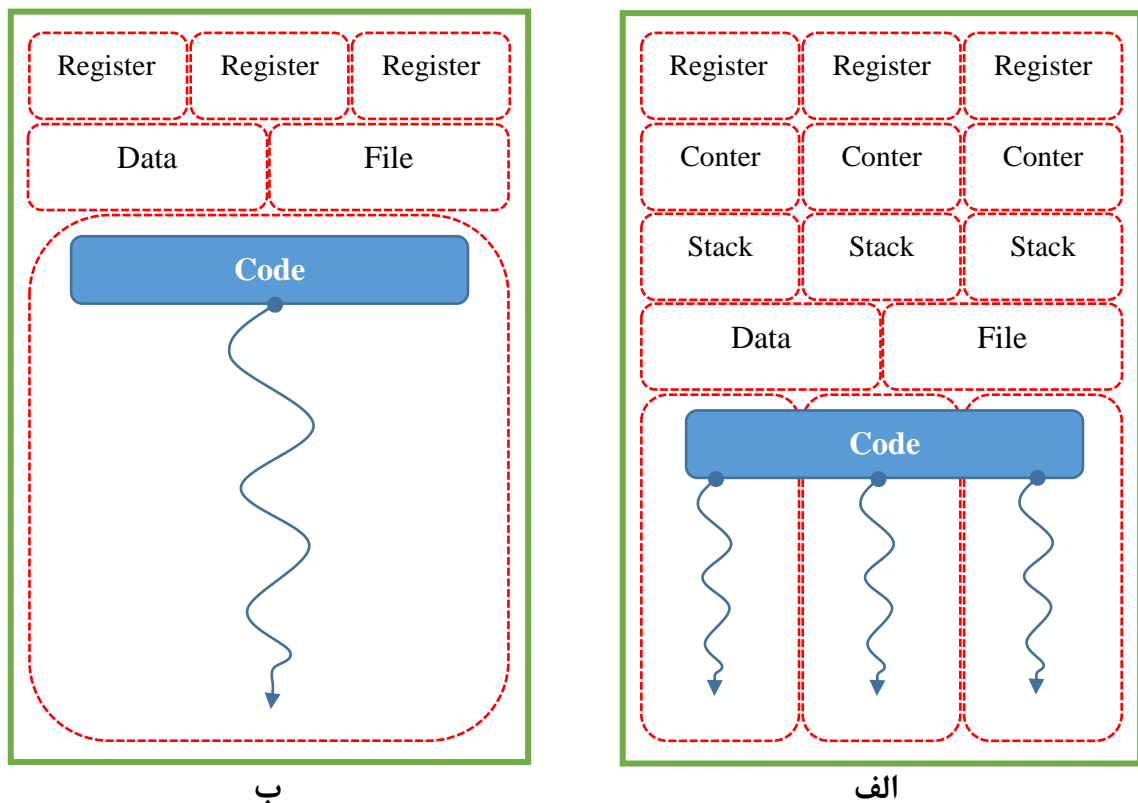
در کد نویسی جاوا می‌توان برای CPUهای تک‌هسته‌ای نیز از الگوریتم‌های چند ریسمانی استفاده

کرد. اما این روش فقط باعث می‌شود تا بتوانیم چندین برنامه مختلف را به صورت هم‌زمان اجرا کنیم و

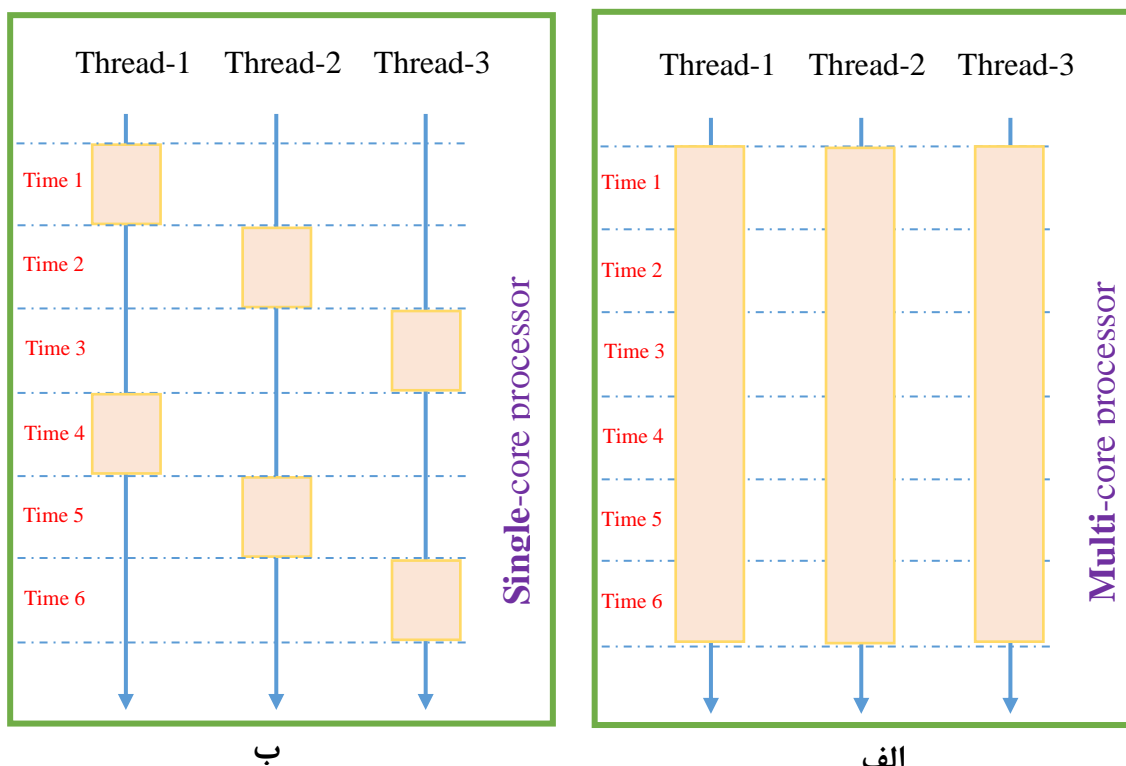
لزوماً افزایش سرعت تضمین نمی‌شود، تنها زمان پردازش پردازنده‌ی اصلی، بین ریسمان‌ها، به اشتراک

گذاشته می‌شود. شکل ۳-۱۸ عملکرد برنامه‌ی تک ریسمانی و چند ریسمانی را نشان می‌دهد. شکل

۳-۱۹ اجرای الگوریتم چند ریسمانی بر روی CPUهای تک و چندهسته‌ای است.



شکل ۳-۱۸ الف) عملکرد برنامه چند ریسمانی، ب) عملکرد برنامه تک ریسمانی



شکل ۳-۱۹ زمان اجرای برنامه چند ریسمانی بر روی CPU الف) چند هسته‌ای و ب) تک‌هسته‌ای

یکی از روش‌های ارائه‌شده در این تحقیق قطعه‌بندی افقی تصویر اصلی به چندین تصویر کوچک‌تر

است. این قطعات برای بازدهی بهتر معمولاً به تعداد هسته‌های پردازنده محاسباتی انتخاب می‌شوند.

شکل ۳-۲۰ طریق قطعه‌بندی تصویر و عملکرد ریسمان‌ها را نشان می‌دهد.

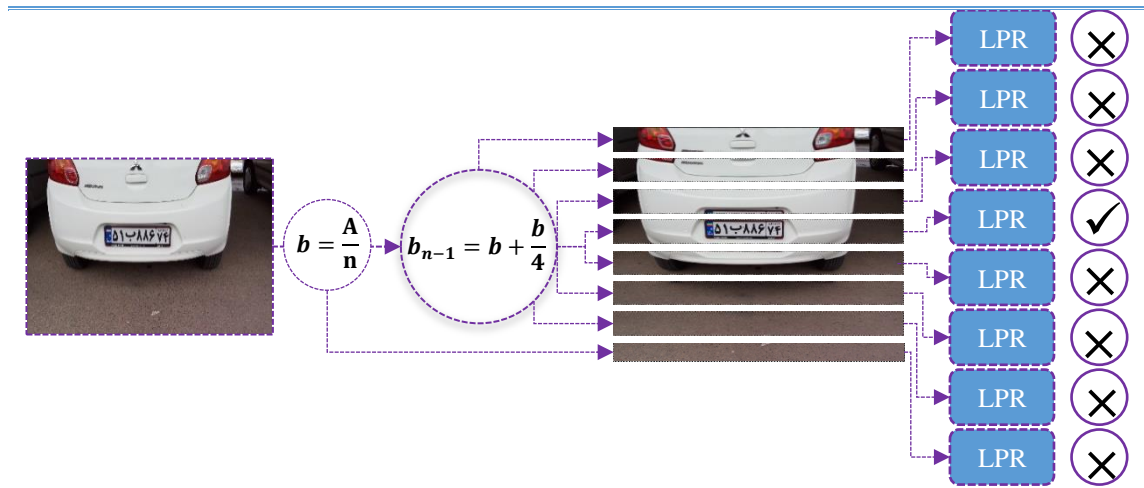
تعداد ریسمان‌های انتخاب‌شده نباید از تعداد هسته‌های پردازنده بیشتر باشد زیرا این افزایش

لزوماً افزایش سرعت پردازش را تضمین نخواهد کرد تنها باعث می‌شود که پردازنده زمان پردازش هر

هسته خود را بین چندین ریسمان به اشتراک بگذارد.

در این تحقیق از ۲ تا ۸ ریسمان در جهت بهبود عملکرد زمانی سامانه تشخیص پلاک استفاده

شده است که در فصل چها نتایج ۱ تا ۵ ریسمان را آورده ایم . نتایج به دست آمده کاهش زمان محاسباتی به حدود ۳۰٪ تا ۶۰٪ زمان پردازش اصل را نشان دادند.



شکل ۳-۲۰ نحوه اجرای برنامه چند ریسمانی

در شکل ۳-۲۰،  $A$  تصویر اصلی،  $b$  تصویر برش خورده و  $n$  تعداد ریسمان ها است. با فرض این که  $n$  برابر ۸ باشد، ۸ تصویر برش خورده داریم که  $n-1$  تصویر با همپوشانی  $\frac{1}{4}$  تصویر  $b$  خواهد بود و تصویر آخر بدون همپوشانی در نظر گرفته خواهد شد. بر روی هر قطعه از تصویر، کل مراحل تشخیص پلاک صورت می پذیرد. در نهایت هر قطعه از تصویر که حاوی پلاک باشد به عنوان برنده انتخاب خواهد شد و پلاک نمایش داده خواهد شد.

# فصل چہارم

## نتائج تجربی

## ۱-۴ نتایج تجربی

آزمایش‌های انجام‌شده بر روی تصاویر گرفته‌شده توسط دوربین تلفن همراه صورت پذیرفته است. این پایگاه داده شامل حدود ۱۲۰ تصویر است. وضوح تصاویر گرفته از 0.3Mp تا 2Mp متفاوت بود. الگوریتم ارائه‌شده تقریباً می‌تواند پلاک موجود در تمام تصاویری که از فاصله ۱ تا ۸ متری گرفته‌شده‌اند را به‌خوبی تشخیص دهد. درصد تشخیص محل پلاک و حروف به ترتیب ۹۷/۸٪ و ۹۶/۱٪ به‌دست آمد. هنگام عکس‌برداری اگر فاصله تا خودرو و زاویه عکس‌برداری رعایت شود درصد تشخیص محل پلاک ۱۰۰ درصد به‌دست خواهد آمد.

روش ارائه‌شده به خاطر بعضی از محدودیت‌های سیستم‌عامل اندروید برای نمایش مراحل مختلف پروژه ابتدا به کمک زبان C++ و با استفاده از کتابخانه OpenCV-2.4.13 در Visual Studio 2013 مورد ارزیابی قرار گرفت و سپس به کمک زبان Java و با استفاده از کتابخانه OpenCV-2.4.13-sdk و AndroidStudio بر روی گوشی LG از کلاس میان‌رده با مشخصات فرکانس ۱/۳GH، پردازنده ۸ هسته‌ای مدل 8x ARM Cortex-A53 و رم ۲ انجام پذیرفت. نسبت سرعت پردازش‌های انجام‌شده بر روی تصاویر مختلف در رایانه و گوشی اندروید تقریباً ۱ به ۲۰ به‌دست آمد یعنی سرعت پردازش بر روی رایانه ۲۰ برابر گوشی اندروید به‌دست آمد که منطقی به نظر می‌رسد، زیرا واضح است که بدون شک سرعت و قدرت پردازنده رایانه از پردازنده‌ی در نظر گرفته‌شده برای گوشی اندروید بیشتر خواهد بود. زمان اجرای برنامه برای تصاویر با وضوح مختلف طبق جدول ۱-۴ به‌دست آمد.

جدول ۴-۱ نتایج خروجی نرم‌افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح مختلف

| ردیف | وضوح تصویر  | فاصله تا دوربین / متر | سیسم‌عامل اندروید    |                     |                      | رایانه               |
|------|-------------|-----------------------|----------------------|---------------------|----------------------|----------------------|
|      |             |                       | بیشترین زمان / ثانیه | کمترین زمان / ثانیه | میانگین زمان / ثانیه | میانگین زمان / ثانیه |
| ۱    | ۱۰۸۰ × ۱۹۲۰ | ۴-۸                   | ۱۰/۵۸                | ۰۳/۰۵               | ۵/۶۳                 | ۰/۵۷۱                |
| ۲    | ۱۹۲۰ × ۱۰۸۰ | ۲-۴                   | ۰۱۰/۴۱               | ۰۴/۱                | ۵/۸۰                 | ۰/۵۸۲                |
| ۳    | ۹۲۰ × ۵۴۰   | ۱-۲                   | ۰۳/۰۹                | ۰۱/۰۷               | ۱/۸۳                 | ۰/۲۴۳                |
| ۴    | ۳۲۰ × ۲۴۰   | ۱                     | ۰۰/۸۸                | ۰۰/۵۴               | ۰/۶۷                 | ۰/۱۳۱                |

در جدول‌های ۴-۳، ۴-۴ مثال‌هایی از بازشناسی ارقام با استفاده از روش یک ریسمان تا ۵

ریسمان را آورده‌ایم. در جدول ۴-۵ نیز به خاطر کوچک بودن اندازه تصاویر حداکثر از سه ریسمان

استفاده کرده‌ایم. در جدول‌های ۴-۳، ۴-۴ و ۴-۵ مواردی که به اشتباه تشخیص داده شده است را با

صورت مشکی و مواردی که تشخیص داده نشده است را با خط تیره نشان داده‌ام.

همان‌طور که اشاره شد حداکثر ریسمان به کار گرفته شده برای پردازش موازی در این تحقیق ۸

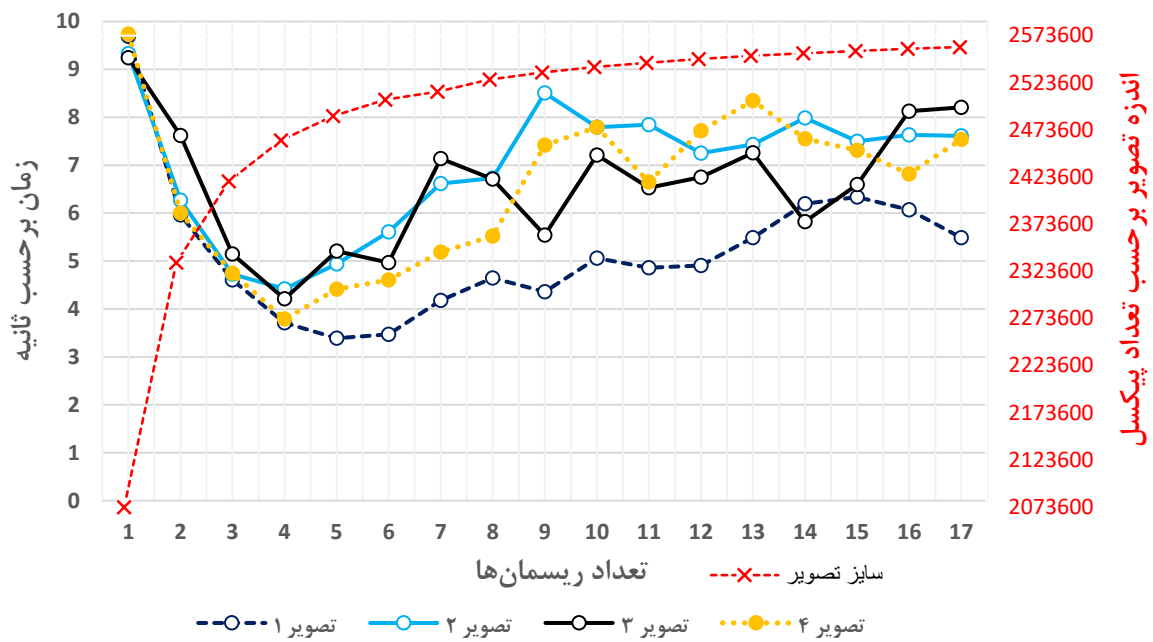
عدد است زیرا با افزایش تعداد ریسمان‌ها میزان همپوشانی قطعات تصویر نیز افزایش پیدا می‌کند که

قطعاً در کاهش سرعت پردازش تأثیرگذار خواهد بود. در شکل ۴-۱ با رسم یک نمودار برای چهار تصویر

مختلف روند افزایش و سپس کاهش سرعت را بر اساس افزایش تعداد ریسمان‌ها نشان داده‌ایم.

جدول ۴-۲ نتایج زمانی خروجی نرم‌افزار برای ۱ تا ۱۷ ریسمان

| ردیف | زمان بر حسب ثانیه برای تصاویری با وضوح ۱۹۲۰ * ۱۰۸۰ |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|--|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|      | ۱  | ۲    | ۳    | ۴    | ۵    | ۶    | ۷    | ۸    | ۹    | ۱۰   | ۱۱   | ۱۲   | ۱۳   | ۱۴   | ۱۵   | ۱۶   | ۱۷   |
| ۱    | ۹,۶۹   | ۵,۹۶ | ۴,۶۱ | ۳,۷۱ | ۳,۳۹ | ۳,۴۷ | ۴,۱۸ | ۴,۶۵ | ۴,۳۶ | ۵,۰۶ | ۴,۸۶ | ۴,۹۱ | ۵,۴۹ | ۶,۲  | ۶,۳۴ | ۶,۰۷ | ۵,۴۹ |
| ۲    | ۹,۳۳   | ۶,۲۷ | ۴,۷۳ | ۴,۴۲ | ۴,۹۴ | ۵,۶۱ | ۶,۶۲ | ۶,۷۳ | ۸,۵۱ | ۷,۷۹ | ۷,۸۵ | ۷,۲۵ | ۷,۴۳ | ۷,۹۹ | ۷,۵  | ۷,۶۳ | ۷,۶۱ |
| ۳    | ۹,۲۴   | ۷,۶۲ | ۵,۱۵ | ۴,۲۱ | ۵,۲۱ | ۴,۹۷ | ۷,۱۴ | ۶,۷۱ | ۵,۵۴ | ۷,۲۱ | ۶,۵۳ | ۶,۷۵ | ۷,۲۶ | ۵,۸۲ | ۶,۶  | ۸,۱۳ | ۸,۲۱ |
| ۴    | ۹,۷۴   | ۶    | ۴,۷۵ | ۳,۷۹ | ۴,۴۱ | ۴,۶۱ | ۵,۱۹ | ۵,۵۲ | ۷,۴۲ | ۷,۷۹ | ۶,۶۵ | ۷,۷۲ | ۸,۳۵ | ۷,۵۵ | ۷,۳۱ | ۶,۸۲ | ۷,۵۴ |



شکل ۴-۱ روند افزایش و سپس کاهش سرعت بر اساس انتخاب تعداد ریسمان‌ها

همان‌طور که در جدول ۴-۲ دیده می‌شود سریع‌ترین پاسخ‌ها را برای ۲ تا ۵ ریسمان بدست آوردیم.

در شکل ۴-۲ تا ۴-۵ مواردی که به‌درستی تشخیص داده‌شده‌اند را در سه دقت متفاوت آورده‌ایم.

همان‌طور که دیده می‌شود پلاک‌هایی که تا حدی به صورت زاویه‌دار عکس‌برداری شده‌اند به‌درستی









تشخیص داده‌شده‌اند. همچنین پلاک‌هایی که در معرض آلودگی و سایه بوده‌اند نیز به‌درستی تشخیص

داده‌شده‌اند.

شکل ۴-۶ تصاویری را نشان می‌دهد که در آن پلاک یا تشخیص داده نشده است و یا به اشتباه



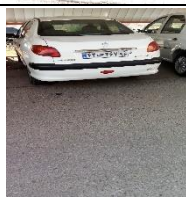
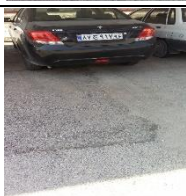
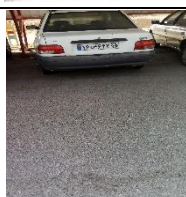

تشخیص داده شده است.

جدول ۳-۴ بررسی خروجی نرم افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح ۱۹۲۰ × ۱۰۸۰

| سایز تصاویر                        | تعداد ریسمان ها | شماره ردیف و عکس | شماره پلاک |   |      |     |   |     | زمان/ثابت | تصاویر استفاده شده |      |   |   |
|------------------------------------|-----------------|------------------|------------|---|------|-----|---|-----|-----------|--------------------|------|---|---|
|                                    |                 |                  | عدد        |   | حروف | عدد |   | عدد |           |                    |      |   |   |
| ۱۰۸۰ × ۱۹۲۰<br>فاصله<br>۸ تا ۴ متر | ۱               | ۰۰۱۴             | ۱          | ۱ | ج    | ۲   | ۱ | ۳   | ۹         | ۶                  | ۹/۷۸ |  |   |
|                                    | ۲               |                  | ۱          | ۱ | ج    | ۲   | ۱ | ۳   | ۹         | ۶                  | ۶/۲۰ |   |   |
|                                    | ۳               |                  | ۱          | ۱ | ج    | ۲   | ۱ | ۳   | ۹         | ۶                  | ۴/۴۸ |   |   |
|                                    | ۴               |                  | ۱          | ۱ | ج    | ۳   | ۱ | ۳   | ۹         | ۶                  | ۳/۸۴ |   |   |
|                                    | ۵               |                  | ۱          | ۱ | ج    | ۲   | ۱ | ۳   | ۹         | ۶                  | ۳/۶۵ |   |   |
|                                    | ۱               | ۰۰۰۲             | ۲          | ۱ | ۷    | ص   | ۲ | ۷   | ۶         | ۹                  | ۶    | ۹/۹۶  |    |
|                                    | ۲               |                  |            | ۱ | ۷    | ص   | ۲ | ۷   | ۶         | ۹                  | ۶    | ۵/۹۲  |   |
|                                    | ۳               |                  |            | ۱ | ۷    | ص   | ۲ | ۷   | ۶         | ۹                  | ۶    | ۴/۸۰  |   |
|                                    | ۴               |                  |            | ۱ | ۷    | ص   | ۲ | ۷   | ۶         | ۹                  | ۶    | ۴/۳۳  |   |
|                                    | ۵               |                  |            | ۱ | ۷    | ص   | ۲ | ۷   | ۶         | ۹                  | ۶    | ۳/۹۹  |   |
|                                    | ۱               | ۰۰۱۰             | ۳          | ۱ | ۲    | ج   | ۴ | ۴   | ۶         | ۹                  | ۶    | ۹/۴۴  |   |
|                                    | ۲               |                  |            | ۱ | ۲    | ج   | ۴ | ۴   | ۶         | ۹                  | ۶    | ۶/۱۲  |   |
|                                    | ۳               |                  |            | ۱ | ۲    | ج   | ۴ | ۴   | ۶         | ۹                  | ۶    | ۴/۳۸  |   |
|                                    | ۴               |                  |            | ۱ | ۲    | ج   | ۴ | ۴   | ۶         | ۹                  | ۶    | ۳/۹۸  |   |
|                                    | ۵               |                  |            | ۱ | ۲    | ج   | ۴ | ۴   | ۶         | ۹                  | ۶    | ۳/۳۲  |   |
|                                    | ۱               | ۰۰۱۲             | ۴          | ۱ | ۴    | ص   | ۶ | ۵   | ۵         | ۹                  | ۶    | ۱۰/۱۱   |  |
|                                    | ۲               |                  |            | ۱ | ۴    | ص   | ۶ | ۵   | ۵         | ۹                  | ۶    | ۶/۶۳  |   |
|                                    | ۳               |                  |            | ۱ | ۴    | ص   | ۶ | ۵   | ۵         | ۹                  | ۶    | ۴/۸۵  |   |
|                                    | ۴               |                  |            | ۱ | ۴    | ص   | ۶ | ۵   | ۵         | ۹                  | ۶    | ۴/۴۳  |   |
|                                    | ۵               |                  |            | ۱ | ۴    | ص   | ۶ | ۵   | ۵         | ۹                  | ۶    | ۴/۶۰  |   |
|                                    | ۱               | ۰۰۱۶             | ۵          | ۷ | ۷    | د   | ۵ | ۵   | ۹         | ۱                  | ۲    | ۱۰/۱۴   |  |
|                                    | ۲               |                  |            | - | -    | --  | - | -   | -         | -                  | -    | ۵/۹۰  |   |
|                                    | ۳               |                  |            | ۷ | ۷    | د   | ۵ | ۵   | ۹         | ۱                  | ۲    | ۴/۳۹  |   |
|                                    | ۴               |                  |            | ۷ | ۷    | د   | ۵ | ۵   | ۹         | ۱                  | ۲    | ۳/۳۶  |   |
|                                    | ۵               |                  |            | ۷ | ۷    | د   | ۵ | ۵   | ۹         | ۱                  | ۲    | ۳/۸۰  |   |
|                                    | ۱               | ۰۰۰۷             | ۶          | ۴ | ۲    | ج   | ۴ | ۷   | ۲         | ۹                  | ۶    | ۱۰/۵۵   |  |
|                                    | ۲               |                  |            | ۴ | ۲    | ج   | ۴ | ۷   | ۲         | ۹                  | ۶    | ۶/۲۵  |   |
|                                    | ۳               |                  |            | ۴ | ۲    | ج   | ۴ | ۷   | ۲         | ۹                  | ۶    | ۴/۹۹  |   |
|                                    | ۴               |                  |            | ۴ | ۲    | ج   | ۴ | ۷   | ۲         | ۹                  | ۶    | ۴/۱۵  |   |
|                                    | ۵               |                  |            | ۴ | ۲    | ج   | ۴ | ۷   | ۲         | ۹                  | ۶    | ۹۳/۴  |   |
|                                    | ۱               | ۰۰۰۶             | ۷          | ۵ | ۳    | ص   | ۶ | ۸   | ۳         | ۹                  | ۶    | ۱۰/۵۷   |  |
|                                    | ۲               |                  |            | - | -    | --  | - | -   | -         | -                  | -    | ۵/۴۲  |   |
|                                    | ۳               |                  |            | ۵ | ۳    | ص   | ۶ | ۸   | ۳         | ۹                  | ۶    | ۴/۴۶  |   |
|                                    | ۴               |                  |            | ۵ | ۳    | ص   | ۶ | ۸   | ۳         | ۹                  | ۶    | ۳/۵۰  |   |
|                                    | ۵               |                  |            | ۵ | ۳    | ص   | ۶ | ۸   | ۳         | ۹                  | ۶    | ۳/۲۹  |   |
|                                    | ۱               | ۰۰۱۹             | ۸          | ۶ | ۷    | د   | ۵ | ۷   | ۱         | ۳                  | ۶    | ۱۰/۸۵   |  |
|                                    | ۲               |                  |            | - | -    | --  | - | -   | -         | -                  | -    | ۶/۵۷  |   |
|                                    | ۳               |                  |            | ۶ | ۷    | د   | ۵ | ۷   | ۱         | ۳                  | ۶    | ۴/۷۳  |   |
|                                    | ۴               |                  |            | ۶ | ۷    | د   | ۵ | ۷   | ۱         | ۳                  | ۶    | ۳/۹۴  |   |
|                                    | ۵               |                  |            | ۶ | ۷    | د   | ۵ | ۷   | ۱         | ۳                  | ۶    | ۳/۵۵  |   |





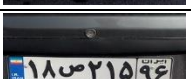








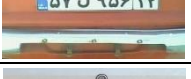

ادامه جدول ۳-۴

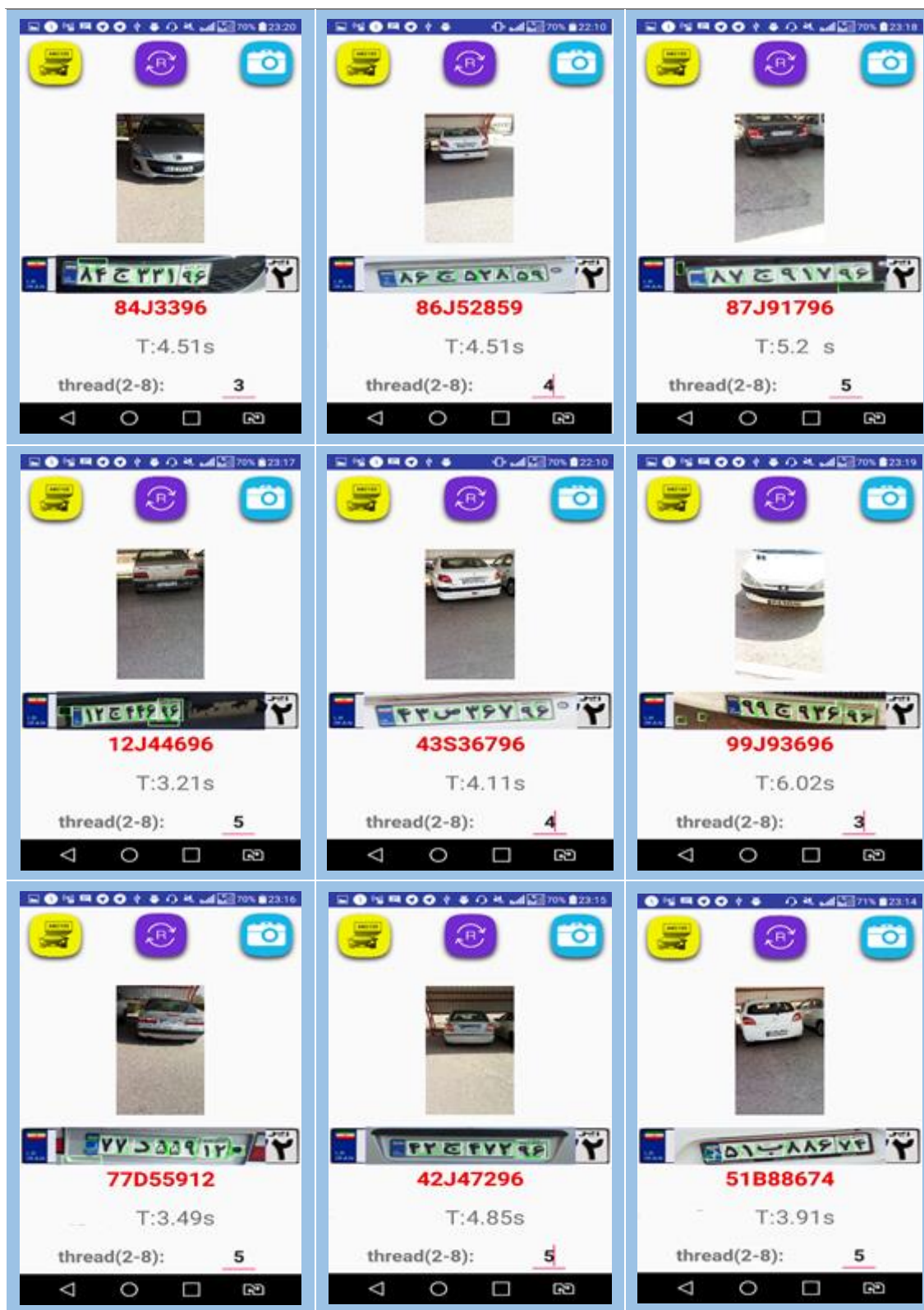
| سایز تصاویر                        | تعداد ریسمانها | شماره ردیف و عکس | شماره پلاک |      |     |     | زمان/تایمه |   | تصاویر استفاده شده |       |   |   |
|------------------------------------|----------------|------------------|------------|------|-----|-----|------------|---|--------------------|-------|---|---|
|                                    |                |                  | عدد        | حروف | عدد | عدد |            |   |                    |       |   |   |
| ۱۰۸۰ × ۱۹۲۰<br>فاصله<br>۴ تا ۸ متر | ۱              | ۹<br>۰۰۱۸        | ۸          | ۴    | ج   | ۳   | ۳          | ۱ | ۹                  | ۶     | ۱۰/۵۸   |    |
|                                    | ۲              |                  | ۸          | ۴    | ج   | ۳   | ۳          | ۱ | ۹                  | ۶     | ۶/۴۶  |   |
|                                    | ۳              |                  | ۸          | ۴    | ج   | ۳   | ۳          | ۱ | ۹                  | ۶     | ۴/۴۱  |   |
|                                    | ۴              |                  | ۸          | ۴    | ج   | ۳   | ۳          | ۱ | ۹                  | ۶     | ۴/۰۱  |   |
|                                    | ۵              |                  | ۸          | ۴    | ج   | ۳   | ۳          | ۱ | ۹                  | ۶     | ۳/۴۷  |   |
|                                    | ۱              | ۱۰<br>۰۰۰۱       | ۹          | ۹    | ج   | ۹   | ۳          | ۶ | ۹                  | ۶     | ۹/۹۹  |    |
|                                    | ۲              |                  | ۹          | ۹    | ج   | ۹   | ۳          | ۶ | ۹                  | ۶     | ۶/۵۱  |   |
|                                    | ۳              |                  | ۹          | ۹    | ج   | ۹   | ۳          | ۶ | ۹                  | ۶     | ۵/۱۰  |   |
|                                    | ۴              |                  | ۹          | ۹    | ج   | ۹   | ۳          | ۶ | ۹                  | ۶     | ۴/۸۷  |   |
|                                    | ۵              |                  | ۹          | ۹    | ج   | ۹   | ۳          | ۶ | ۹                  | ۶     | ۷/۹۶  |   |
|                                    | ۱              | ۱۱<br>۰۰۱۳       | ۴          | ۳    | ص   | ۳   | ۶          | ۷ | ۹                  | ۶     | ۹/۹۷  |   |
|                                    | ۲              |                  | ۴          | ۳    | ص   | ۳   | ۶          | ۷ | ۹                  | ۶     | ۶/۵۹  |   |
|                                    | ۳              |                  | ۴          | ۳    | ص   | ۳   | ۶          | ۷ | ۹                  | ۶     | ۴/۹۵  |   |
|                                    | ۴              |                  | ۴          | ۳    | ص   | ۳   | ۶          | ۷ | ۹                  | ۶     | ۴/۳۵  |   |
|                                    | ۵              |                  | ۴          | ۳    | ص   | ۳   | ۶          | ۷ | ۹                  | ۶     | ۴/۸۶  |   |
|                                    | ۱              | ۱۲<br>۰۰۰۸       | ۸          | ۷    | ج   | ۹   | -          | ۷ | ۹                  | ۶     | ۹/۹۳  |  |
|                                    | ۲              |                  | -          | -    | --  | -   | -          | - | -                  | -     | ۶/۳۹  |   |
|                                    | ۳              |                  | ۸          | ۷    | ج   | ۹   | ۱          | ۷ | ۹                  | ۶     | ۴/۹۵  |   |
|                                    | ۴              |                  | ۸          | ۷    | ج   | ۹   | ۱          | ۷ | ۹                  | ۶     | ۴/۲۳  |   |
|                                    | ۵              |                  | ۸          | ۷    | ج   | ۹   | ۱          | ۷ | ۹                  | ۶     | ۴/۴۶  |   |
|                                    | ۱              | ۱۳<br>۰۰۰۹       | ۱          | ۶    | ص   | ۶   | ۳          | ۳ | ۹                  | ۶     | ۹/۹۶  |  |
|                                    | ۲              |                  | ۱          | ۶    | ص   | ۶   | ۳          | ۳ | ۹                  | ۶     | ۶/۴۳  |   |
|                                    | ۳              |                  | ۱          | ۶    | ص   | ۶   | ۳          | ۳ | ۹                  | ۶     | ۴/۴۷  |   |
|                                    | ۴              |                  | ۱          | ۶    | ص   | ۶   | ۳          | ۳ | ۹                  | ۶     | ۳/۰۷  |   |
|                                    | ۵              |                  | ۱          | ۶    | ص   | ۶   | ۳          | ۳ | ۹                  | ۶     | ۴/۰۹  |   |
| ۱                                  | ۱۴<br>۰۰۰۴     | ۶                | ۷          | د    | ۵   | ۷   | ۱          | ۳ | ۶                  | ۱۱/۱۷ |  |   |
| ۲                                  |                | ۶                | ۷          | د    | ۵   | ۷   | ۱          | ۳ | ۶                  | ۶/۶۴  |   |   |
| ۳                                  |                | ۶                | ۷          | د    | ۵   | ۷   | ۱          | ۳ | ۶                  | ۴/۶۷  |   |   |
| ۴                                  |                | ۶                | ۷          | د    | ۵   | ۷   | ۱          | ۳ | ۶                  | ۴/۳۰  |   |   |
| ۵                                  |                | ۶                | ۷          | د    | ۵   | ۷   | ۱          | ۳ | ۶                  | ۴/۲۰  |   |   |
| ۱                                  | ۱۵<br>۰۰       |                  |            |      |     |     |            |   |                    |       |   |   |
| ۲                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۳                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۴                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۵                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۱                                  | ۱۶<br>۰۰       |                  |            |      |     |     |            |   |                    |       |   |   |
| ۲                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۳                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۴                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |
| ۵                                  |                |                  |            |      |     |     |            |   |                    |       |   |   |

جدول ۴-۴ بررسی خروجی نرم افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح ۵۴۰ × ۹۲۰

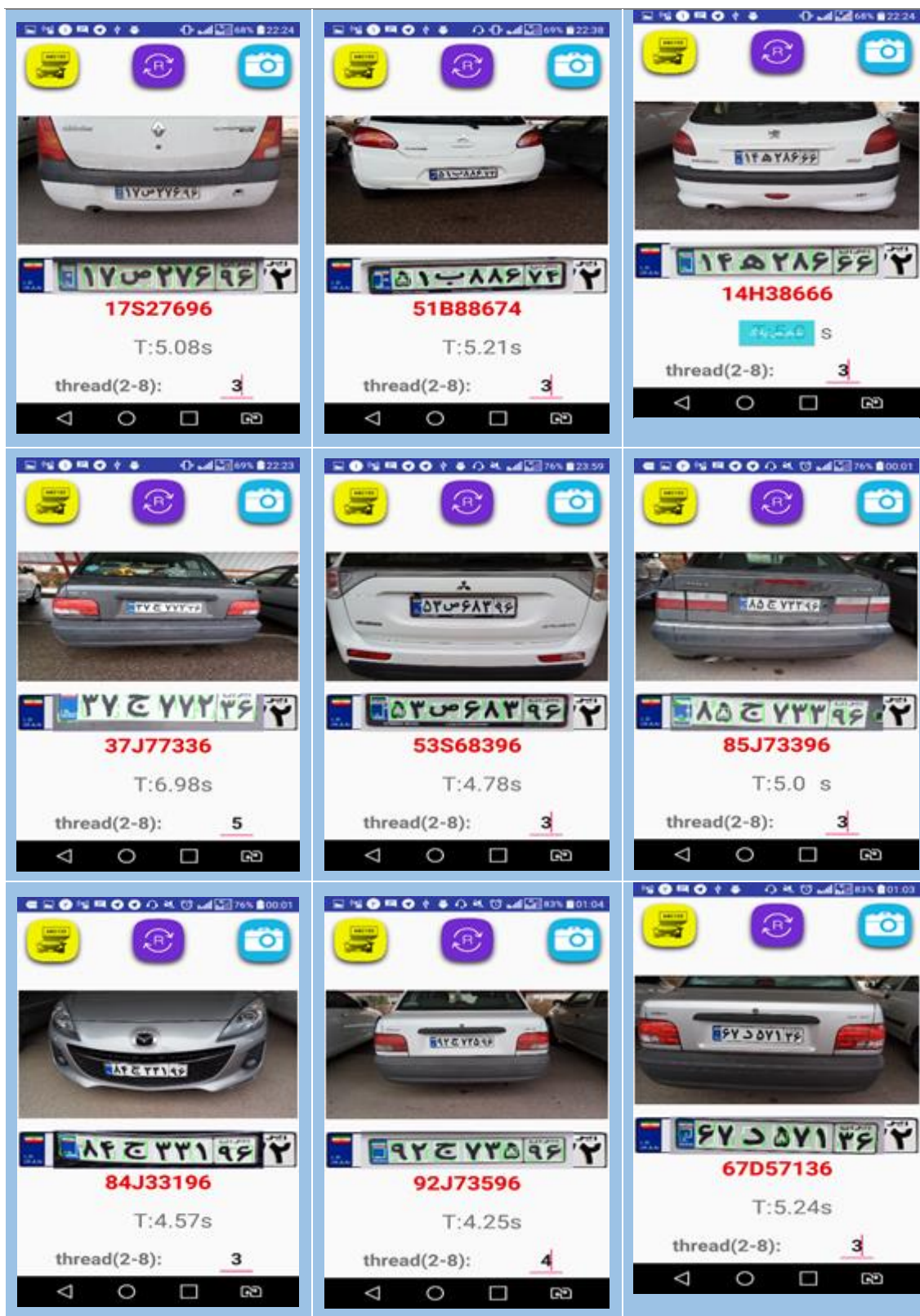
| سایز تصاویر                      | تعداد ریسمانها | شماره ردیف و عکس | شماره پلاک |   |       |     |   |     | زمان/تایمه | تصاویر استفاده شده |      |  |
|----------------------------------|----------------|------------------|------------|---|-------|-----|---|-----|------------|--------------------|------|--|
|                                  |                |                  | عدد        |   | خروجی | عدد |   | عدد |            |                    |      |  |
| ۹۲۰ × ۵۴۰<br>فاصله<br>۱ تا ۲ متر | ۱              | ۰۰۳۳             | ۱          | ۴ | ص     | ۶   | ۵ | ۵   | ۹          | ۶                  | ۲/۸۴ |  |
|                                  |                |                  | ۲          | ۴ | ص     | ۶   | ۵ | ۵   | ۹          | ۶                  | ۱/۸۶ |  |
|                                  |                |                  | ۳          | ۴ | ص     | ۶   | ۵ | ۵   | ۹          | ۶                  | ۱/۵۵ |  |
|                                  |                |                  | ۴          | ۴ | ص     | ۶   | ۵ | ۵   | ۹          | ۶                  | ۲/۰۳ |  |
|                                  |                |                  | ۵          | ۴ | ص     | ۶   | ۵ | ۵   | ۹          | ۶                  | ۲/۴۴ |  |
|                                  | ۲              | ۰۰۲۹             | ۱          | ۴ | ۰     | ۲   | ۸ | ۶   | ۶          | ۶                  | ۲/۴۵ |  |
|                                  |                |                  | ۲          | ۴ | ۰     | ۲   | ۸ | ۶   | ۱          | ۶                  | ۱/۷۶ |  |
|                                  |                |                  | ۳          | ۴ | ۰     | ۲   | ۸ | ۶   | ۱          | ۶                  | ۱/۴۵ |  |
|                                  |                |                  | ۴          | ۴ | ۰     | ۲   | ۸ | ۶   | ۶          | ۶                  | ۱/۴۹ |  |
|                                  |                |                  | ۵          | ۴ | ۰     | ۲   | ۸ | ۶   | ۶          | ۶                  | ۲/۱۲ |  |
|                                  | ۳              | ۰۰۲۷             | ۱          | ۷ | ج     | ۹   | ۱ | ۷   | ۹          | ۶                  | ۳/۰۹ |  |
|                                  |                |                  | ۲          | ۷ | ج     | ۹   | ۱ | ۷   | ۹          | ۶                  | ۲/۱۴ |  |
|                                  |                |                  | ۳          | ۷ | ج     | ۹   | ۱ | ۷   | ۹          | ۶                  | ۱/۶۶ |  |
|                                  |                |                  | ۴          | - | ج     | ۹   | ۱ | ۷   | ۹          | ۶                  | ۱/۴۳ |  |
|                                  |                |                  | ۵          | ۷ | ج     | ۹   | ۱ | ۷   | ۹          | ۶                  | ۱/۳۷ |  |
|                                  | ۴              | ۰۰۲۲             | ۱          | ۵ | ج     | ۷   | ۳ | ۳   | ۲          | ۲                  | ۲/۶۷ |  |
|                                  |                |                  | ۲          | ۵ | ج     | ۷   | ۳ | ۳   | ۲          | ۲                  | ۲/۱۷ |  |
|                                  |                |                  | ۳          | ۵ | ج     | ۷   | ۳ | ۳   | ۲          | ۲                  | ۱/۹۱ |  |
|                                  |                |                  | ۴          | ۵ | ج     | ۷   | ۳ | ۳   | ۲          | ۲                  | ۱/۵۸ |  |
|                                  |                |                  | ۵          | - | --    | -   | - | -   | -          | -                  | ۲/۵۰ |  |
|                                  | ۵              | ۰۰۱۵             | ۱          | ۸ | ۰     | ۷   | ۸ | ۶   | ۹          | ۹                  | ۲/۶۱ |  |
|                                  |                |                  | ۲          | ۸ | ۰     | ۷   | ۸ | ۶   | ۹          | ۹                  | ۱/۷۰ |  |
|                                  |                |                  | ۳          | ۸ | ۰     | ۷   | ۸ | ۶   | ۹          | ۹                  | ۱/۴۹ |  |
|                                  |                |                  | ۴          | ۸ | ۰     | ۷   | ۸ | ۶   | ۹          | ۹                  | ۱/۵۱ |  |
|                                  |                |                  | ۵          | ۸ | ۰     | ۷   | ۸ | ۶   | ۹          | -                  | ۲/۳۲ |  |
|                                  | ۶              | ۰۰۱۷             | ۱          | ۱ | ب     | ۸   | ۸ | ۶   | ۷          | ۴                  | ۲/۵۷ |  |
|                                  |                |                  | ۲          | ۱ | ب     | ۸   | ۸ | ۶   | ۷          | ۴                  | ۱/۸۷ |  |
|                                  |                |                  | ۳          | ۱ | ب     | ۸   | ۸ | ۶   | ۷          | ۴                  | ۱/۴۹ |  |
|                                  |                |                  | ۴          | ۱ | ب     | ۸   | ۸ | ۶   | ۷          | ۴                  | ۱/۴۸ |  |
|                                  |                |                  | ۵          | ۱ | ب     | ۸   | ۸ | ۶   | ۷          | ۴                  | ۲/۰۷ |  |
|                                  | ۷              | ۰۰۱۶             | ۱          | ۷ | د     | ۵   | ۷ | ۱   | ۳          | ۶                  | ۲/۳۴ |  |
|                                  |                |                  | ۲          | ۷ | د     | ۵   | ۷ | ۱   | ۳          | ۶                  | ۱/۶۴ |  |
|                                  |                |                  | ۳          | ۷ | د     | ۵   | ۷ | ۱   | ۳          | ۶                  | ۱/۴۲ |  |
|                                  |                |                  | ۴          | ۷ | د     | ۵   | ۷ | ۱   | ۳          | ۶                  | ۱/۳۶ |  |
|                                  |                |                  | ۵          | ۷ | د     | ۵   | ۷ | ۱   | ۳          | ۶                  | ۱/۸۵ |  |
|                                  | ۸              | ۰۰۳۷             | ۱          | ۱ | ج     | ۲   | ۱ | ۳   | ۹          | ۶                  | ۲/۰۶ |  |
|                                  |                |                  | ۲          | ۱ | ج     | ۲   | ۱ | ۳   | ۹          | ۶                  | ۱/۳۰ |  |
|                                  |                |                  | ۳          | ۱ | ج     | -   | ۱ | -   | ۹          | -                  | ۱/۳۵ |  |
|                                  |                |                  | ۴          | ۱ | ج     | ۲   | ۱ | ۳   | ۹          | ۶                  | ۱/۲۸ |  |
|                                  |                |                  | ۵          | ۱ | ج     | ۳   | ۱ | ۳   | ۹          | ۶                  | ۱/۴۲ |  |

جدول ۴-۵ بررسی خروجی نرم افزار برای الگوریتم ۱ تا ۵ ریسمان برای تصاویری با وضوح ۰.۲۴۰ x ۰.۳۲۰

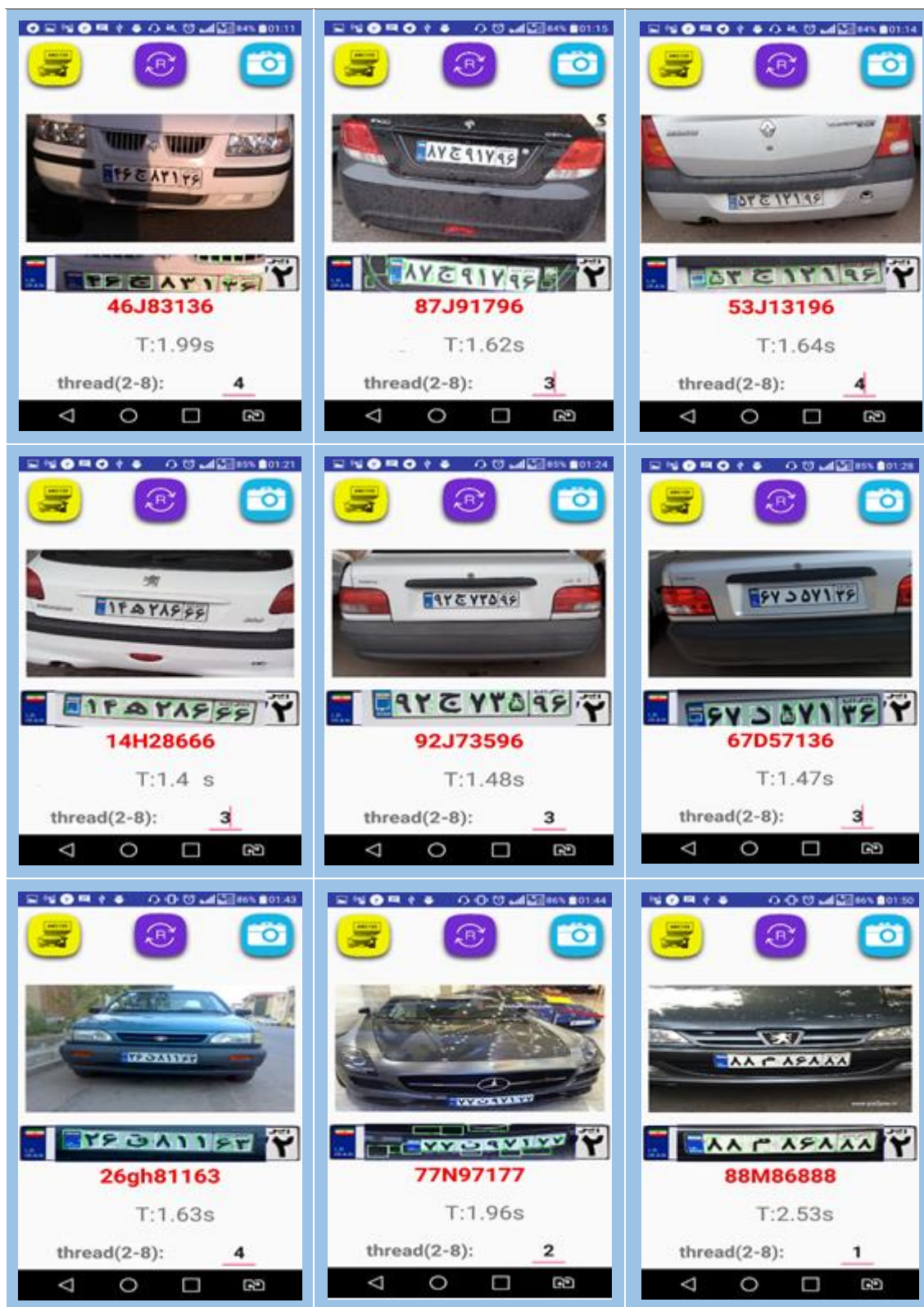
| سایز تصویر                  | تعداد ریسمانها | شماره ردیف و عکس | شماره پلاک |      |     |     |     |     |     | زمان/ثابت | تصاویر استفاده شده  |   |
|-----------------------------|----------------|------------------|------------|------|-----|-----|-----|-----|-----|-----------|---|---|
|                             |                |                  | عدد        | حروف | عدد | عدد | عدد | عدد | عدد |           |   |   |
| ۳۲۰ x ۲۴۰<br>فاصله<br>۱ متر | ۱              | ۱                | ۶          | ۶    | ط   | ۸   | ۳   | ۱   | ۱   | ۲         | ۰/۷۳  |    |
|                             | ۲              |                  | ۶          | ۶    | ط   | ۸   | ۳   | ۱   | ۱   | ۲         | ۰/۶۰  |   |
|                             | ۳              | ۰۱۱۵             | ۶          | ۶    | ط   | ۸   | ۳   | ۱   | ۱   | ۲         | ۰/۵۶  |   |
|                             | ۱              | ۲                | ۷          | ۵    | ق   | ۹   | ۳   | ۱   | ۵   | ۵         | ۰/۷۸  |    |
|                             | ۲              |                  | ۷          | ۵    | ق   | ۹   | ۳   | ۱   | ۵   | ۵         | ۰/۷۲  |   |
|                             | ۳              | ۰۱۱۳             | ۷          | ۵    | ق   | ۹   | ۳   | ۱   | ۵   | ۵         | ۰/۷۲  |   |
|                             | ۱              | ۳                | ۱          | ۸    | ص   | ۲   | ۱   | ۵   | ۹   | ۶         | ۱/۵۸  |    |
|                             | ۲              |                  | ۱          | ۸    | ص   | ۲   | ۱   | ۵   | -   | -         | ۱/۱۸  |   |
|                             | ۳              | ۰۰۹۹             | ۱          | ۸    | ص   | ۲   | ۱   | ۵   | ۹   | ۶         | ۱/۰۳  |   |
|                             | ۱              | ۴                | ۸          | ۷    | ج   | ۹   | ۱   | ۷   | ۹   | ۶         | ۰/۸۱  |    |
|                             | ۲              |                  | ۸          | ۷    | ج   | ۹   | ۱   | ۷   | ۹   | ۶         | ۰/۸۳  |   |
|                             | ۳              | ۰۱۲۰             | ۸          | ۷    | ج   | ۹   | ۱   | ۷   | ۹   | ۶         | ۰/۶۰  |   |
|                             | ۱              | ۵                | ۳          | ۷    | ج   | ۷   | ۷   | ۲   | ۳   | ۶         | ۰/۶۷  |    |
|                             | ۲              |                  | ۳          | ۷    | ج   | ۷   | ۷   | ۳   | ۳   | ۶         | ۰/۵۴  |   |
|                             | ۳              | ۰۱۱۴             | ۳          | ۷    | ج   | ۷   | ۷   | ۳   | ۳   | ۶         | ۰/۶۸  |   |
|                             | ۱              | ۶                | ۳          | ۳    | ب   | ۶   | ۷   | ۶   | ۹   | ۶         | ۰/۷۵  |  |
|                             | ۲              |                  | ۳          | ۳    | ب   | ۶   | ۷   | ۶   | ۹   | ۶         | ۰/۵۸  |   |
|                             | ۳              | ۰۱۱۲             | ۳          | ۳    | ب   | ۶   | ۷   | ۶   | ۹   | ۶         | ۰/۶۷  |   |
|                             | ۱              | ۷                | ۶          | ۷    | د   | ۵   | ۷   | ۱   | ۳   | ۶         | ۰/۷۵  |  |
|                             | ۲              |                  | ۶          | ۷    | د   | ۵   | ۷   | ۱   | ۳   | ۶         | ۰/۶۵  |   |
|                             | ۳              | ۰۱۰۲             | ۶          | ۷    | د   | ۵   | ۷   | ۱   | ۳   | ۶         | ۰/۶۳  |   |
|                             | ۱              | ۸                | ۱          | ۸    | ص   | ۲   | ۱   | ۵   | ۹   | ۶         | ۰/۷۱  |  |
|                             | ۲              |                  | -          | -    | -   | -   | -   | -   | -   | -         | ۰/۶۲  |   |
|                             | ۳              | ۰۰۹۲             | ۱          | ۸    | ص   | ۲   | ۱   | ۵   | ۹   | ۶         | ۰/۴۹  |   |
|                             | ۱              | ۹                | ۴          | ۱    | ج   | ۸   | ۱   | ۹   | ۹   | ۶         | ۰/۸۸  |  |
|                             | ۲              |                  | ۴          | ۱    | ج   | ۸   | ۱   | ۹   | ۹   | ۶         | ۰/۷۲  |   |
|                             | ۳              | ۰۱۱۰             | ۴          | ۱    | ج   | ۸   | ۱   | ۹   | ۹   | ۶         | ۰/۵۴  |   |
|                             | ۱              | ۱۰               | ۴          | ۹    | ص   | ۳   | ۹   | ۷   | -   | ۶         | ۰/۷۴  |  |
|                             | ۲              |                  | ۴          | ۹    | ص   | ۳   | ۹   | ۷   | ۹   | ۶         | ۰/۶۶  |   |
|                             | ۳              | ۰۰۹۵             | ۴          | ۹    | ص   | ۳   | ۹   | ۷   | ۹   | ۶         | ۰/۷۱  |   |
|                             | ۱              | ۱۱               | ۵          | ۷    | ل   | ۹   | ۵   | ۶   | ۱   | ۳         | ۰/۶۳  |  |
|                             | ۲              |                  | ۵          | ۷    | ل   | ۹   | ۵   | ۶   | ۱   | ۳         | ۰/۶۰  |   |
| ۳                           | ۰۰۹۴           | ۵                | ۷          | ل    | ۹   | ۵   | ۶   | ۱   | ۲   | ۰/۶۸      |   |   |
| ۱                           | ۱۲             | ۴                | ۷          | ج    | ۳   | ۴   | ۸   | ۹   | ۶   | ۰/۷۵      |  |   |
| ۲                           |                | ۴                | ۷          | ج    | ۳   | ۴   | ۸   | ۹   | ۶   | ۰/۵۸      |   |   |
| ۳                           | ۰۱۱۱           | ۴                | ۷          | ج    | ۳   | ۴   | ۸   | ۹   | ۶   | ۰/۶۶      |   |   |
| ۱                           | ۱۳             | ۱                | ۴          | ص    | ۶   | ۵   | ۵   | ۹   | ۶   | ۰/۷۸      |  |   |
| ۲                           |                | ۱                | ۴          | ص    | ۶   | ۵   | ۵   | ۹   | ۶   | ۰/۶۸      |   |   |
| ۳                           | ۰۱۱۶           | ۱                | ۴          | ص    | ۶   | ۵   | ۵   | ۹   | ۶   | ۰/۶۹      |   |   |



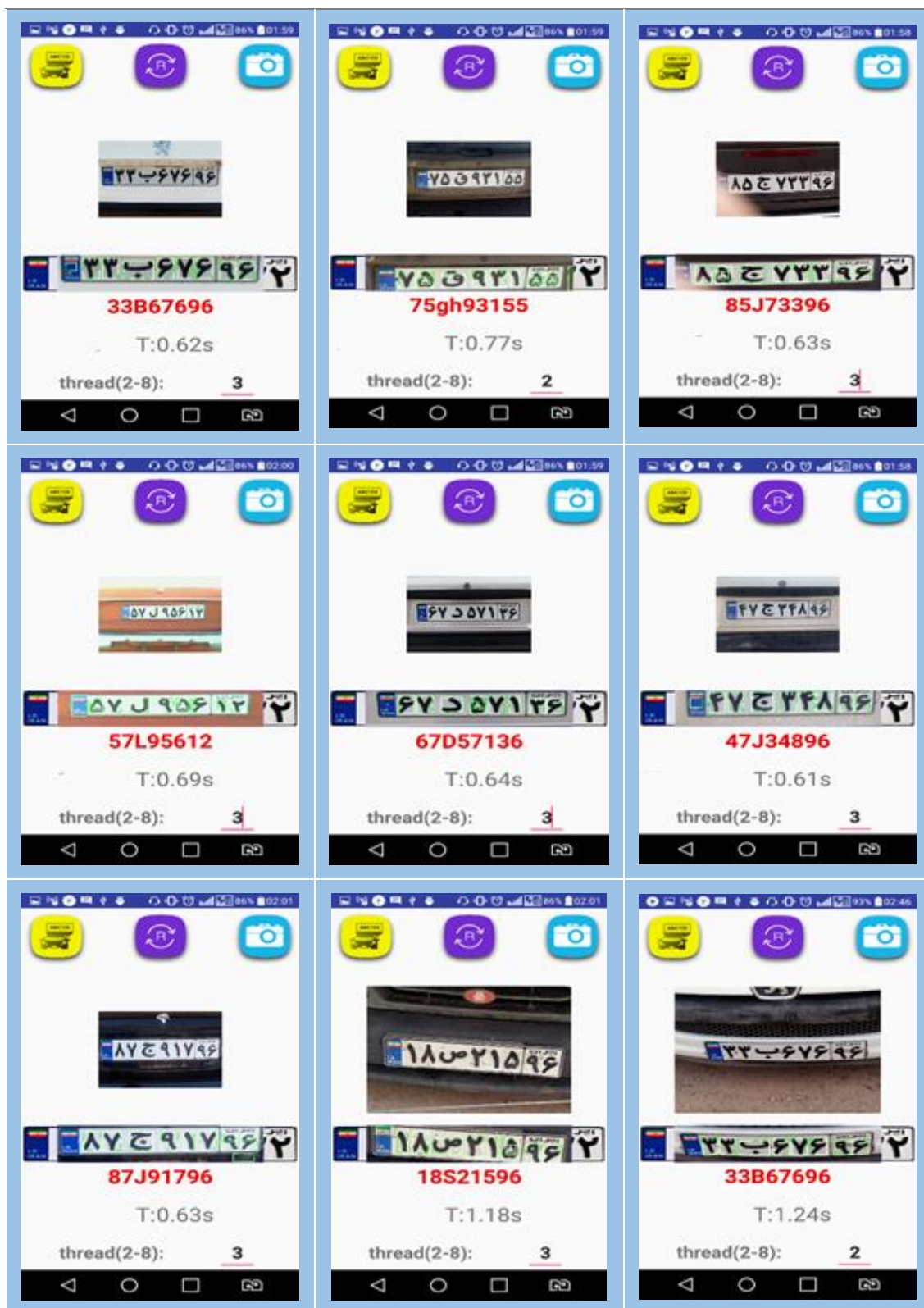
شکل ۴-۲ خروجی برنامه برای تصاویری با وضوح  $1080 \times 1920$  در فاصله ۴ تا ۸ متری از دوربین



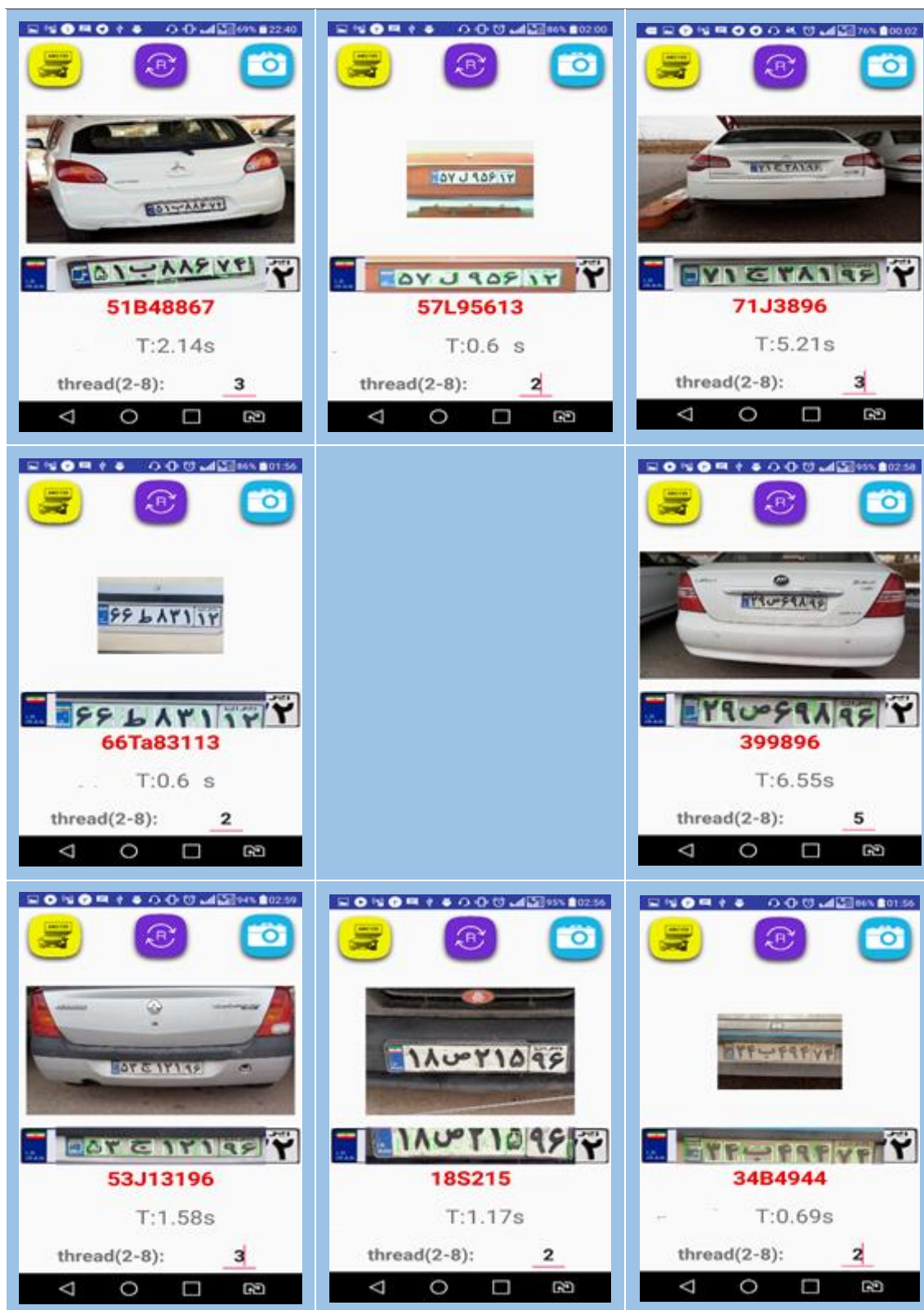
شکل ۳-۴ خروجی برنامه برای تصاویری با وضوح  $1920 \times 1080$  در فاصله ۲ تا ۴ متری از دوربین



شکل ۴-۴ خروجی برنامه برای تصاویری با وضوح  $۹۲۰ \times ۵۴۰$  در فاصله ۱ تا ۲ متری از دوربین



شکل ۴-۵ خروجی برنامه برای تصاویری با وضوح  $240 \times 320$  در فاصله ۱ متری از دوربین



شکل ۴-۶ خروجی‌های نادرست





# فصل پنجم

## جمع بندی و پیشنهادها

## ۱-۵ جمع‌بندی

در این پایان‌نامه تلاش کردیم تا سامانه تشخیص پلاک خودرو را بر روی گوشی هوشمند مبتنی بر سیستم‌عامل اندروید پیاده‌سازی کنیم. هرچند در اوایل راه، با مشکلات زیادی از جمله تطبیق نیافتن کتابخانه OpenCV-android-sdk با برخی از نرم‌افزارهای برنامه‌نویسی اندروید مانند QtC++ و Eclipse و نیز خطاهای متعدد این نرم‌افزارها روبرو شدیم. اما سرانجام توانستیم این سامانه را توسط نرم‌افزار AndroidStudio به سرانجام برسانیم. در پیوست الف روش ایجاد پروژه و اضافه کردن کتابخانه OpenCV آمده است.

روند اجرای الگوریتم بدین طریق انجام پذیرفت. ابتدا با استفاده از فیلتر میانه نویزهای احتمالی تصویر و همچنین لبه‌های ایجادشده توسط ریگ‌های کف خیابان را برطرف کردیم. سپس با استفاده از لبه یاب عمودی سوبل، لبه‌های عمودی تصویر را آشکار کردیم و با اعمال افکنش افقی محل احتمالی پلاک را کشف نمودیم. در مرحله بعدی با اعمال عملگر گسترش لبه‌های انتخاب‌شده در مرحله قبل را به هم متصل می‌کنیم تا یک ناحیه محصور به محدوده لبه‌ها ایجاد کنیم. پس از آن با اعمال الگوریتم واکاوی اجزای متصل نامزدهای احتمالی محل پلاک را از میان ناحیه‌های ایجادشده جدا می‌کنیم. حال اگر حروف پلاک در نامزدهای مورد نظر توسط شبکه عصبی تشخیص داده شود آن نامزد، محل پلاک است در غیر این صورت نامزد بعدی مورد ارزیابی قرار می‌گیرد.

پردازش‌های انجام‌شده بر روی سه مدل از تصاویر، با دقت "۱۰۸۰×۱۹۲۰"، "۵۴۰×۹۲۰"

و "۲۴۰ × ۳۲۰" صورت پذیرفت. فاصله دوربین تا خودرو هنگام عکسبرداری به ترتیب ۸-۴،۴-۲ و ۱ متر بوده است.

برای این کاربرد خاص بلادرنگ بودن اهمیت بسزایی ندارند اما سعی کردیم تا الگوریتم سریعی را ارائه کنیم. از طرف دیگر سرعت و قدرت پردازنده‌های تلفن هوشمند در مقایسه با رایانه‌های رومیزی و صنعتی بسیار پایین تر است. بنابراین با بهینه‌سازی الگوریتم و استفاده از پردازش موازی به روش الگوریتم چند ریسمانه سرعت را تا حد قابل قبولی افزایش دادیم. سرعت با استفاده از این روش ۳۰ تا ۶۰ درصد افزایش پیدا کرد. همچنین درصد تشخیص دقیق محل پلاک خودرو در این الگوریتم ۹۷/۸٪ و درصد تشخیص ارقام پلاک ۹۶/۱٪ در صد به دست آمد.

روش پیشنهادی در تصاویری با تراکم پیکسل کم و فاصله‌های کوتاه برای کاربردهایی مانند ثبت پلاک پارکبان عملکرد و سرعت مناسبی دارد. از آن گذشته سرعت این نرم‌افزار با نرم‌افزارهای مشابه عرضه شده در کشورهای دیگر نیز برابری می‌کند.

## ۲-۵ پیشنهادها

پیشنهاد می‌شود در کارهای آینده از نمونه‌های بیشتری برای داده‌های آموزش استفاده شود همچنین. پیاده‌سازی با استفاده از الگوریتم چند ریسمانی به روش دیگری نیز اجرا شود. استفاده از

کتابخانه‌های پردازش موازی مانند OpenCL، کاهش کادر عکس برداری از تمام خودرو به محدوده پلاک

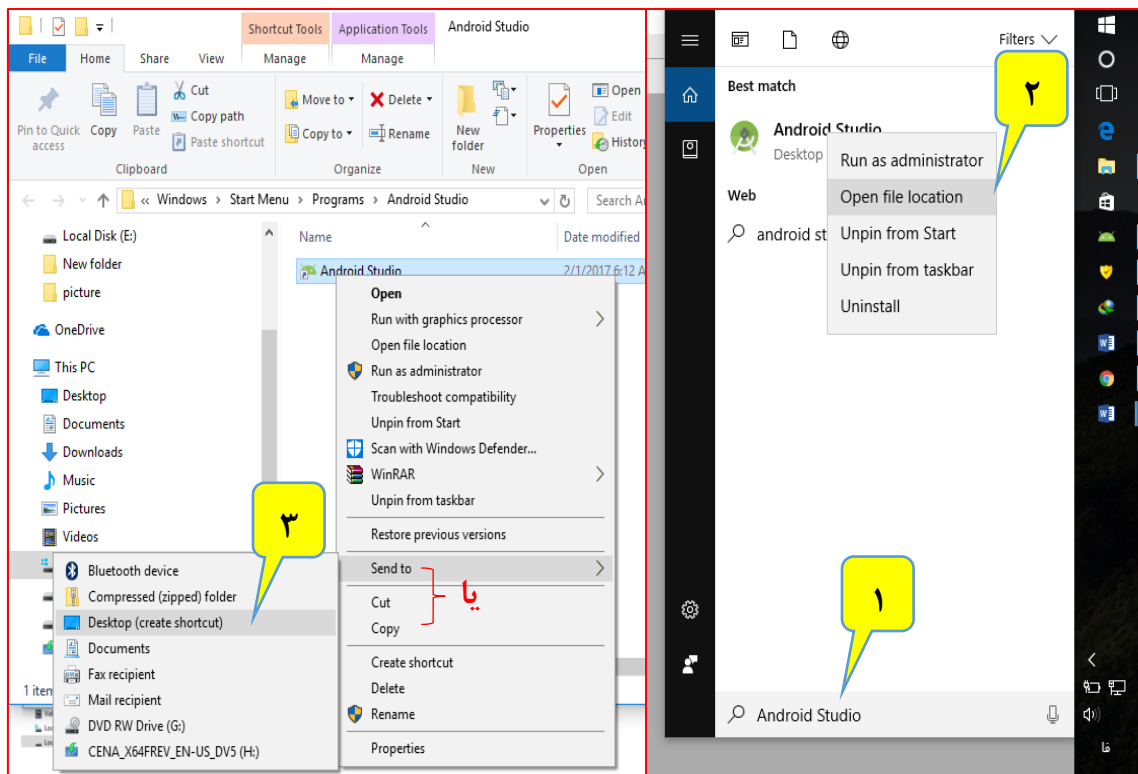
به منظور استفاده در سامانه‌های پارکبان و تشخیص برخط پلاک با استفاده از ویدئو پیشنهاد می‌شود.

# پوست الف

**نحوه ایجاد یک پروژه و اضافه کردن کتابخانه Opencv در  
Android Studio**

## الف-۱ ایجاد یک پروژه در android studio :

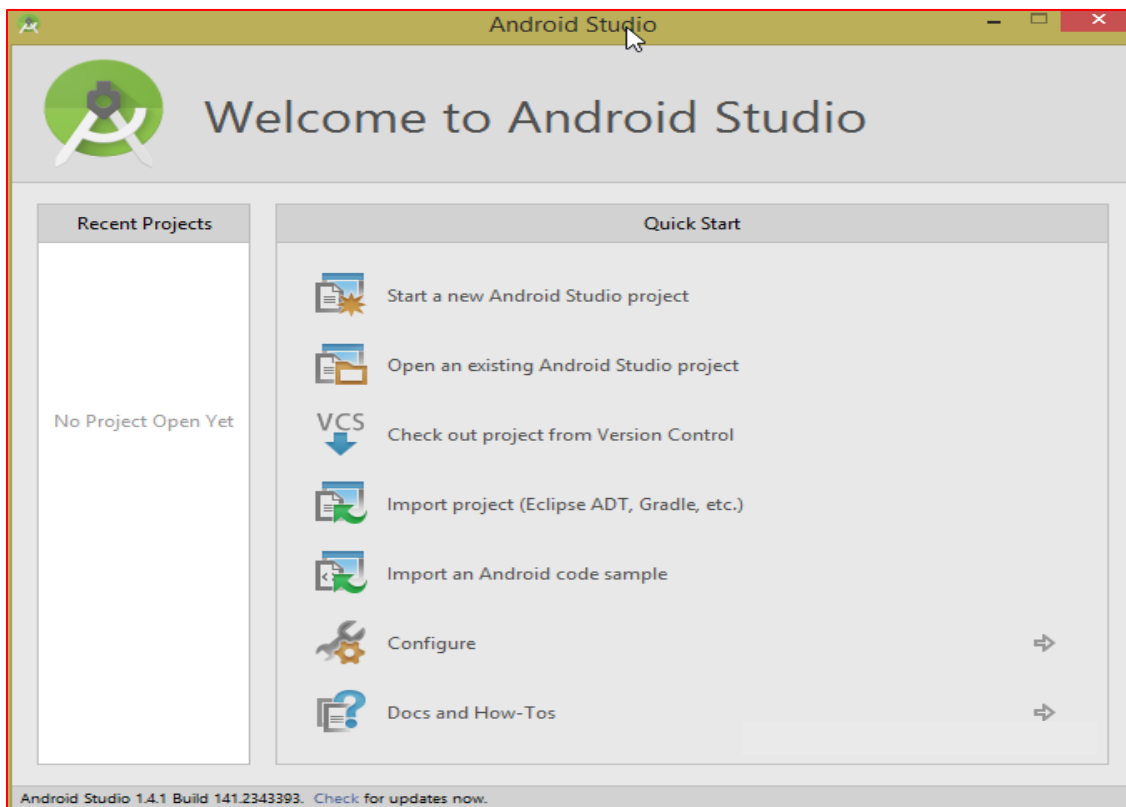
در این آموزش می‌خواهیم با محیط اندروید استودیو آشنا شده و اولین پروژه اندروید را ایجاد کنیم. اگر بعد از نصب اندروید استودیو، در ویندوز ۱۰ آیکون آن روی دسکتاپ اضافه نشده باشد، مطابق شکل الف-۱ در قسمت جستجوی نرم‌افزارها نام Android Studio را تایپ می‌کنیم و پس از مشاهده آن در لیست جستجو، روی آیکون راست کلیک می‌کنیم و سپس با انتخاب گزینه Open File Location وارد مسیری می‌شویم که فایل میانبر برنامه در آن موجود است. فایل میانبر را کپی می‌کنیم و روی دسکتاپ یا هر جای دیگری که مایل هستیم از آنجا دسترسی به محیط برنامه‌نویسی داشته باشیم، جایگذاری می‌کنیم و یا گزینه Send to را انتخاب می‌کنیم.



شکل ۶-۱ اضافه کردن آیکن اندروید بر روی میز کار

نرم افزار android studio را اجرا می کنیم. بارگذاری کامل محیط برنامه نویسی چند ثانیه طول

خواهد کشد بنابراین منتظر می مانیم تا صفحه ای مانند شکل الف -۲ نمایش داده شود:



شکل ۶-۲ ایجاد اولین پروژه

در شکل الف -۲ چند گزینه پیش روی ما قرار دارد.

۱. Start a new Android Studio Project : این گزینه برای ساخت پروژه اندروید جدید استفاده

می شود.

۲. Open an existing Android Studio Project : برای باز کردن پروژه هایی هست که قبلاً در محیط

اندروید استودیو ساخته شده.

۳. Import Project : برای وارد کردن پروژه هایی هست که قبلاً در محیط هایی مانند Eclipse توسعه

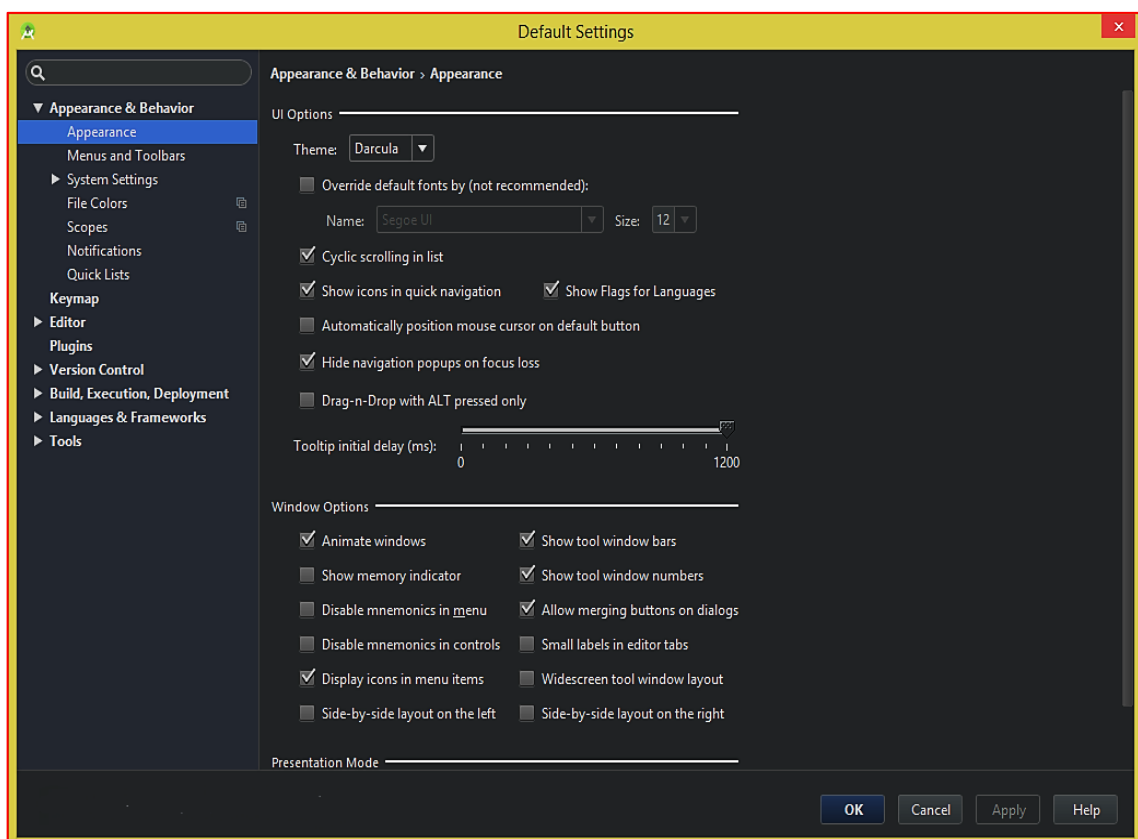


داده شده است.

۴. Import an Android code sample : برای وارد کردن کدهای نمونه اندروید استفاده می‌شود.

۵. Configure : با استفاده از این گزینه و قبل از باز کردن یک پروژه جدید می‌توان به تنظیمات محیط برنامه‌نویسی دسترسی داشت.

بعد از انتخاب Configure و سپس انتخاب گزینه Settings ، می‌توانیم به تنظیمات اصلی دسترسی داشته باشیم. به عنوان مثال مطابق شکل الف -۳ بنا به سلیقه خود می‌توانیم در شاخه Appearance & Behavior و زیرشاخه Appearance، در قسمت Themes، گزینه Darcula را انتخاب کنید که تم محیط اندروید استودیو را از رنگ روشن به رنگ تیره تغییر می‌دهد.



شکل ۳-۶ تنظیمات محیط کار android studio

بعد از ذخیره تنظیمات به صفحه‌ی مطابق شکل الف-۲ مراجعه می‌کنیم. با انتخاب گزینه Start

a new Android Studio Project وارد مرحله بعدی ساخت پروژه می‌شویم. در این مرحله اطلاعات

اصلی برنامه‌ای که بنا داریم بسازیم را مطابق شکل الف-۴ وارد می‌کنیم.

Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.  
By targeting API 15 and later, your app will run on approximately 98.3% of the devices that are active on the Google Play Store.  
[Help me choose](#) Stats load failed. Value may be out of date.

Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

Android Auto

Glass (Not Available)

Minimum SDK:

Previous Next Cancel Finish

شکل الف-۴ پیکربندی تنظیمات پروژه جدید

موضوع Application name مربوط به نام برنامه هست که در واقع کاربر با نصب آن روی دستگاه

مبتنی بر اندروید خود، آن را مشاهده می‌کند. به عنوان مثال می‌توانید نام MyApplication را وارد

کنید.

Company domain : برنامه‌های توسعه‌دهندگان اندروید در فروشگاه برنامه‌ها مانند Google

Play و بازار بر اساس نام بسته آن‌ها تشخیص داده می‌شوند و نام برنامه مهم نیست. دلیل این هست که در هنگام به‌روزرسانی اپلیکیشن‌ها توسط کاربر، تداخل بین چند اپلیکیشن به وجود نیاید و app store بتواند برنامه‌های نصب شده روی دستگاه کاربر را تشخیص بدهد. به عنوان مثال می‌توانید در قسمت Company domain، عبارت android-studio.ir را وارد کنیم. مشاهده می‌کنیم که در قسمت نام بسته، عبارت ir.android\_studio.myapplication نمایش داده می‌شود. استفاده از domain باعث می‌شود که اپلیکیشن‌های توسعه داده شده هر فرد یا شرکت، با نویسه‌های اولیه‌ی متفاوتی ایجاد شوند. اما به این معنا نیست که حتماً باید یک قلمرو حقیقی با مالکیت خودمان داشته باشیم. می‌توانیم یک domain دلخواه وارد کنید. در قسمت Project Location هم مسیر ذخیره فایل‌های پروژه را مشخص می‌کنید و به مرحله بعد می‌رویم. در این مرحله با چند گزینه مواجه می‌شویم. گزینه Phone and Tablet برای توسعه برنامه‌های تلفن همراه و تبلت، Wear برای برنامه‌های گجت‌های پوشیدنی که از سیستم‌عامل Android Wear استفاده می‌کنند (مانند برخی ساعت‌های هوشمند)، TV که برای برنامه‌های تلویزیون‌های هوشمند با سیستم‌عامل اندرویدی بکار می‌رود، Android Auto که برای ساخت برنامه‌ای است که در خودروهایی با سیستم‌عامل اندروید استفاده می‌شود و گزینه نهایی که البته به صورت پیش‌فرض غیرفعال هست، Glass بوده که مربوط به پروژه عینک Google Glass است.

با توجه به اینکه هدف ما ساخت برنامه‌ای برای موبایل و تبلت هست، گزینه اول را انتخاب

می‌کنیم.

بعد از این مرحله، باید Minimum SDK مورد نظرمون را از لیست انتخاب کنیم. واژه Minimum به

معنی حداقل بوده و در اینجا به این معنی است که شما باید تعیین کنید اپلیکیشن مورد نظرتان

پایین‌ترین نسخه اندرویدی را که پشتیبانی خواهد کرد کدام است. به عنوان مثال وقتی شما API 8

(Android 2.2) را انتخاب می‌کنید یعنی اپلیکیشن مورد نظر روی نسخه پایین‌تر از ۲/۲ قابل نصب

نخواهد بود. معمولاً برای اپلیکیشن‌های عادی، نسخه API 10 به عنوان مینیمم استفاده می‌شود و تقریباً

صد درصد موبایل‌های فعلی دارای این نسخه و یا نسخه بالاتر هستند. علت الزامی بودن تعیین مینیمم

هم این است که در هر نسخه جدید از اندروید، امکاناتی اضافه می‌شود که اگر این امر لحاظ نشود،

ممکن است در مواردی برای دارندگان دستگاه‌هایی با نسخه‌های پایین‌تر، مشکلاتی به وجود آید.

توجه داشته باشیم حداقل API ای را که انتخاب می‌کنیم باید SDK آن را قبلاً در محیط SDK

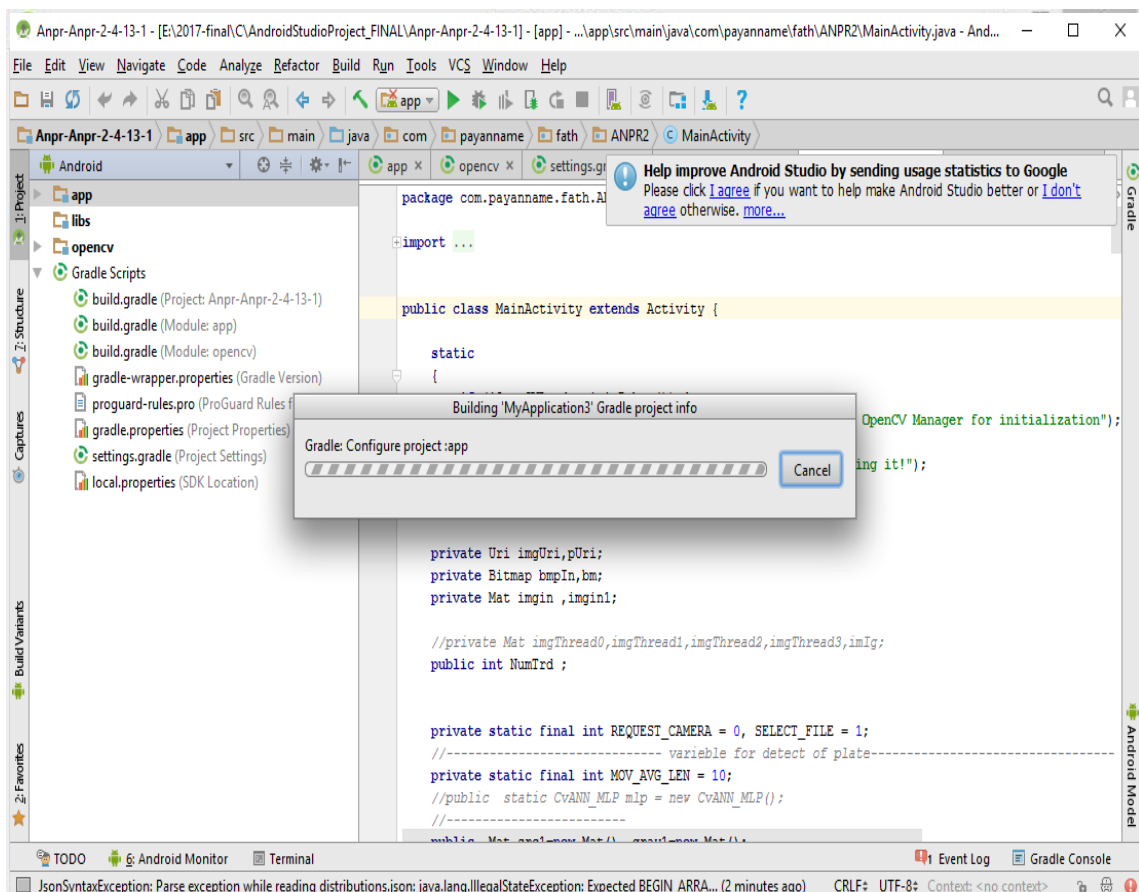
Manager نصب کرده باشید. در مرحله بعد با چند نوع اکتیویته پیش‌فرض روبرو می‌شویم.

Activity ها در واقع صفحاتی هستند که کاربر داخل اپلیکیشن مشاهده می‌کند. هر صفحه یک

Activity است. در این مرحله ما از اکتیویته‌های پیش‌فرض گوگل استفاده نمی‌کنیم و گزینه Blank

Activity را انتخاب می‌کنیم. در مرحله بعد اطلاعات مربوط به اکتیویته از ما خواسته

می‌شود. `ActivityName` که نام اکتیویتهی هست و همان عبارت `MainActivity` می‌تواند مناسب باشد. `LayoutName` را هم تغییر نمی‌دهیم. مورد نهایی، `Menu Resource name` نام منویی هست که در برنامه استفاده خواهد شد. همه این موارد را بدون تغییر رها کرده و با انتخاب گزینه `Finish` منتظر می‌مانیم تا پروژه ساخته شود. فعلاً به همین بسنده می‌کنیم. کدهای جاوای مربوط به این اکتیویتهی که در این مرحله ساختیم در `MainActivity` و عناصر مربوط به طراحی، اجزا و محتوای اکتیویتهی در `Layout` ذخیره می‌شوند.



شکل ۵-۶ بارگذاری پروژه جدید

چند لحظه منتظر می‌مانیم تا ساخته شدن پروژه انجام شود. در حین انجام این فرایند مطابق

شکل الف-۵ با واژه Gradle برخورد می‌کنیم. گریدل یک سازنده سیستم متن‌باز برای زبان جاوا است

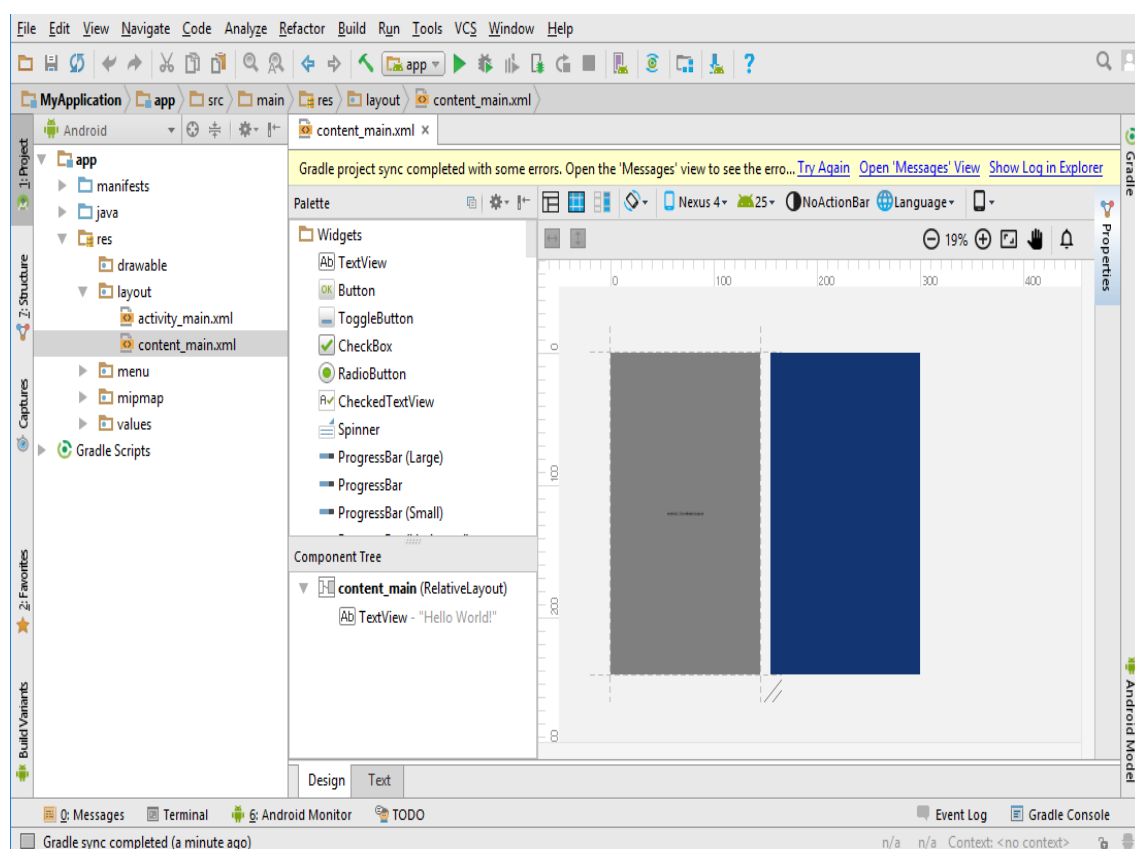
که به زبان Groovy نوشته شده و گوگل در سال ۲۰۱۳ اعلام کرد که از این سازنده سیستم برای

کامپایل کدها در اندروید استودیو استفاده می‌کند. اکنون کار ساخت شدن پروژه تمام شد و وارد محیط

اصلی مدیریت پروژه می‌شویم.

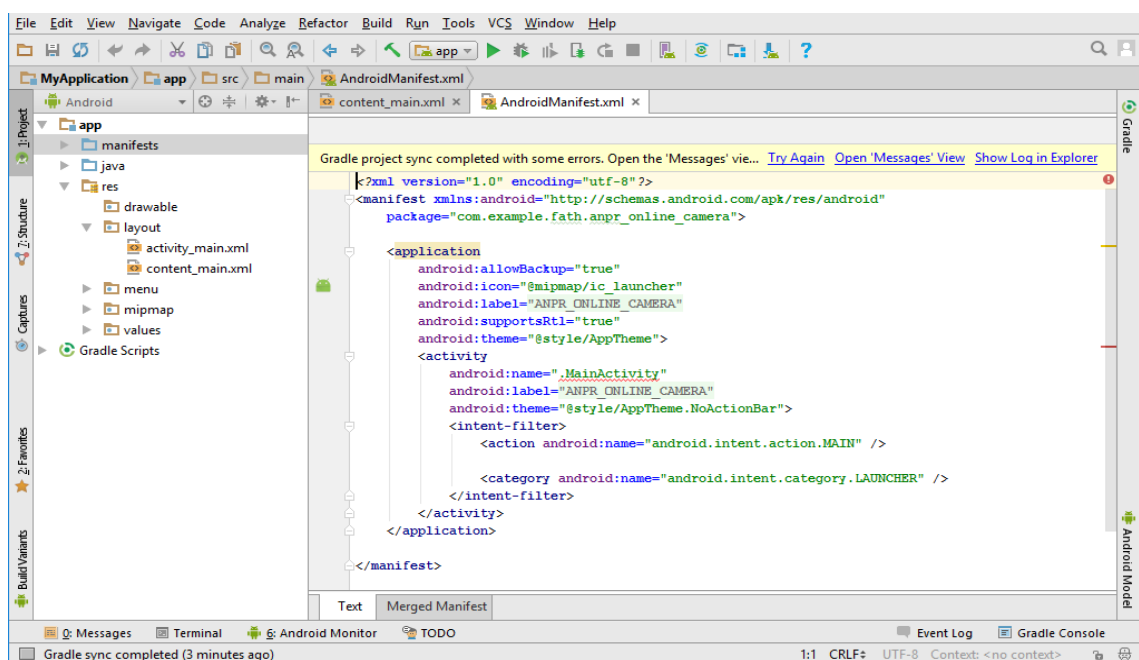
همان‌طور که در تصویر الف-۶ مشاهده می‌کنید، اندروید استودیو به صورت پیش‌فرض دو فایل

MainActivity.java و content\_main.xml را در سمت راست نمایش داده است.



شکل ۶-۶ محیط طراحی ظاهر نرم افزار

همان طور که از پسوند این دو فایل پیداست، مورد اول شامل کدهای جاوای مرتبط با اکتیویتی و مورد دوم مربوط به کدهای xml که برای طراحی و چینش عناصر در داخل اکتیویتی بکار می‌رود. (بهتر است قبل از مطالعه ادامه آموزش، یک مطالعه مختصر و کوتاه در مورد زبان xml و جاوا داشته باشیم. یک برنامه‌نویس و توسعه‌دهنده باید حداقل اطلاعات پایه را در مورد زبان مورد استفاده داشته باشد. با جستجوی عبارت “زبان xml” می‌توانیم به نتایج خوبی در اینترنت برسیم). بخش xml به دو صورت Design و Text قابل نمایش است که در اولی خروجی کدها و دومی خود کدها را نشان می‌دهد. ابتدا ساختار پروژه را بررسی می‌کنیم. مطابق شکل الف-۷ در سمت چپ محیط برنامه‌نویسی، فولدر app را با زیرشاخه‌های manifests، java و res مشاهده می‌کنید. برای باز کردن فایل‌ها روی آن دابل کلیک کنید. اطلاعات ضروری و پایه هر برنامه در داخل فایل Manifest ذخیره می‌شود.



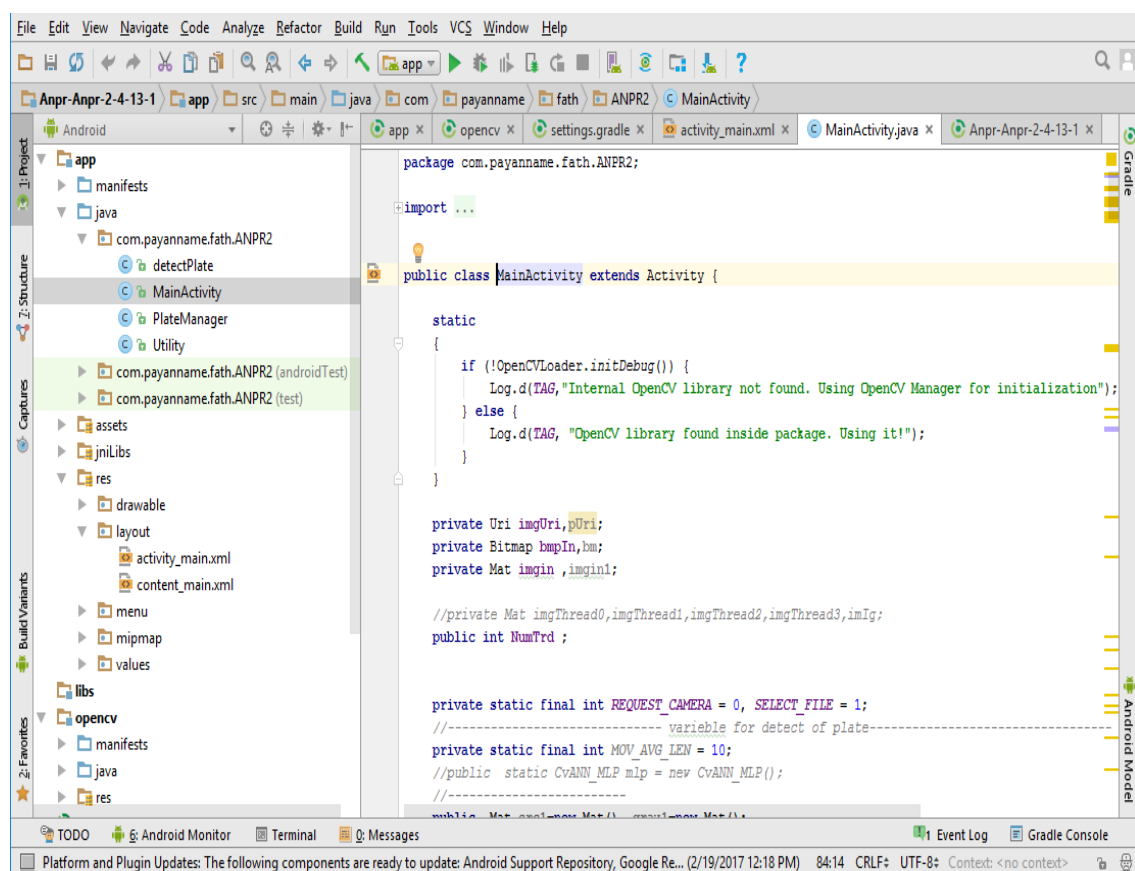
شکل ۶-۷ محیط ثبت و تعیین مجوز

به عنوان مثال اطلاعاتی که در ابتدای کار قابل مشاهده هست، نام بسته‌ای است که موقع ساخت

پروژه آن را تعریف کردیم (خط ۳) ، آیکون اپلیکیشن (خط ۷) ، نام اپلیکیشن (خط ۸) ، اکتیویته

تعریف شده در ابتدای کار است .

Java : اکتیویته‌ها و کلاس‌های مربوطه به جاوا در این پوشه قرار می‌گیرند.



شکل ۶-۸ محیط MainActivity

در تصویر الف-۸، اکتیویته با نام MainActivity را که تعریف کرده بودیم به همراه کلاس‌ها و

محتوای داخل آن مشاهده می‌کنیم.

res : این دایرکتوری شامل منابع مختلفی است.



Drawable : شامل تصاویر مورد استفاده در اپلیکیشن (مانند آیکون)، برخی فایل های xml مانند

bitmapها که از طریق کلاس R.drawable قابل دسترسی هستند.

Layout : که شامل لایه های طراحی واسط کاربری هست (مانند content\_main.xml) که از

طریق کلاس R.layout قابل دسترسی هستند.

Menu : فایل های xml مربوط به منوهای تعبیه شده در اپلیکیشن را شامل می شود که با کلاس

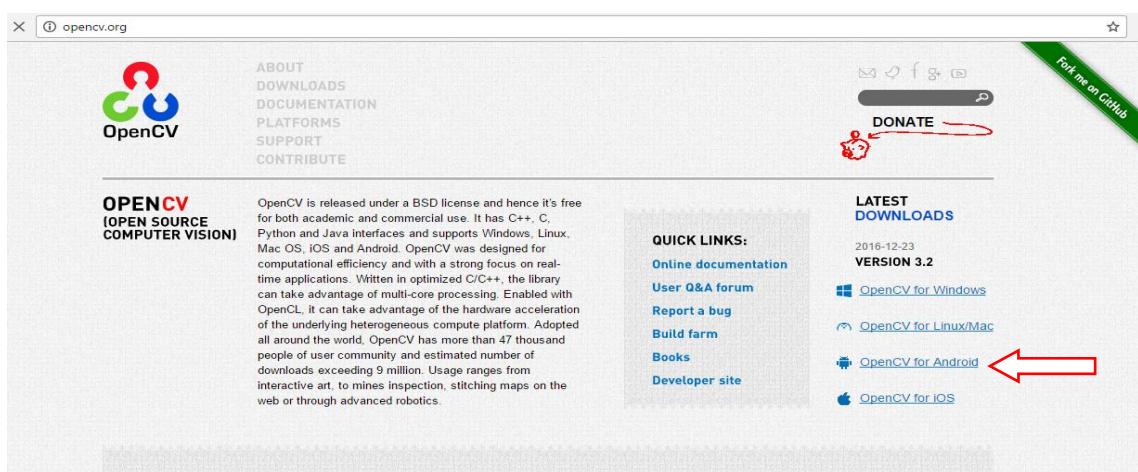
R.menu قابل دسترسی است.

Values : فایل های xml با محتوای آرایه ها (strings.xml) ، ظواهر پروژه (styles.xml) ، رنگ ها

(colors.xml) ، ابعاد (dimens.xml) در این دایرکتوری قرار می گیرند.

## الف-۲ اضافه کردن کتابخانه OpenCV

ابتدا کتابخانه OpenCV را مطابق شکل الف-۹ از وبسایت [www.opencv.org](http://www.opencv.org) بارگیری می نمایم.



شکل ۹-۶ وبسایت opencv

سپس کتابخانه مذکور را از حالت فشرده خارج و در پوشه دلخواه جایگذاری می‌نماییم و طبق

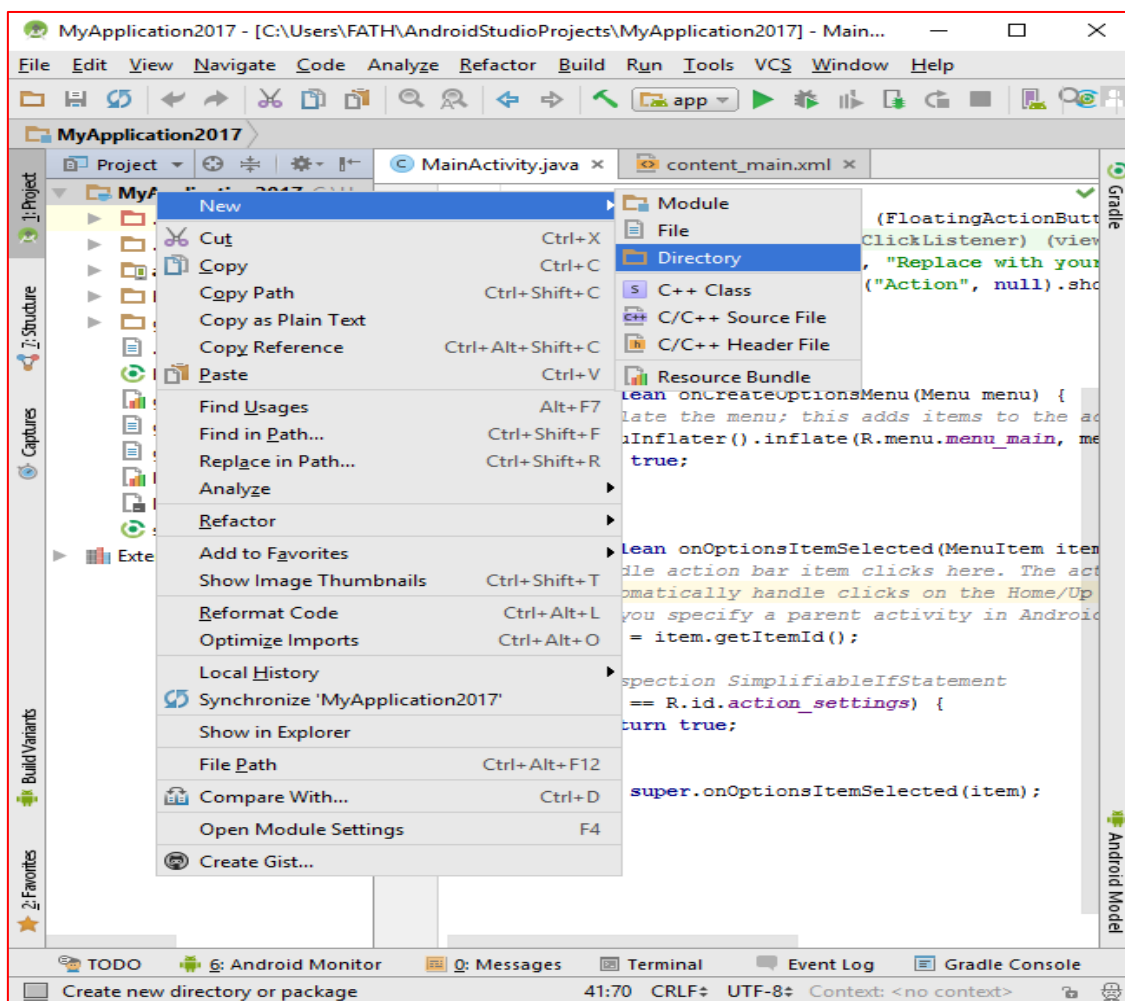
مراحل زیر عمل می‌کنیم.

ابتدا پروژه‌ای را که از قبل ایجاد کرده‌ایم در محیط اندروید استودیو باز می‌کنیم. سپس مطابق مراحل

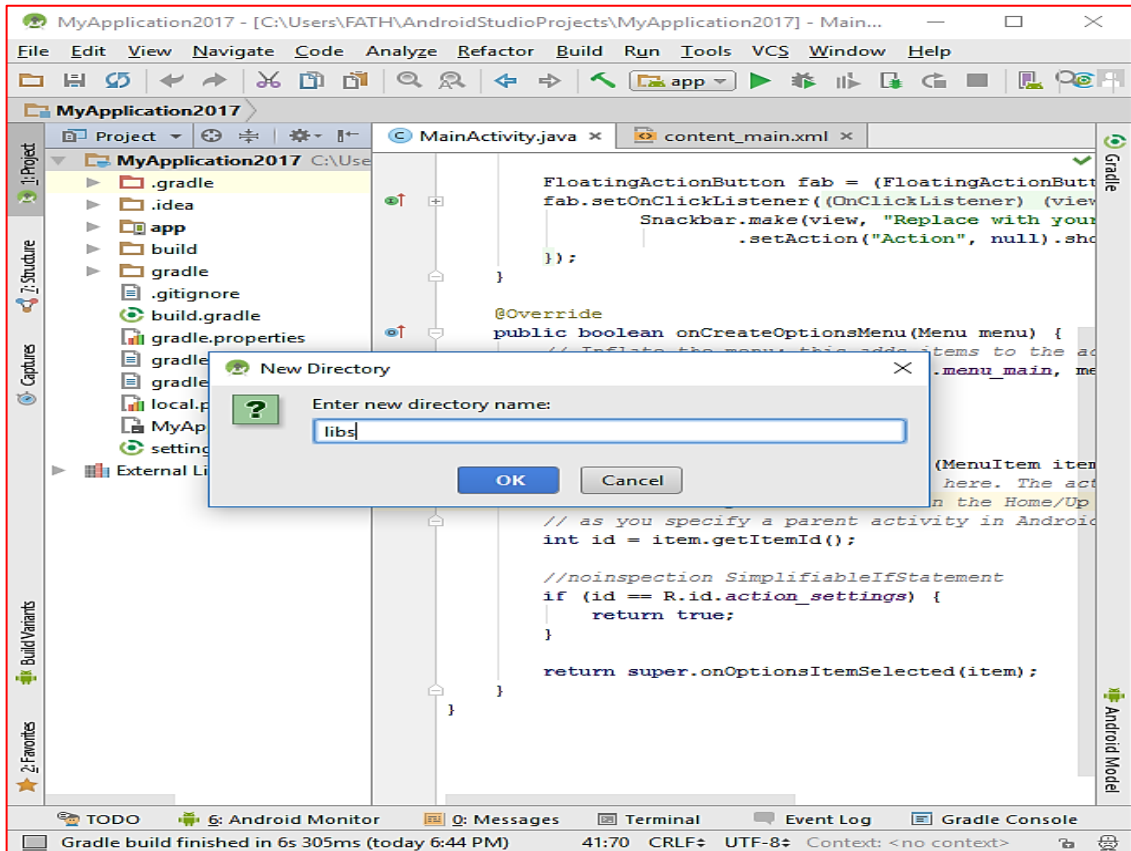
شکل الف-۱۰ تا الف-۱۲ عمل می‌کنیم. اول روی فایل پروژه راست‌کلیک می‌کنیم و سپس یک

دایرکتوری جدید با نام دلخواهی (مثلا libs) ایجاد می‌کنیم. در شکل الف-۱۲ فایل مورد نظر ایجاد

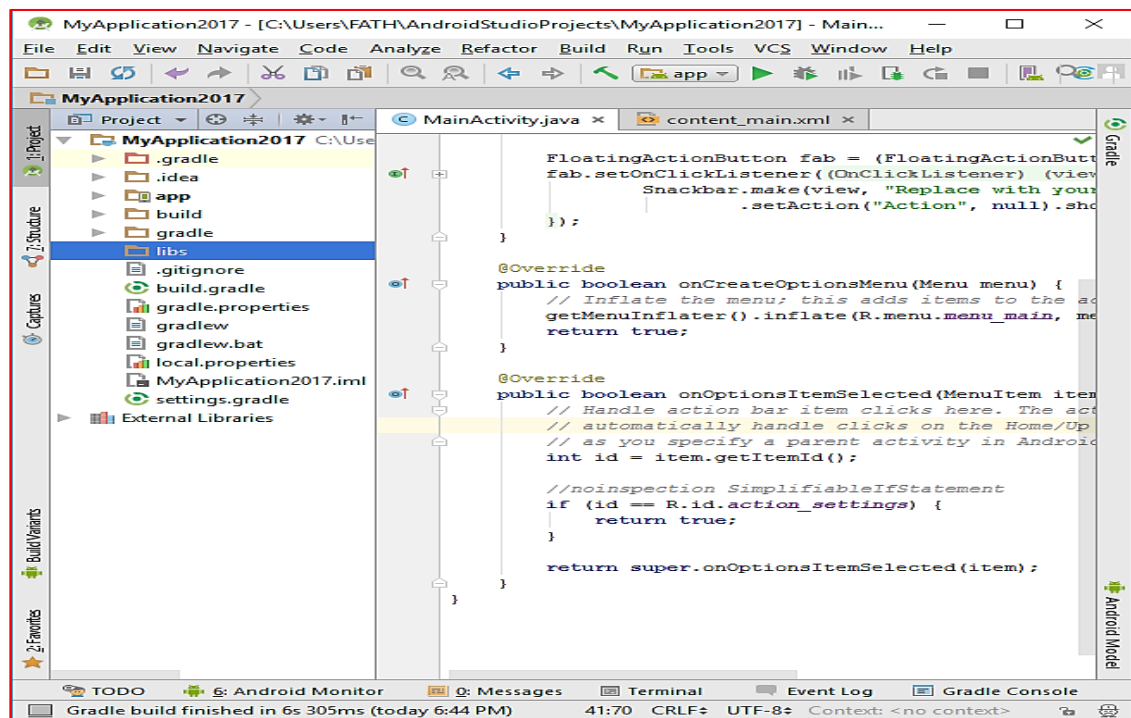
خواهد شد.



شکل ۶-۱۰ ایجاد یک دایرکتوری برای کتابخانه opencv



شکل ۶-۱۱ تعیین نام دایرکتوری کتابخانه opencv



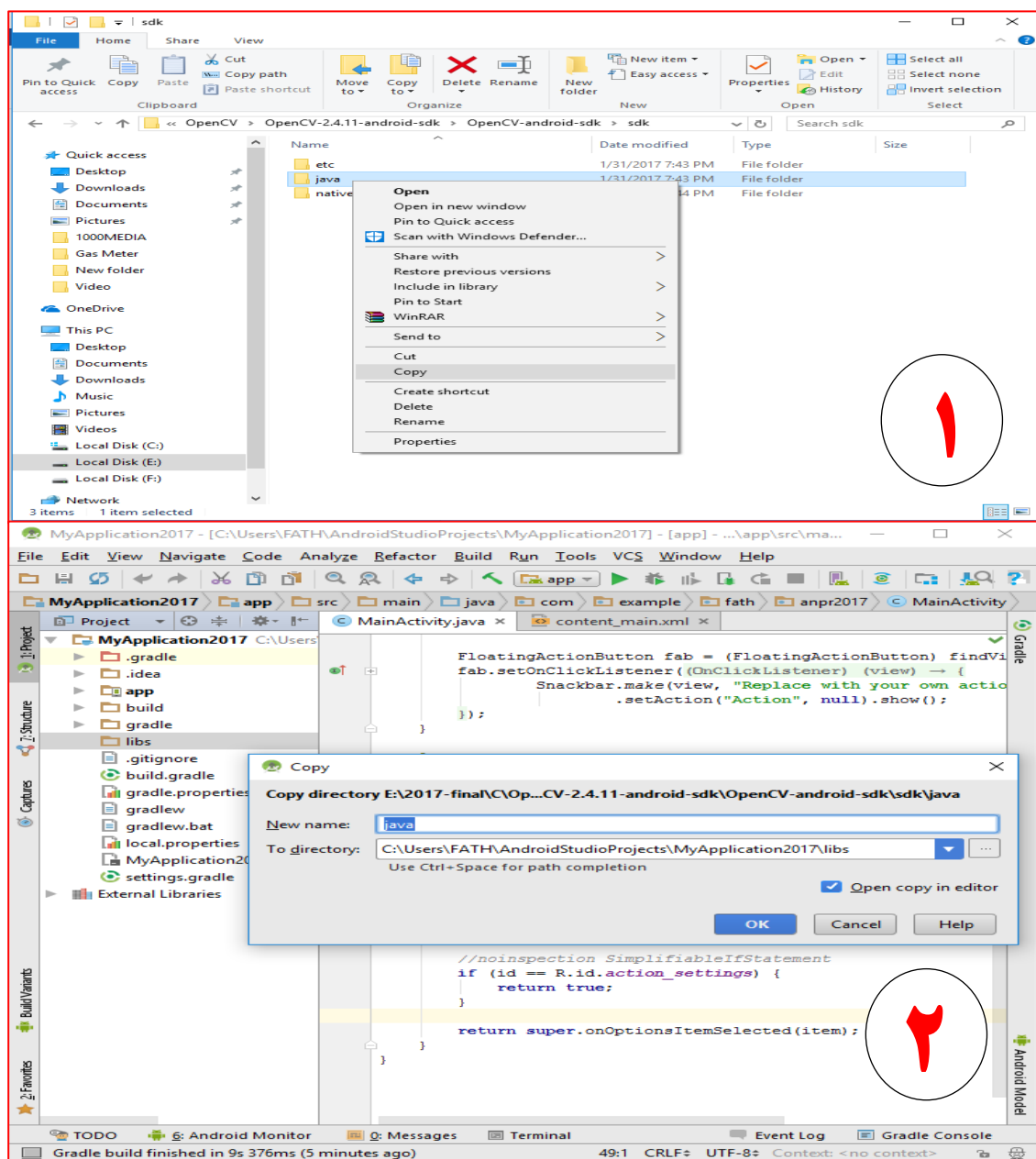
شکل ۶-۱۱ نمای ایجاد شده دایرکتوری کتابخانه opencv

سپس به محل نصب کتابخانه Opencv باز می گردیم و از فایل های که با نام java است یک رونوشت

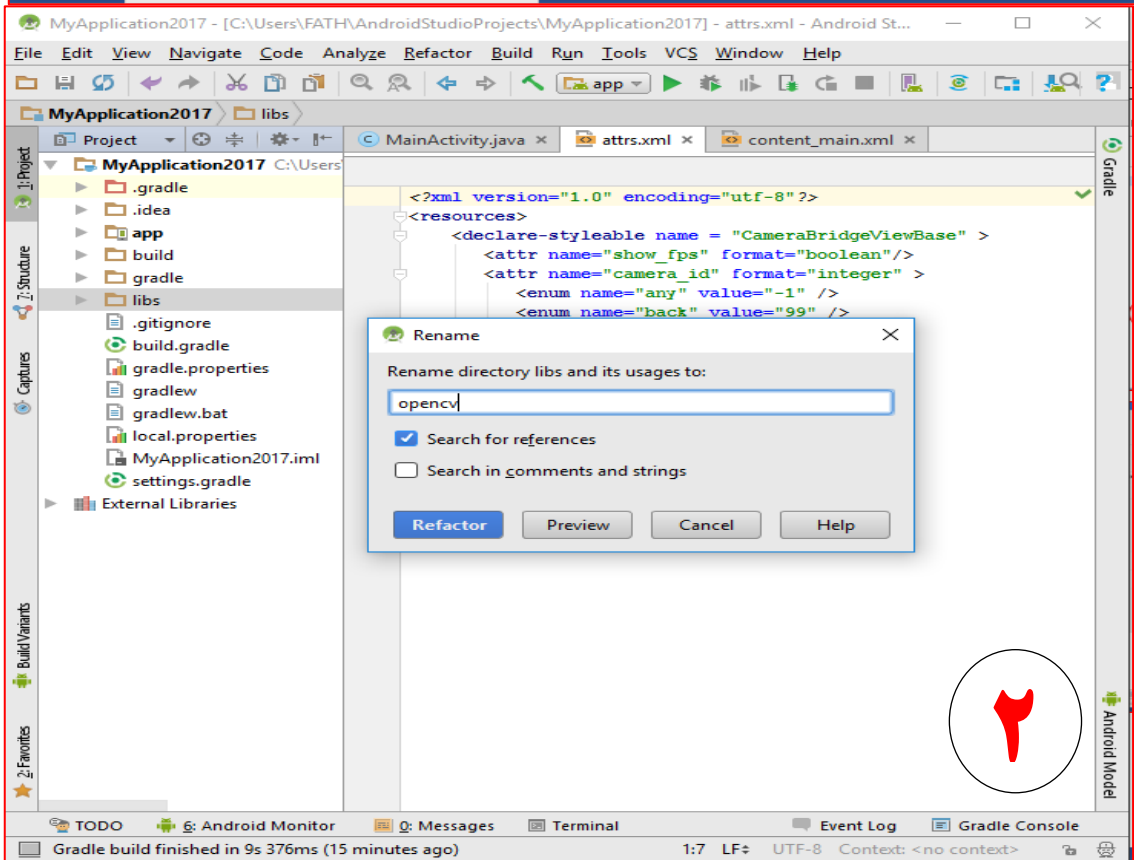
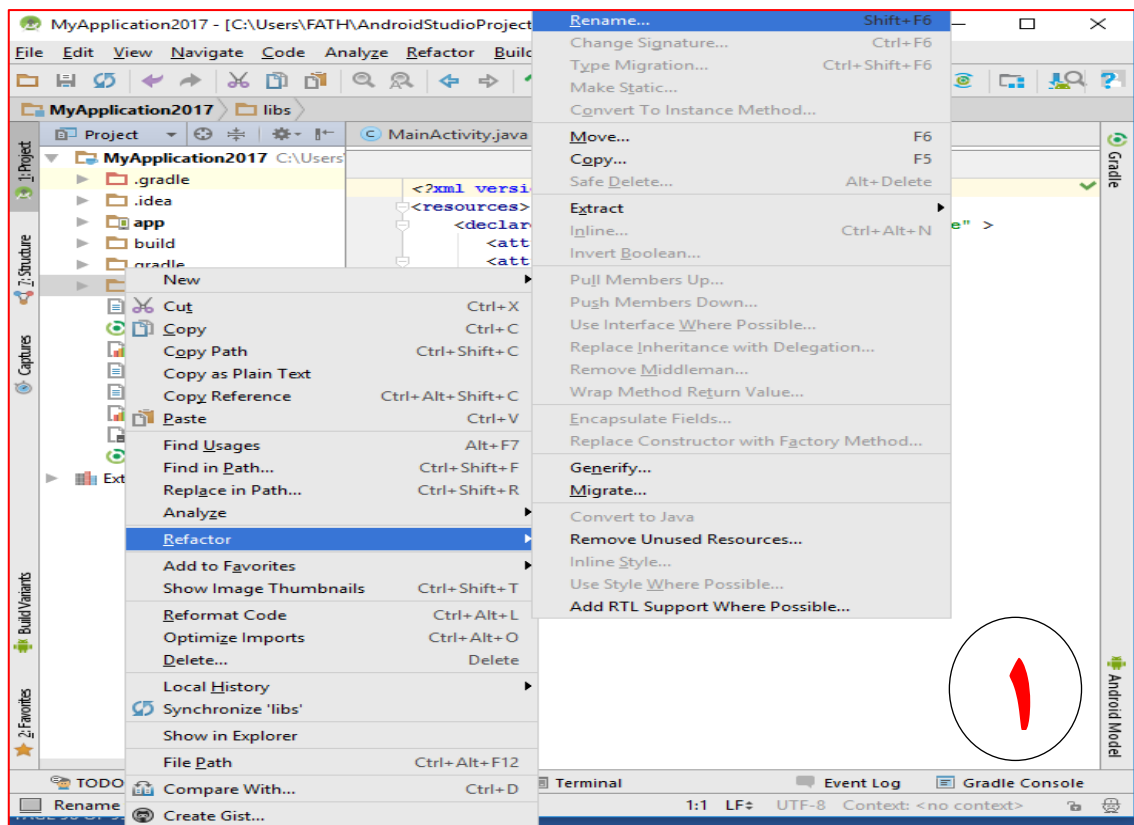
تهیه می کنیم. فایل مورد نظر در این مسیر ( \OpenCV-2.4.11-android-sdk\OpenCV-

android-sdk\sdk) قرار دارد. در مرحله بعدی فایل java را در دایرکتوری libs که در بخش قبلی ایجاد

کرده ایم جایگذاری می کنیم و گزینه Ok را انتخاب می کنیم مطابق مراحل ۱ و ۲ شکل الف-۱۳.



شکل ۶-۱۲ اضافه کردن کتابخانه کد های جاوا opencv



شکل ۶-۱۳ تغییر نام کتابخانه ایجاد شده از java به opencv

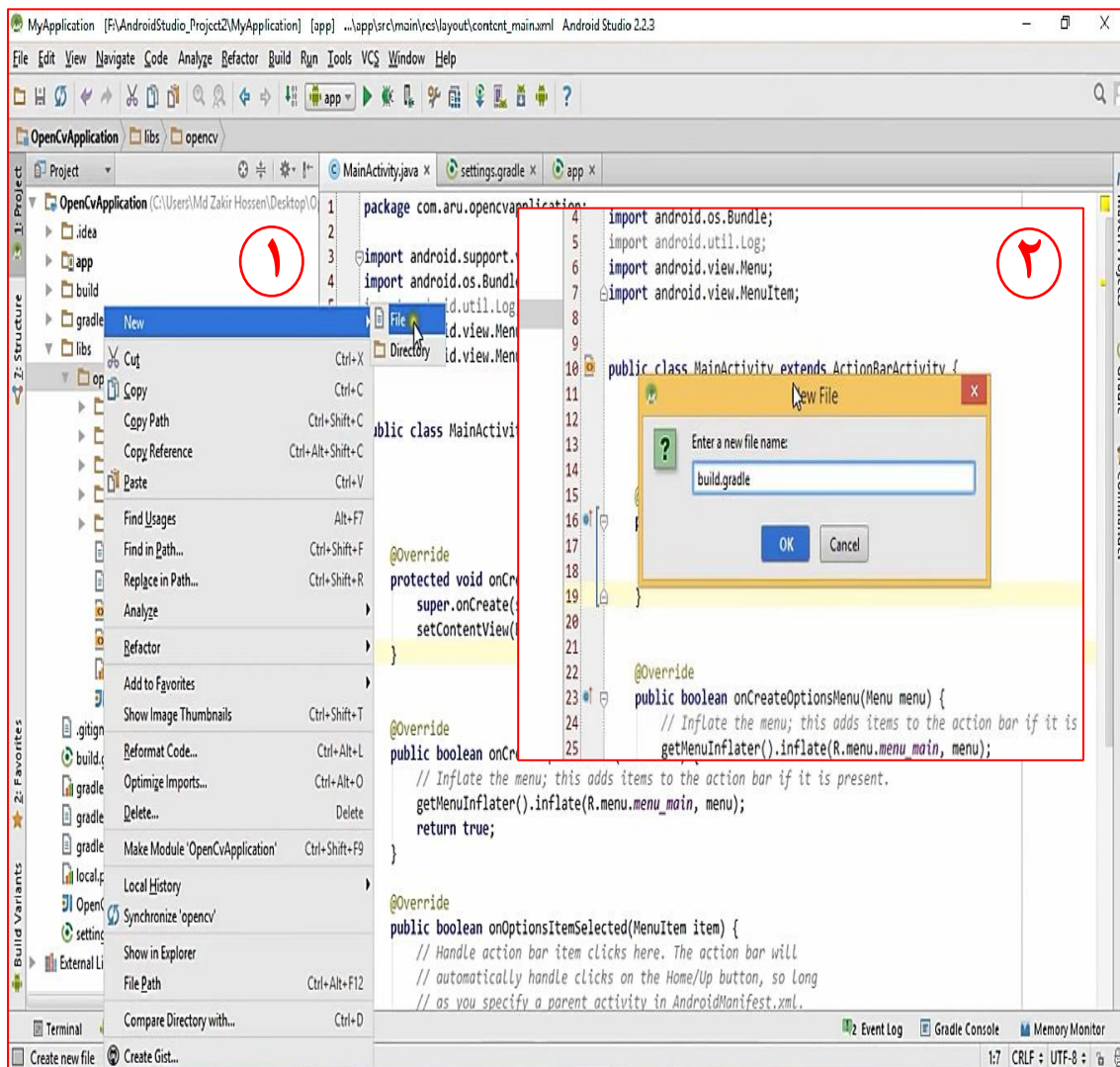
حال مطابق مراحل ۱ و ۲ شکل الف-۱۴ ابتدا روی فایل java راست کلیک کرده و گزینه refactor

و سپس rename را انتخاب کرده و اسم فایل مورد نظر را از java به opencv تغییر می دهیم.

در این بخش بر روی فایل opencv ساخته شده راست کلیک کرده و گزینه New و پس از آن

گزینه File یا پوشه را انتخاب می کنیم. پوشه جدید ایجاد شده را با نام build.gradle نام گذاری می کنیم.

این مراحل طبق قسمت ۱ و ۲ شکل الف-۱۵ انجام می شود.



شکل ۶-۱۴ تعیین یک پوشه با نام build.gradle

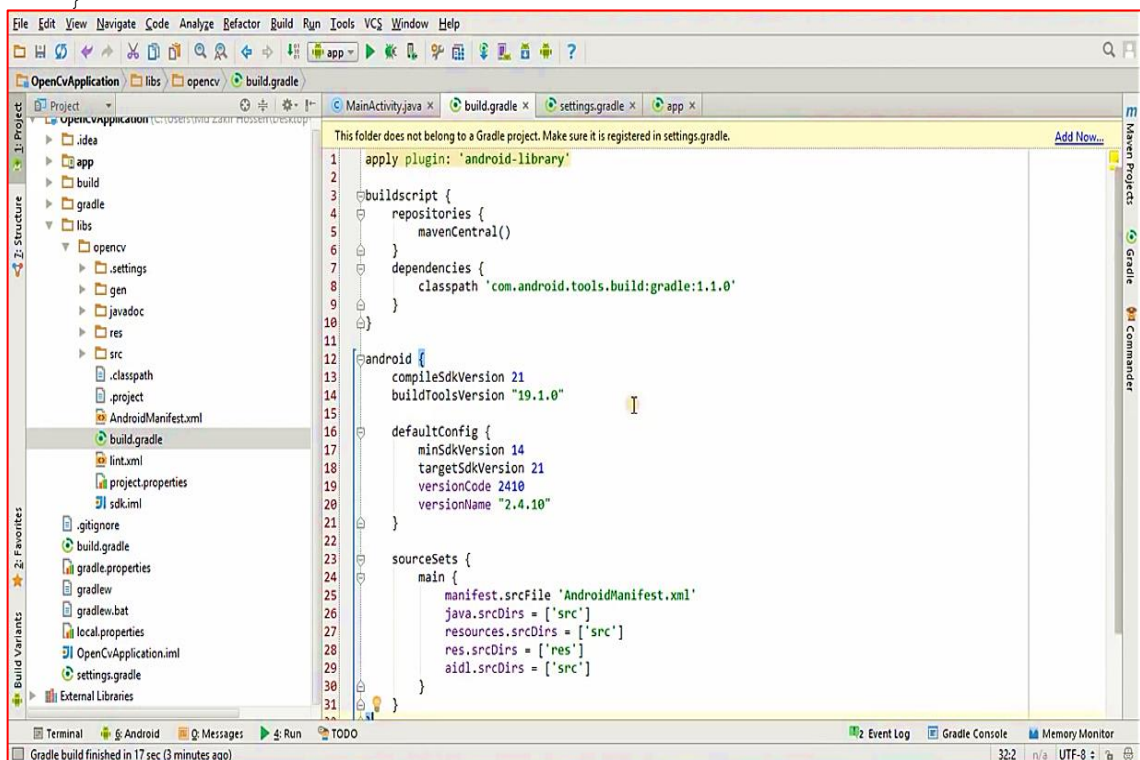
سپس کدهای زیر را مانند تصویر الف-۱۶ در فایل build.gradle جایگذاری می‌کنیم.

```
apply plugin: 'android-library'

buildscript {
    repositories {
        mavenCentral()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.3'
    }
}

android {
    compileSdkVersion 24
    buildToolsVersion "24.0.1"

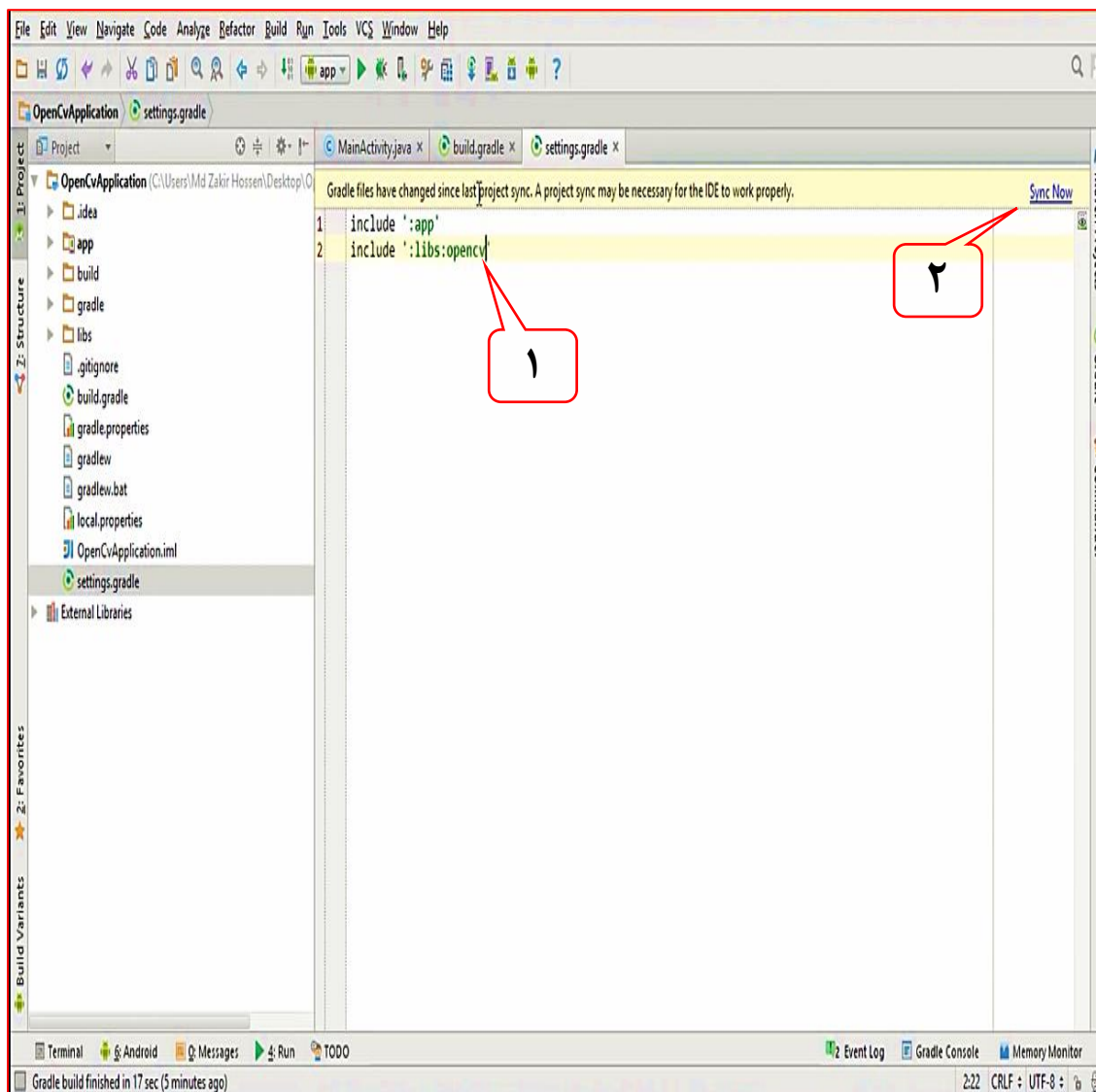
    defaultConfig {
        minSdkVersion 14
        targetSdkVersion 24
        versionCode 3100
        versionName "3.1.0"
    }
    sourceSets{
        main{
            manifest.srcFile 'AndroidManifest.xml'
            java.srcDirs = ['src']
            resources.srcDirs = ['src']
            res.srcDirs = ['res']
            aidl.srcDirs = ['src']
        }
    }
}
```



شکل ۶-۱۵ نمای اضافه کردن کدها به پوشه build.gradle

در این بخش مانند شکل الف-۱۷ که `include ':libs:opencv'` را درون فایل `setting.gradle`

می‌نویسیم و روی گزینه `Sync Now` کلیک می‌نماییم.



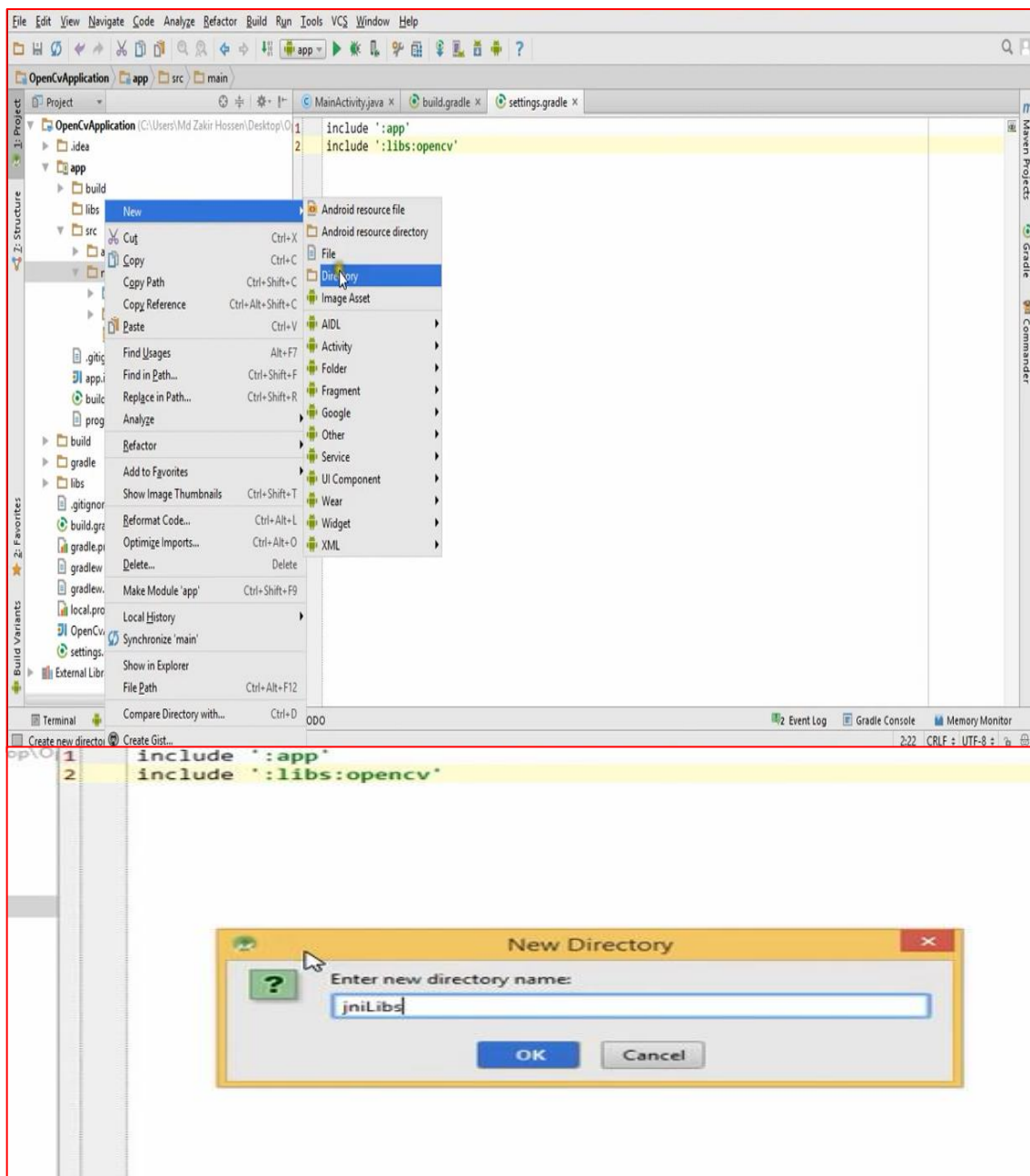
شکل ۶-۱۶ تعریف کتابخانه `opencv` در `stting.gradle`

در این مرحله مطابق شکل الف-۱۸ به بخش `main<src<app` می‌رویم و بر روی فایل `main`

راست کلیک می‌کنیم و یک دایرکتوری جدید ایجاد می‌کنیم. حال نام `jniLibs` را برای آن تعیین

می‌کنیم. می‌کنیم.





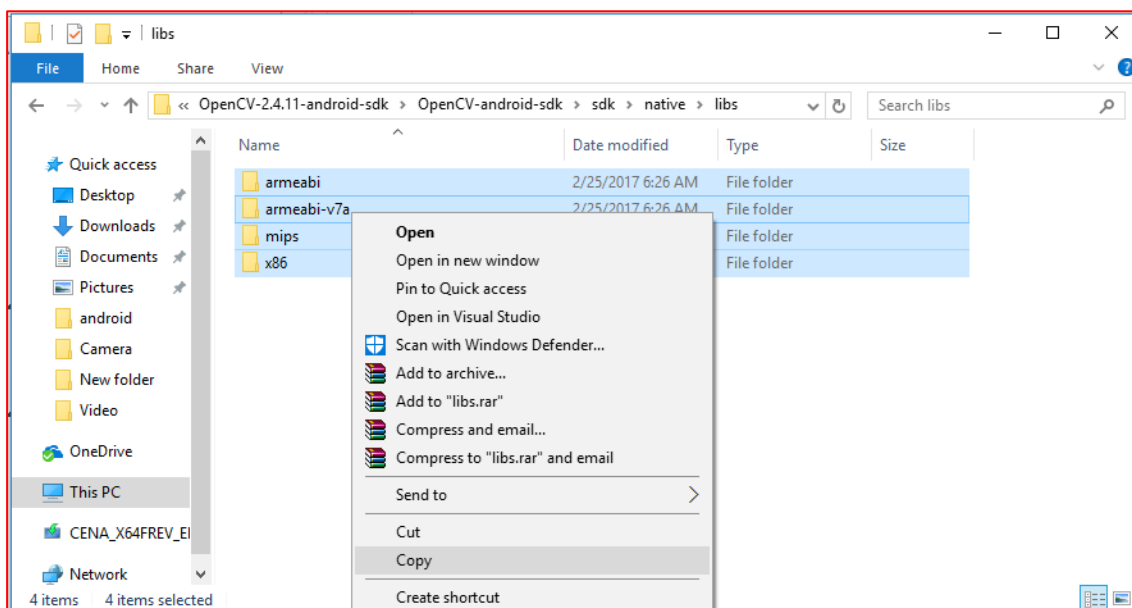
شکل ۶-۱۷ اضافه کردن کتابخانه بخش native مربوط به opencv

حال به مسیر `C:\OpenCV-2.4.11-android-sdk\OpenCV-android-sdk\sdk\native\libs`

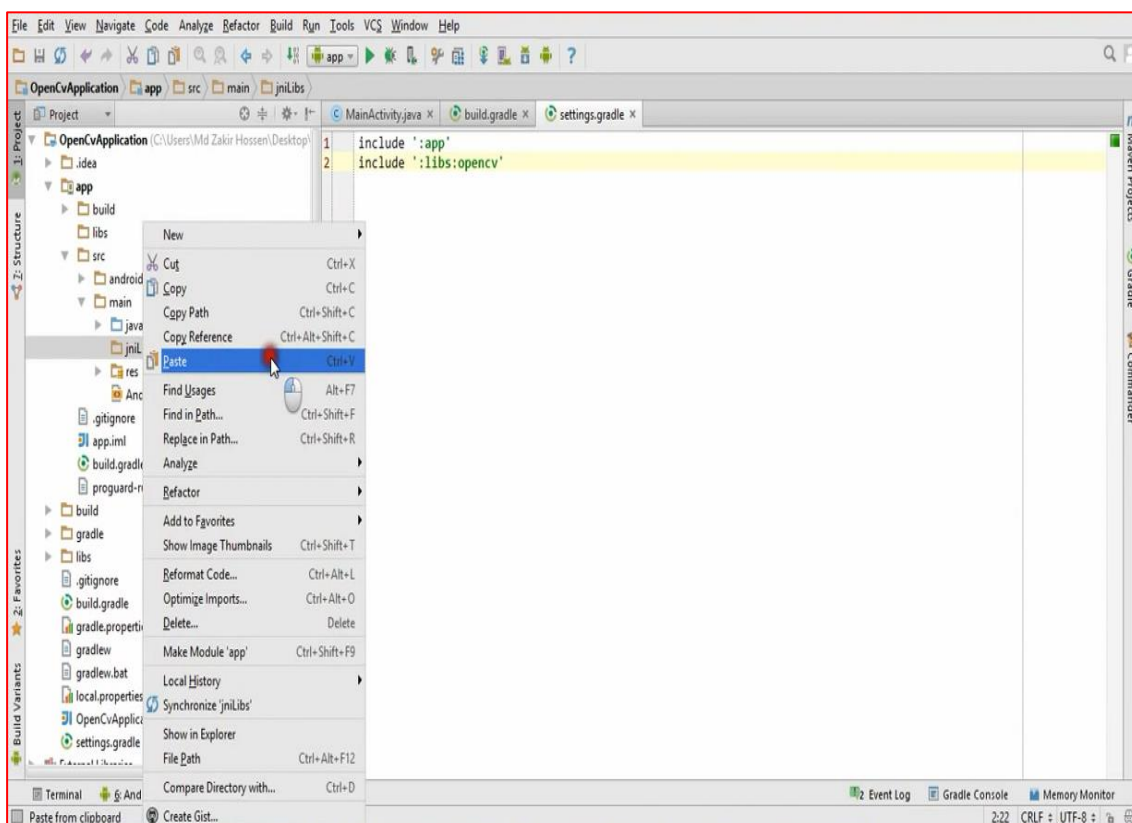
می‌رویم و از تمامی فایل‌های واقع در این مسیر یک رونوشت می‌گیریم و در مسیر

`jniLibs<main<src<app` جایگذاری می‌کنیم. این فرایند مانند شکل الف-۱۹ و الف-۲۰ انجام

می‌پذیرد.

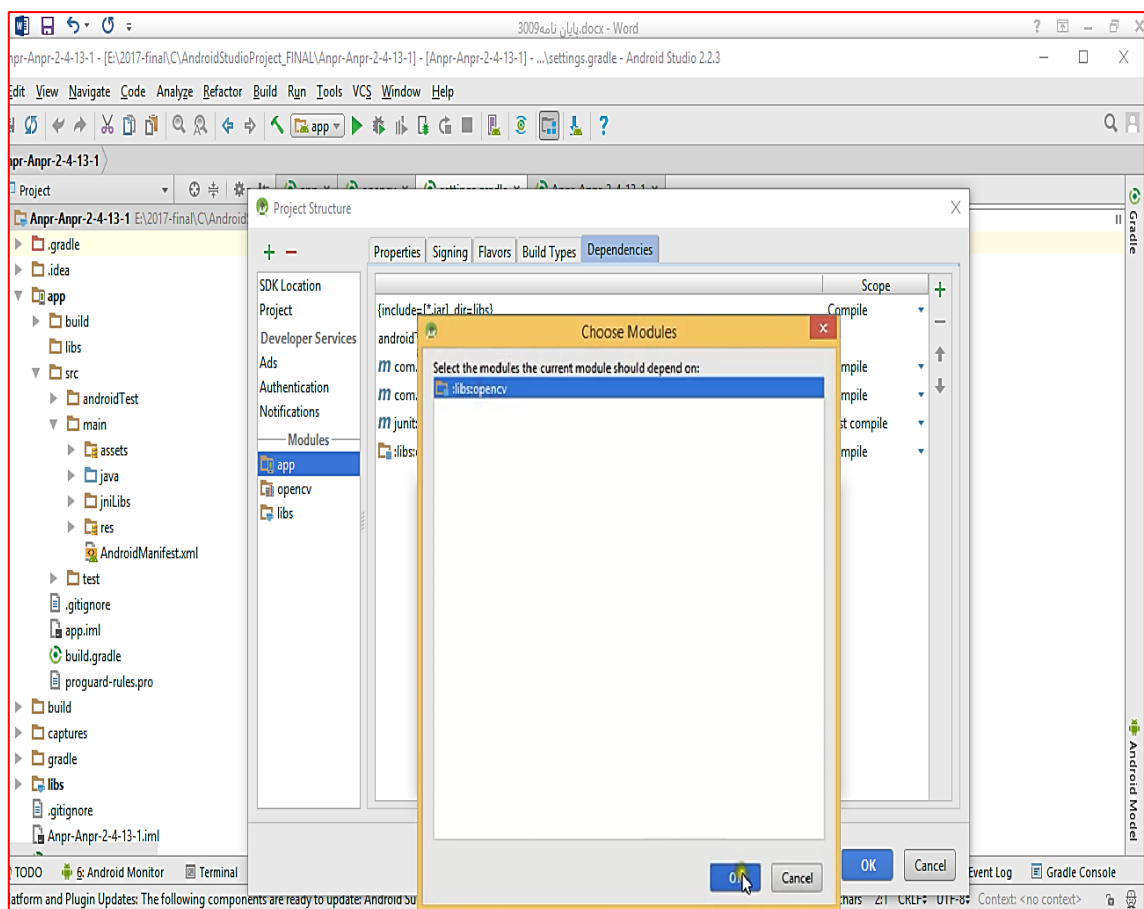


شکل الف-۱۸ فرایند رو نوشت گرفتن از کتابخانه بخش native مربوط به opencv



شکل الف-۲۰ فرایند جایگزاری کتابخانه بخش native مربوط به opencv

در این مرحله به قسمت project structure رفته و مطابق شکل الف-۲۱ عمل می کنیم.



شکل الف-۲۱ تعریف کتابخانه opencv در project structure

ابتدا روی گزینه app سپس Dependencies و add (+) کلیک کرده حال libs:opencv را انتخاب

می‌کنیم و در پایان بر روی Ok کلیک می‌نماییم. هم‌اکنون کتابخانه opencv اضافه شده است و می‌توان

از کدهای مربوط به این کتابخانه استفاده کنیم.

# مراجع:

[1] Y. R. Wang, W. H. Lin, S. J. Horng, "A sliding window technique for efficient license plate localization based on discrete wavelet transform," Expert Systems with Applications, Vol. 38, pp. 3142-3146, 2011.

[2] H. Chen, X. Xie, Y. Feng, J. Zhou, "A Novel Method of Trip Route Estimation Based on Vehicle License Plate Recognition System," Procedia - Social and Behavioral Sciences, Vol. 96, pp. 643-652, 2013.

[۳] و. سلیمانی، ح. واحدی شعاع، "شناسایی پلاک خودرو (نوع ایران) توسط گوشی‌های تلفن همراه با استفاده از تکنیک‌های بینایی ماشین"، اولین کنفرانس ملی محاسبات نرم و فناوری اطلاعات، ماه‌شهر، ایران، ۱۳۸۹.

[4] L. Luo, H. Sun, W. Zhou, Li. Luo, "An efficient method of license plate location," In Information Science and Engineering (ICISE), pp. 770 - 773, 2009.

[۵] ح. نجاتیان، ح. حسن‌پور، "تشخیص محل پلاک خودرو در تصاویر باکیفیت نامناسب"، هشتمین کنفرانس پیشرفت‌های علم و فناوری، مشهد، ایران، ۱۳۹۲.

[۶] علی برومن دنیا، محمود فتحی، "تشخیص اتوماتیک پلاک خودرو با استفاده از پردازش تصویر"، سومین کنفرانس مهندسی برق، دانشگاه علم و صنعت، اردیبهشت ۱۳۷۴.

[7] P. Sagharichi, M. Shakeri. "License Plate Automatic Recognition based on edge detection." IEEE In Artificial Intelligence and Robotics (IRANOPEN), pp. 170-174, 2016.

[۸] س. سرافراز، م. عباسی، ح نظام‌آبادی "آشکارسازی پلاک خودرو در تصاویر رنگی با استفاده از اطلاعات لبه"، ششمین کنفرانس بینایی و پردازش تصویر، اصفهان، ایران، ۱۳۸۹.

[9] A.H. Ashtari, M.J. Nordin, M. Fathy, "An Iranian license plate recognition system based on color features" IEEE Transactions on Intelligent Transportation Systems, 15(4), pp.1690-1705. 2014

[۱۰] ا. اشتری، م. فتحی، "روش جدید مکانیابی پلاک خودرو در تصاویر رنگی"، پنجمین کنفرانس ماشین بینایی و پردازش تصویر، ایران، ۱۳۸۷.

[11] S. Davoodnia, M. Ghasemzadeh, "A new algorithm for location recognition of Iranian car number plates, based on RGB color model and geometrical figures", e-Commerce in Developing Countries: With focus on e-Business (ECDC), 9th International Conference, pp. 1-5, 2015.

[12] T. Duan, T. Hong Du, T. Phuoc and N. Hoang, "Building an automatic vehicle license plate recognition system," in *International Conference Computer Science*, Can Tho, Vietnam, pp. 59-63, 2005.

[۱۳] م. روحانی، ح. عامری، "جایابی پلاک خودرو در یک فریم از فیلم ویدیویی با استفاده از تحلیل طیفی اصلاح شده"، سومین کنفرانس اطلاعات و دانش، مشهد، ایران، ۱۳۸۶.

[۱۴] و. ابوالقاسمی، ع. احمدی فر، "کاربرد تبدیل IFT در سیستم شناسایی پلاک خودرو"، سومین کنفرانس اطلاعات و دانش، مشهد، ایران، ۱۳۸۶.

[15] A. Mahabadi, A. Ranjbar, "Reliable License Plate Recognition" The CSI Journal on Computer Science and Engineering Vol. 7, No. 2, pp. 51-67, 2009

[16] H.N. Patel, D. Khushboo, T. Panchal. "An algorithm for automatic license plate detection from video using corner features." IEEE International Conference on Information Processing (ICIP), pp. 301-306, 2015.

[17] H. Rajput, T. Som, S. Kar, "An Automated Vehicle License Plate Recognition System" Computer 48th, No. 8, pp. 56-61, 2015.

[18] A. Broumandnia and M. Fathy, "Application of pattern recognition for Farsi license plate recognition," in International Conference GVIP, Cairo, Egypt, 2005.

- [19] J. Nijhuis, M. Brugge, K. Helmholt, J. Pluim, L. Spaanenburg, R. Venema and M. Westenberg, "Car license plate recognition with neural networks and fuzzy logic," IEEE, International Conference on Neural Networks (ICNN), Vol. 5, pp. 2232-2236. 1995.
- [20] L. Fuliang and G. Shuangxi, "Character Recognition System Based on Back Propagation Neural Network," in International Conference on Machine Vision and Human-Machine Interface (MVHI), 2010.
- [21] A. Nagare, "License plate character recognition system using neural network," International Journal of Computer Applications, Vol. 25, No. 10, July 2011.
- [22] C. Anagnostopoulos, E. Kayafas and V. Loumos, "Digital image processing and neural networks for vehicle license plate identification," Journal of Electrical Engineering, Vol. 1, No. 2, pp. 2-7, 2000.
- [23] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos and E. Kayafas, "A license plate recognition algorithm for intelligent transportation system applications," IEEE Transaction on intelligent transportation system, Vol. 7, No. 3, pp. 377-392, September 2006.
- [24] K. Kim, K. Kim, J. Kim and H. Kim, "Learning-Based Approach, for License Plate Recognition," IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing, Vol. 2, pp. 614-623, 2000.
- [25] R. Ebrahimpour, E. Kabir, H. Esteky and M. Yousefi, "A mixture of multilayer perceptron experts network for modeling face/nonface recognition in cortical face processing regions," Intelligent Automation and Soft Computing, Vol. 14, No. 2, pp. 145-156, 2008.
- [26] R. Ebrahimpour, E. Kabir and M. Yousefi, "Face detection using mixture of MLP experts," Neural Processing Letters, Vol. 26, pp. 69-82, 2007.
- [27] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing." (2002).

[۲۸] م. قاصری، ح. ابراهیم نژاد، "بازیابی تصاویر چهره با استفاده از هیستوگرام گرادیان‌های جهت‌دار در حالت تک مدل"، بیستمین کنفرانس مهندسی برق ایران، دانشگاه تهران، ایران، ۱۳۹۱.







## **Abstract**

Before the smart phones being so ever-present, license plate recognition systems were usually based on a high resolution camera, a powerful computer and a separated transmitter for sending information. In recent years, some of the organizations such as parking management and traffic monitoring companies suggested the implementation of ALPR on Android system in order to offering services by smart phones. In this thesis, we tried to answer to this demand and implement license plate recognition on smart phones. Even though, the speed is not very important in this usage, we tried to offer a quick algorithm for detection and recognition of license plates. To reduce the computational complexity, we used a simple and effective way. First, we use Vertical edge detection and Horizontal projection to distinguish candidate's areas of license plate. Then characters are isolated with a connected component labeling algorithm and finally a neural network is used to recognize characters of the plate. Furthermore, we used parallel processing to increase the running's speed of the application.

The proposed method was tested on pictures taken from smart phones. At first, this method was tested by C++ language and the OpenCV library in Visual Studio. Then, we implemented it by Java and OpenCV in Android Studio for smart phones. The average of processing time and the percentage of recognition were 0.67 and 96.1% respectively.

**Keywords:** Android, License Plate, Horizontal projection, Neural Nnetwork, ALPR, OpenCV, Parallel Processing





**Shahrood University of Technology**

**Faculty of Electrical Engineering and Robotic**

**M.Sc. Thesis in Digital Electronic Systems Engineering**

**Design and Implementation of Fast License Plate Recognition  
System for Android**

**By: Javad Khakshoor Kame Oliya**

**Supervisor:**

**Dr Hossein Khosravi**

February 2017