



مستند

گزارش پروژه

نرم افزار

تشخیص پلاک هوشمند

AI-PLATE

(پلاک بین)

سال و ترم تحصیلی:	تابستان ۱۴۰۴	نام استاد:	دکتر غلامعلی نژاد
تاریخ تألیف:	۳۱ شهریور ۱۴۰۴	مشخصات دانشجو:	احسان خدادوست ۴۰۱۱۱۲۰۱۰۴

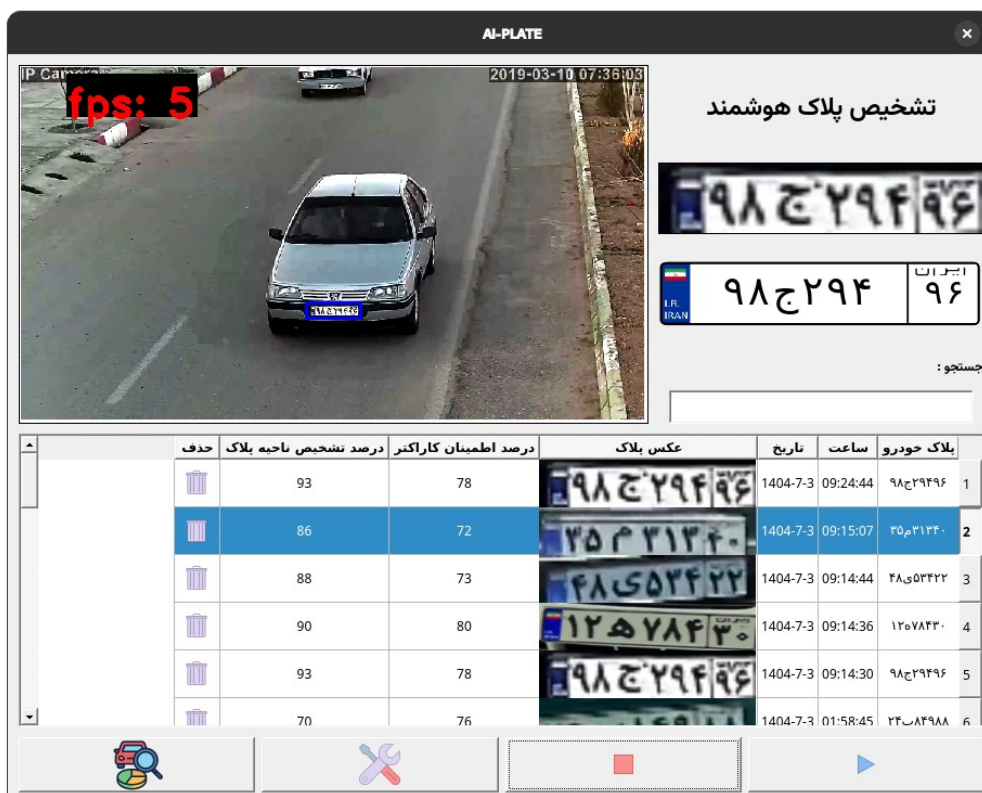


فهرست

چکیده:	2
مقدمه:	3
زبان برنامه نویسی و کتابخانه های استفاده شده:	4
رابط کاربری و ظاهر نرم افزار:	5
پایگاه داده و مدیریت داده ها:	6
ساختار پروژه:	7
الگوریتم ها و عملکرد کد:	8
روند پیشرفت پروژه:	10
موارد استفاده:	12
نوآوری ها و خلاقیت ها:	12
محدودیت ها و پیشنهادات برای بهبود:	12
نتیجه گیری:	13
منابع:	13

چکیده

نرم افزار **AI-Plate** (پلاک بین) یک سیستم تشخیص پلاک خودرو مبتنی بر هوش مصنوعی است که با بهره گیری از الگوریتم های پیشرفته یادگیری عمیق و پردازش تصویر، قادر به شناسایی و ثبت خودکار پلاک های خودرو از منابع ویدیویی مختلف (وبکم، فایل ویدیویی، یا جریان RTSP) است. این نرم افزار با استفاده از مدل های آموزش دیده YOLOv5، پلاک های خودرو و کاراکترهای آن ها را با دقت بالا شناسایی کرده و اطلاعات مربوطه را در یک پایگاه داده ذخیره می کند. رابط کاربری گرافیکی (GUI) این نرم افزار با استفاده از PySide6 طراحی شده و امکاناتی نظیر نمایش زنده ویدیو، جستجوی پیشرفته، و خروجی گیری به فرمت های اکسل و متنی را فراهم می کند. این گزارش به بررسی جامع جنبه های فنی، ساختار پروژه، الگوریتم ها، و روند توسعه این نرم افزار می پردازد.



The screenshot displays the AI-PLATE software interface. At the top, there's a title bar with 'AI-PLATE' and a close button. Below it, a live video feed from an 'IP Camera' shows a silver car on a road. A red overlay indicates 'fps: 5'. To the right of the video, the license plate recognition results are shown, including the detected plate '۹۸ ج ۲۹۴ ۹۶' and a smaller version of the same plate. Below the video, a table lists detected vehicles with columns for 'بلاک خودرو' (Vehicle Block), 'ساعت' (Time), 'تاریخ' (Date), 'عکس پلاک' (Plate Image), 'درصد اطمینان کاراکتر' (Character Confidence), 'درصد تشخیص ناحیه پلاک' (Plate Area Detection Confidence), and 'حذف' (Delete). The table contains six rows of data, with the first row highlighted in blue. At the bottom, there are icons for search, settings, and a red square button.

بلاک خودرو	ساعت	تاریخ	عکس پلاک	درصد اطمینان کاراکتر	درصد تشخیص ناحیه پلاک	حذف
۹۸ ج ۲۹۴ ۹۶	09:24:44	1404-7-3	۹۸ ج ۲۹۴ ۹۶	78	93	
۳۵ م ۳۱۳۴۰	09:15:07	1404-7-3	۳۵ م ۳۱۳۴۰	72	86	
۴۸ م ۵۳۲۲۲	09:14:44	1404-7-3	۴۸ م ۵۳۲۲۲	73	88	
۱۲ م ۷۸۴۳۰	09:14:36	1404-7-3	۱۲ م ۷۸۴۳۰	80	90	
۹۸ ج ۲۹۴ ۹۶	09:14:30	1404-7-3	۹۸ ج ۲۹۴ ۹۶	78	93	
۲۴ م ۸۴۹۸۸	01:58:45	1404-7-3	۲۴ م ۸۴۹۸۸	76	70	



مقدمه

تشخیص پلاک خودرو (License Plate Recognition - LPR) یکی از کاربردهای کلیدی فناوریهای هوش مصنوعی و پردازش تصویر در حوزههای امنیتی، ترافیکی، و مدیریت پارکینگ است. نرم افزار AI-Plate (پلاک بین) با هدف ارائه یک راه حل بومی و کارآمد برای شناسایی پلاکهای ایرانی طراحی شده است. این نرم افزار با پشتیبانی از حروف و اعداد فارسی و سازگاری با تقویم شمسی، نیازهای محلی را به خوبی برآورده می کند. از ویژگیهای برجسته این نرم افزار می توان به دقت بالای تشخیص، رابط کاربری کاربرپسند، و قابلیت ادغام با سیستمهای خارجی اشاره کرد.

هدف این گزارش، ارائه یک تحلیل جامع از جنبه های مختلف پروژه شامل زبان برنامه نویسی، کتابخانه ها، طراحی رابط کاربری، ساختار پایگاه داده، الگوریتم های مورد استفاده، و روند توسعه است. همچنین، چالش ها و راه حل های ارائه شده در طول توسعه پروژه بررسی می شوند.



۱. شروع: مقداردهی اولیه سیستم.

۲. ورودی: جریان تصویر/ویدئو را دریافت می کند.

۳. تشخیص پلاک: YOLOv5 پلاک ها را شناسایی می کند.

۴. تشخیص کاراکتر: مدل سفارشی کاراکترهای فارسی را تشخیص می دهد.

۵. بررسی پایگاه داده: پلاک را با پایگاه داده از نظر وضعیت و مالک مقایسه می کند.

۶. نمایش نتایج: نتایج تشخیص و شناسایی را در رابط کاربری گرافیکی نشان می دهد.

۷. ورودی لاگ: جدول ورودی های اخیر و پایگاه داده را به روزرسانی می کند.

۸. پایان تکرار: با ورودی جدید ادامه می یابد یا عملیات را به پایان می رساند.



زبان برنامه‌نویسی و کتابخانه‌های استفاده‌شده

زبان برنامه‌نویسی

نرم‌افزار AI-Plate به‌طور کامل با استفاده از پایتون (نسخه ۳.۱۲ و بالاتر) توسعه یافته است. انتخاب پایتون به دلیل سادگی، انعطاف‌پذیری، و پشتیبانی گسترده از کتابخانه‌های پردازش تصویر و یادگیری عمیق صورت گرفت. این زبان امکان توسعه سریع و ادغام آسان با ابزارهای مختلف را فراهم کرد.

کتابخانه‌های کلیدی

- **PyTorch** (نسخه ۲.۰+): برای اجرای مدل‌های یادگیری عمیق.
- **PySide۶**: که امکان طراحی یک رابط کاربرپسند و پویا را فراهم کرد، Qt مبتنی بر (GUI) برای ایجاد رابط کاربری گرافیکی.
- **OpenCV (cv۲)**: برای پردازش تصویر و ویدیو، شامل تغییر اندازه، چرخش، و اعمال فیلترهای پیش‌پردازش.
- **Pillow (PIL)**: برای پردازش تصاویر (مانند ذخیره تصاویر پلاک و تبدیل فرمت‌ها).
- **pandas**: برای مدیریت داده‌ها و خروجی‌گیری به فرمت اکسل.
- **sqlite۳**: جهت ذخیره اطلاعات پلاک‌ها SQLite برای مدیریت پایگاه داده.
- **datetime**: برای پشتیبانی از تاریخ و تقویم شمسی در رابط کاربری و پایگاه داده.
- **numpy**: برای انجام محاسبات ماتریسی و پردازش آرایه‌ها در پردازش تصویر.

رابط کاربری و ظاهر نرم افزار

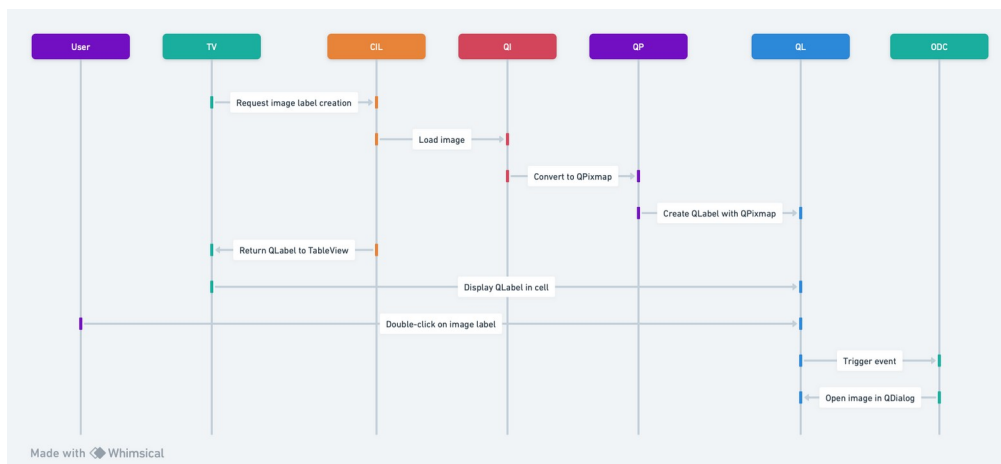
طراحی رابط کاربری

رابط کاربری نرم افزار با استفاده از **PySide6** و فایل طراحی **UI (mainFinal.ui)** ایجاد شده است. این رابط کاربری به صورت راست به چپ (RTL) طراحی شده تا با زبان فارسی سازگار باشد. اجزای اصلی رابط کاربری شامل موارد زیر است:

- **نمایشگر اصلی ویدیو:** برای نمایش زنده فریم های ویدیویی با مستطیل قرمز اطراف پلاک های شناسایی شده.
- **نمایشگر پلاک:** برای نمایش تصویر برش خورده پلاک و متن پلاک به صورت فارسی.
- **جدول اطلاعات:** نمایش لیست پلاک های ثبت شده با جزئیات (شماره پلاک، تاریخ، ساعت، درصد اطمینان، و تصویر پلاک).
- **دکمه های کنترلی:** شامل دکمه های شروع، توقف، انتخاب منبع (وب کم، فایل ویدیویی، RTSP)، و جستجو.
- **پنجره جستجوی پیشرفته:** امکان جستجو بر اساس بازه زمانی و فیلتر پلاک با نمایش نمودار و آمار.

ویژگی های بصری

- **آیکون ها:** آیکون های سفارشی (واقع در پوشه **icons**) برای دکمه های شروع، توقف، جستجو، و حذف استفاده شده اند.
- **قالب پلاک:** تصویر قالب پایه پلاک (**template-base.png**) برای نمایش زیباتر شماره پلاک.
- **نمودارها:** استفاده از **QChart** برای نمایش نمودار دایره ای (**Pie Chart**) از پرتکرارترین پلاک ها در پنجره جستجو.
- **تم و استایل:** استفاده از استایل **Windows** با فونت های خوانا و اندازه های مناسب برای تجربه کاربری بهتر.





پایگاه داده و مدیریت داده‌ها

ساختار پایگاه داده

پایگاه داده نرم‌افزار از **SQLite** استفاده می‌کند و در فایل `entries.db` ذخیره می‌شود. جدول اصلی به نام `entries` شامل ستون‌های زیر است:

- `platePercent` (عددی): درصد اطمینان تشخیص ناحیه پلاک.
- `charPercent` (عددی): درصد اطمینان تشخیص کاراکترها.
- `eDate` (متن): تاریخ میلادی (فرمت YYYY-MM-DD).
- `eTime` (متن): ساعت (فرمت HH:MM:SS).
- `plateNum` (متن): شماره پلاک به صورت انگلیسی.

عملیات پایگاه داده

- درج داده‌ها: تابع `insertEntries` در فایل `db_entries_utils.py` برای ثبت پلاک‌های جدید با استفاده از دستور `INSERT OR IGNORE`.
- بازیابی داده‌ها: تابع `dbGetAllEntries` برای دریافت تمام ورودی‌ها یا فیلترشده بر اساس شماره پلاک.
- جستجوی پیشرفته: تابع `dbGetEntriesByDateTime` برای جستجو بر اساس بازه زمانی و فیلتر پلاک.
- حذف ورودی: تابع `dbRemoveEntry` برای حذف یک ورودی خاص و تصویر مرتبط آن از پوشه `/temp`.

ذخیره‌سازی تصاویر

تصاویر پلاک‌های برش‌خورده در پوشه `/temp` با نام‌هایی به فرمت

`jpg.{eDate}_{eTime}_{plateNum}`

ذخیره می‌شوند. این تصاویر برای نمایش در جدول و پنجره جستجو استفاده می‌شوند.



ساختار پروژه

پروژه AI-Plate به صورت مدولار طراحی شده و شامل پوشه ها و فایل های زیر است:

پوشه ها و فایل ها

- **ai/**: برای پردازش تصویر و الگوریتم های پیش پردازش img_model.py شامل فایل **ai/**.
- **database/**: شامل فایل های مربوط به پایگاه داده (classEntries.py, db_entries_utils.py, entries.db).
- **gui/**: GUI و ماژول های کمکی (mainFinal.ui) شامل فایل طراحی **gui/**.
- **helper/**: (text_decorators.py) و پردازش متن (jalali.py) شامل ابزارهای کمکی مانند تبدیل تاریخ: **helper/**.
- **icons/**: آیکون های استفاده شده در رابط کاربری **icons/**.
- **model/**: YOLOv5 مدل های آموزش دیده (plateYolo.pt, CharsYolo.pt): **model/**.
- **temp/**: ذخیره تصاویر پلاک های برش خورده: **temp/**.
- **yolov5/**: برای بارگذاری و اجرای مدل ها YOLOv5 کتابخانه: **yolov5/**.
- **home-yolo.py**: فایل اصلی برای اجرای نرم افزار: **home-yolo.py**.
- **config.ini**: فایل پیکربندی برای تنظیمات پایگاه داده، مدل ها، و منابع ورودی: **config.ini**.

معماری نرم افزار

نرم افزار از معماری چندلایه ای استفاده می کند:

1. **لایه ورودی**: دریافت ویدیو از وبکم، فایل، یا RTSP.
2. **لایه پردازش**: پردازش فریم ها با استفاده از مدل های YOLOv5 و الگوریتم های پیش پردازش.
3. **لایه ذخیره سازی**: ثبت داده ها در پایگاه داده SQLite و ذخیره تصاویر در پوشه temp/.
4. **لایه نمایش**: ارائه اطلاعات از طریق رابط کاربری و نمودارهای آماری.

الگوریتم‌ها و عملکرد کد

الگوریتم‌های تشخیص

نرم‌افزار از مدل‌های YOLOv5 برای تشخیص پلاک و کاراکترها استفاده می‌کند:

- **تشخیص پلاک (plateYolo.pt):** شناسایی ناحیه پلاک در فریم‌های ویدیویی با دقت حداقل ۶۰٪.

- **تشخیص کاراکترها (CharsYolo.pt):** شناسایی کاراکترهای پلاک (اعداد و حروف فارسی) با دقت حداقل ۵۰٪ برای هر کاراکتر و میانگین اطمینان ۷۰٪ برای کل پلاک.

• مراحل پردازش:

۱. تغییر اندازه فریم به ۹۶۰x۵۴۰ پیکسل.
۲. اعمال خودکار کنتراست با استفاده از ImageOps.autocontrast.
۳. شناسایی پلاک با YOLOv5 و رسم مستطیل قرمز اطراف آن.
۴. برش ناحیه پلاک و پردازش آن برای تشخیص کاراکترها.
۵. مرتب‌سازی کاراکترها بر اساس مختصات x برای خواندن صحیح ترتیب.

پیش‌پردازش تصویر

فایل img_model.py شامل توابع پیش‌پردازش زیر است:

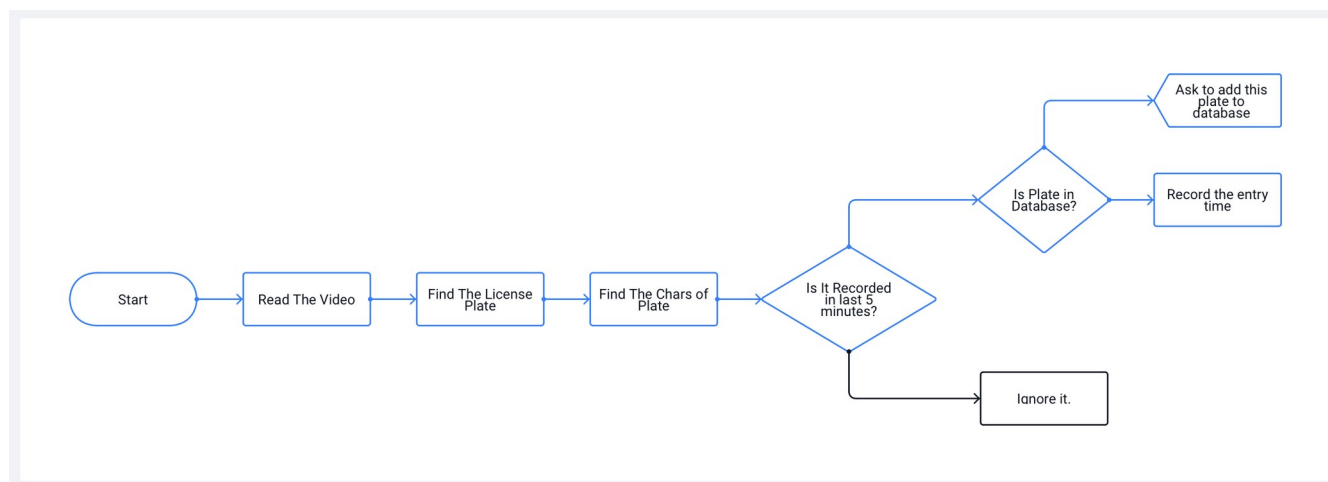
- **sharpen_new:** برای کاهش تعداد رنگ‌ها و واضح‌تر کردن مرزها K-means استفاده از.
- **sharpen_image:** ۳x اعمال فیلتر شارپن با کرنل ۳.
- **grayscale:** تبدیل تصویر به خاکستری و حذف نویز با عملیات مورفولوژی.
- **deskew:** HoughLines محاسبه و اصلاح کجی تصویر با استفاده از.
- **wrap_perspective:** تصحیح پرسپکتیو برای صاف کردن تصویر پلاک.

مدیریت نخ‌ها

- **Worker_۱**: نخ اصلی برای پردازش فریم‌های ویدیویی، تشخیص پلاک، و ارسال سیگنال‌های به‌روزرسانی به رابط کاربری.
- **Worker_۲**: نخ کمکی برای به‌روزرسانی جدول داده‌ها در رابط کاربری.

مدیریت داده‌ها

- **تبدیل حروف**: توابع `convert_persian_to_english` و `convert_english_to_persian` در `text_decorators.py` برای تبدیل پلاک‌ها بین انگلیسی و فارسی.
- **بررسی مشابهت**: تابع `check_similarity_threshold` برای جلوگیری از ثبت پلاک‌های تکراری در بازه زمانی کوتاه.
- **ذخیره‌سازی**: ذخیره تصاویر پلاک و اطلاعات در پایگاه داده با استفاده از `db_entries_time`.





روند پیشرفت پروژه

فاز ۱: برنامه ریزی و طراحی اولیه

- **اهداف:** تعریف نیازها، انتخاب فناوری‌ها (پایتون، PySide۶، YOLOv۵)، و طراحی ساختار پایگاه داده.
- **چالش‌ها:** انتخاب مدل مناسب برای تشخیص پلاک‌های ایرانی و سازگاری با حروف فارسی.
- **راه حل‌ها:** استفاده از مدل‌های YOLOv۵ آموزش دیده و تعریف دیکشنری کاراکترها (char_dict) برای پشتیبانی از حروف فارسی.

فاز ۲: توسعه مدل‌های یادگیری عمیق

- **آموزش مدل‌ها:** آموزش دو مدل YOLOv۵ (plateYolo.pt) برای تشخیص پلاک و CharsYolo.pt برای تشخیص کاراکترها) با دیتاست اختصاصی پلاک‌های ایرانی.
- **چالش‌ها:** کمبود دیتاست جامع برای پلاک‌های ایرانی و نویز در تصاویر.
- **راه حل‌ها:** استفاده از تکنیک‌های افزایش داده (Data Augmentation) و پیش پردازش تصویر برای بهبود دقت.

فاز ۳: پیاده سازی رابط کاربری

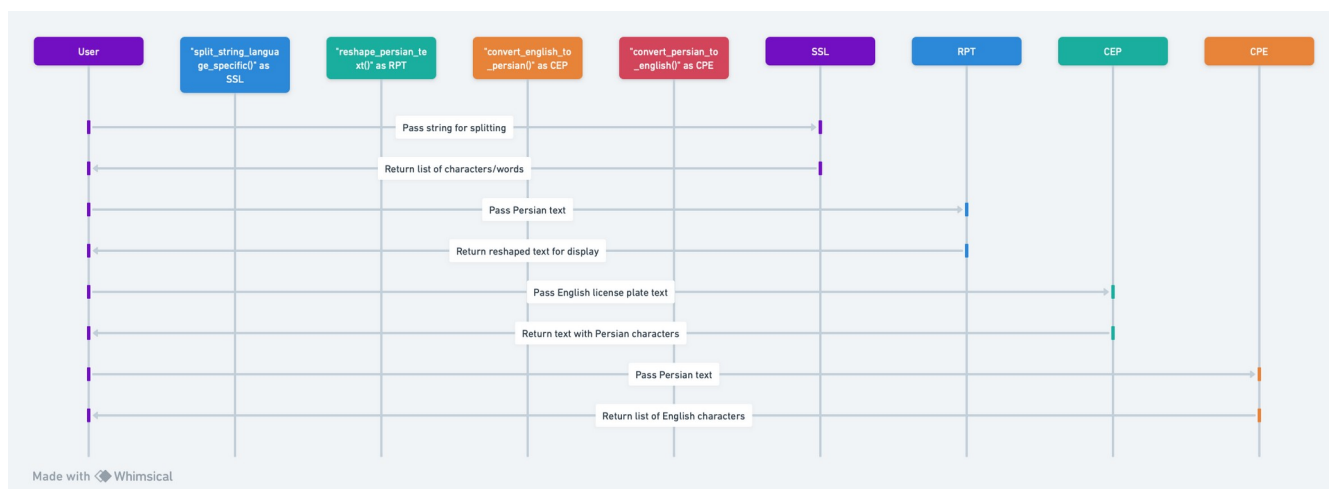
- **طراحی فایل mainFinal.ui با Qt Designer.**
- **پیاده سازی نخ‌ها (Worker۱, Worker۲) برای پردازش غیرهمزمان و به روزرسانی رابط کاربری.**
- **چالش‌ها:** مدیریت نخ‌ها برای جلوگیری از کرش برنامه.
- **راه حل‌ها:** افزودن متد closeEvent برای آزادسازی منابع و توقف نخ‌ها.

فاز ۴: ادغام و آزمایش

- ادغام مدل‌ها، رابط کاربری، و پایگاه داده.
- آزمایش نرم‌افزار با منابع مختلف (وب‌کم، ویدیو، RTSP).
- چالش‌ها: خطاهای مربوط به فرمت تاریخ و ذخیره تصاویر.
- راه‌حل‌ها: اصلاح توابع db_entries_time و timeDifference برای استفاده از تاریخ میلادی و مدیریت خطاها.

فاز ۵: بهینه‌سازی و نهایی‌سازی

- بهینه‌سازی سرعت پردازش با استفاده از GPU (در صورت وجود).
- افزودن قابلیت جستجوی پیشرفته و خروجی‌گیری به اکسل و متن.
- چالش‌ها: نمایش صحیح تاریخ شمسی در رابط کاربری.
- راه‌حل‌ها: استفاده از کتابخانه jdatetime و ماژول jalali.py.





موارد استفاده

- مدیریت پارکینگ: ثبت ورود و خروج خودروها با دقت بالا.
- کنترل ترافیک: شناسایی پلاک‌های متخلف در جاده‌ها.
- امنیت: نظارت بر خودروهای ورودی به مناطق حساس.
- تحلیل داده‌ها: ارائه گزارش‌های آماری از تردد خودروها با استفاده از نمودارها و خروجی‌های اکسل.

نوآوری‌ها و خلاقیت‌ها

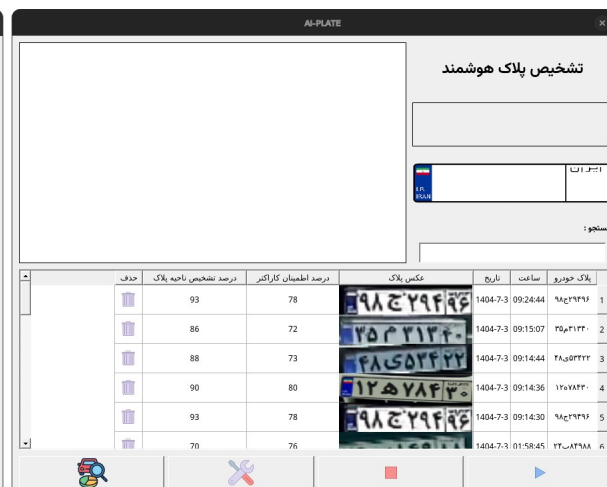
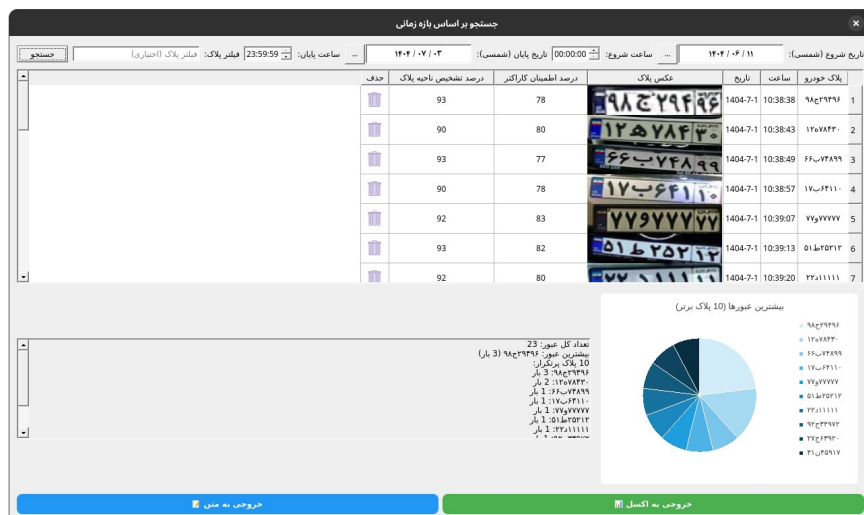
- پشتیبانی از تقویم شمسی: استفاده از `jdatetime` برای نمایش تاریخ به صورت شمسی در رابط کاربری.
- جستجوی پیشرفته: امکان جستجو بر اساس بازه زمانی و فیلتر پلاک با نمایش نمودارهای آماری.
- مدیریت هوشمند تکرار: استفاده از `check_similarity_threshold` برای جلوگیری از ثبت پلاک‌های تکراری.
- رابط کاربری کاربرپسند: طراحی RTL با آیکون‌های سفارشی و قالب پلاک.

محدودیت‌ها و پیشنهادات برای بهبود

- محدودیت‌ها:
 - وابستگی به کیفیت تصویر ورودی (نویز، نور کم، یا زاویه نامناسب می‌تواند دقت را کاهش دهد).
 - عدم پشتیبانی از چند دوربین به صورت همزمان.
- پیشنهادات:
 - افزودن قابلیت پردازش چندمنبعی (Multi-Source Processing).
 - بهبود مدل‌های YOLOv5 با دیتاست بزرگ‌تر برای افزایش دقت در شرایط نوری نامناسب.
 - افزودن قابلیت اتصال به API‌های خارجی برای اعتبارسنجی پلاک‌ها.

نتیجه گیری

نرم افزار **AI-Plate** (پلاک بین) یک راه حل پیشرفته و بومی برای تشخیص پلاک خودروهای ایرانی است که با استفاده از فناوری های مدرن یادگیری عمیق و پردازش تصویر، عملکردی قابل اعتماد ارائه می دهد. این نرم افزار با رابط کاربری کاربرپسند، پایگاه داده کارآمد، و قابلیت های جستجو و تحلیل داده، می تواند در کاربردهای مختلف ترافیکی و امنیتی مورد استفاده قرار گیرد. توسعه این پروژه نشان دهنده توانایی ترکیب فناوری های متن باز و بومی سازی آن ها برای نیازهای محلی است.



منابع

- YOLOv5 Documentation: <https://github.com/ultralytics/yolov5>
- PyTorch Documentation: <https://pytorch.org/docs/stable/index.html>
- PySide6 Documentation: <https://doc.qt.io/qtforpython-6/>
- OpenCV Documentation: <https://docs.opencv.org/>