

بابک آباد - متخصص هوش مصنوعی و رباتیک

<https://www.linkedin.com/in/babak-abad> | Babak.Abad@yahoo.com | ۰۹۹۲۱۶۸۷۰۹۸

تهران | www.ai-programmer.com | <https://www.github.com/babak-abad>

۱۰ سال سابقه در توسعه‌ی نرم‌افزار و طراحی سامانه‌های ماشین بینایی، پردازش زبان طبیعی و مولدهای تصویر و متن. توانایی در بهینه‌سازی مدل‌ها، عیب‌یابی آن‌ها و تبدیل وظایف پیچیده به بخش‌های قابل حل و مدیریت که منجر به ساخت سریع نمونه‌ی اولیه و ماژول‌هایی می‌شود که به سادگی نگهداری می‌شوند.

Deep Learning	Machine Learning	Image Processing	مهارت‌ها: Natural Language Processing	
Generative AI	Data Fusion	Sensor Fusion	Augmentation	Edge Device
Transfer Learning	Optimization	Automatic Labeling	Retrieval-Augmented Generation	
Python	C++	C#	C (AVR)	زبان‌های برنامه‌نویسی:
VS Code	PyCharm	Visual Studio	Colab	ابزار: Matlab Coder
Matlab ToolBox	Docker	Git	Lens	Data Fusions
Edge Device	Netron	AnyLabeling	CVAT	

تجربیات کاری

توسعه دهنده‌ی مدل‌های هوش مصنوعی (پیام گستر توسعه ایرانیان - تهران - ۱۴۰۳/۳/۱ تا هم اکنون)

- خلاصه‌سازی متن‌ها: بردارهای جاسازی (embeddings) متن‌ها، توسط مدل BGE-M3 استخراج شده و سپس توسط langchain به یک مدل 4o شرکت OpenAI داده شد که نیازمند منابع زیادی بود. جهت بهینه‌سازی، پس از فراخوانی BGE-M3، embedding ها توسط روش optics خوشه‌بندی شد، سپس از نزدیکترین متن‌ها به مرکز هر خوشه، چند نمونه انتخاب و به مدل mt5 داده شد که خلاصه آن‌ها استخراج شود.
- شناسایی موجودیت‌های مشخص (NER): مدلی بر پایه‌ی XLM-RoBERTa توسط gRPC با پایگاه داده ارتباط گرفته (با از Kafka) و با کمک دو صف و الگوی producer-consumer، NER را استخراج می‌کند. با بهینه‌سازی تعداد عبارات هر batch، کوانتیزه کردن لایه‌های آن توسط کتابخانه‌ی Transformers و torch و تعداد آیت‌های هر صف، سرعت اجرا حدود ۲ برابر بر روی CPU و ۳ برابر بر روی GPU شد.
- توسعه یک متن‌یاب بر اساس YOLO 11 و CRNN: با استفاده از Transfer learning، لایه‌های آخر VGG Net، بر بستر کتابخانه‌ی Ultralytics یک متن‌یاب آموزش داده شد. با به کارگیری لایه‌های cross batch normalization، کمبود VRAM جهت آموزش مدل رفع گردید تا بتوان با منابع محدود به دقت بالاتری رسید. علاوه بر این، از کتابخانه‌ی optimum جهت بهینه‌سازی هاپر پارامترها استفاده شد. با استفاده از pytorch، یک مدل CRNN نیز جهت خواندن متن محل‌های گزارش شده توسط YOLO، استفاده شد که ترکیبی از CNN و LSTM است. دسته‌بندی متن‌ها در قالب سطرها و پاراگراف‌ها، توسط خوشه‌بندی OPTICS انجام شد. برای augmentation از imgaug و torchvision کمک گرفته شد. حل مشکل vanished features با کمک pytorch-grad-cam و TorchCam که بر پایه‌ی Grad-Cam است، رفع گشت. همچنین، به جای آموزش حروف فارسی، از نمادها (glyphs) استفاده شد. با توجه به این که بسیاری از نمادهای موجود در فونت‌های فارسی شبیه هم هستند، هر یک به صورت چشمی بررسی و برچسب گذاری شد که باعث کاهش زمان یادگیری و کاهش تعداد لایه‌های مورد نیاز نسبت به معماری OCR قبلی (استفاده شده از شناسایی پلاک) شد. دقت این روش بابت یک دیتاست ۲۰ هزار تایی از تصاویر با متن فارسی که از شبکه‌های اجتماعی به دست آمده بود، ۸۹٪ است.

- **جستجوی تصویری:** در این روش از مدل CLIP بابت استخراج embeddings و ذخیره‌ی آن‌ها در پایگاه داده Milvus استفاده شد. بابت کاهش حجم ذخیره‌سازی شده و افزایش سرعت بازیابی، از PCA استفاده شد. همچنین، PCA به صورت مقطعی، یعنی وابسته به زمان یا تعداد درج‌ها، آموزش مجدد داده شد تا بردارهای موثر به صورت محلی استخراج نشوند و آن‌ها را به توان به کل پایگاه داده منتسب کرد.
- **تبدیل گفتار به متن:** این برنامه از طریق باکت s3، یک فایل ویدیویی یا صوتی را دانلود، سپس صوت آن را به کمک کتابخانه librosa را جدا کرده و از طریق مدل whisper، متن آن را استخراج می‌کند.

مهندس نرم‌افزار (تاجران جاده ابریشم - تهران - ۱۳۹۸/۴/۱ تا ۱۴۰۳/۳/۳۱)

- **سرعت‌سنجی خودرو از طریق پردازش تصویر:** در این روش، نخست، سرعت از طریق ردیابی پلاک محاسبه می‌شد. متن پلاک‌ها و موقعیت دقیقشان در یک صف ذخیره و بر پایه‌ی شباهت متن پلاک (ترکیبی از فاصله Levenshtein و Hamming) و فاصله‌ی زمان - مکان دسته‌بندی می‌شدند؛ چرا که در نمای دور، پلاک‌ها به صورت کامل قابل خواندن نیستند. خطای سرعت سنجی این روش بر روی ۱۶ دوربین، ۲٪ در شرایط مختلف به دست آمد (به تایید پلیس راهور). در ادامه، از روشی دیگر که مبتنی بر ردیابی خودرو بود، استفاده شد. در این روش، بخش ReID مربوط به الگوریتم DeepSORT نیز شخصی‌سازی گشت؛ از ۱۰۰ هزار تصویر بابت آموزش و جایگزینی تابع هزینه‌ی triplet و Euclidean با روش مقایسه دوگانه و معیار Cosine استفاده شد. استفاده از این روش، باعث کاهش ۵۰٪ تعداد دوربین‌های مورد نیاز برای پوشش جاده شد. این روش، ثبت خودروهای بدون پلاک را نیز میسر کرد.
- **تشخیص کاراکترهای پلاک (OCR):** مدل ترکیبی (CNN + RNN) CRNN در کتابخانه‌ی Keras، با کمک Albumentations، augmentation بابت AugLy، آموزش داده شد و به دقت ۹۹،۴٪ (تایید شده توسط پلیس راهور)، در شرایط سخت (سایه، تاری، گرد و خاک، اسپری، برف، خم شدن، نور کم و زیاد) رسید. برای حذف گلوگاه سرعت لایه‌های بازگشتی، مدلی مبتنی بر LeNet با پردازش دسته‌ای و بهینه‌سازی بیزی از طریق کتابخانه optimum و کتابخانه‌ی cppflow مدل باز طراحی شد که در نتیجه‌ی آن ۲۲ برابر سریع‌تر شد (بدون کاهش دقت). این تغییر همچنان باعث شد که فایل ONNX و pb (protocol buffer) ساخته شده در نسخه‌های قدیمی‌تر نرم‌افزار نیز قابل استفاده باشد.
- **پلاک‌یابی:** تنظیم دوربین و دریافت توسط کتابخانه‌ی Pylon انجام شد. سپس توسط پلاک‌یابی مبتنی بر YOLO 5 و با بهینه‌سازی بیزی هاپیر پارامترها، استفاده از DeepViz بابت کاهش تعداد لایه‌ها تا حد ممکن و استفاده از OpenVINO، سرعت اجرا از ۱۴ FPS (پیش از بهینه‌سازی) به ۶۰ FPS ارتقاء یافت و پلاک‌یابی نیز به بیش از ۹۸٪ رسید. مهاجرت به YOLO باعث شد که نیازی به استفاده از فیلترهای جهت‌دار (Polarized) و مادون قرمز گذر (IR-Pass) نباشد. با استفاده از لایه‌های quantized شده، این مدل بر روی Raspberry Pi 4 Model B و Neural Compute Stick 2 قابل اجرا شد که دقت آن به ۹۲٪ افت کرد ولی توان مصرفی، از ۷۰ به ۷ وات رسید. این مدل با تصدیق چند مرحله‌ای به دقت ۹۸٪ رسید که در کاربردهایی با سرعت بسیار پایین، مانند ورودی پارکینگ قابل استفاده شد.
- **همگام‌ساز دوربین‌ها، رادار و فلش:** در ابتدا به منظور ادغام داده‌های دوربین‌های مختلف، Radar و یا Lidar از مدل FUTR3D استفاده شد. سپس، جهت کاهش هزینه‌ها و حل مشکل عدم دسترسی به تجهیزات Radar و Lidar، از روش bird's-eye view (BEV) جهت ادغام داده‌ها استفاده شد. همچنین، به منظور همگام‌سازی آن‌ها با نورافکن‌ها، مداری مبتنی بر میکرو کنترلر AVR طراحی شد که باعث کاهش دمای پروژکتورها به میزان ۸۰ درجه، افزایش طول عمر نورافکن‌ها از ۱ به بیش از ۶ سال و کاهش توان مصرفی برق به میزان ۱۲ برابر شد.
- **مولد پلاک:** توسط یک مدل C-GAN، پلاک‌ها در شرایط مختلف ساخته می‌شود تا توسط این روش بتوان کارایی یک روش پلاک‌یابی را سنجید و یا آن را آموزش داد. این مدل در انواع حالت روشنایی نامتقارن، انواع حروف و عدد فارسی، انواع سایه و دیگر متغیرها، پلاک تولید می‌کند. علاوه بر این، با استفاده از تکنیک‌های augmentation، نظیر افزودن نویز، مات کردن، کشیدن، چرخش، افزایش و کاهش تضاد، نمونه‌های تولیدی افزایش یافت.
- **فرموله کردن نیازمندی‌های سخت افزاری:** تهیه‌ی چند فرمول که توسط آن مشخصات رایانه، دوربین، لنز، تعداد و نوع نورافکن‌ها و کارت شبکه‌های مورد نیاز با توجه به حداکثر سرعت خودروها، میزان تردد، عرض جاده و نوع کاربرد (سرعت سنج، چراغ قرمز، عوارض و ورود ممنوع) تعیین می‌شود. در این فرمول‌ها، کارایی کوانتومی، رنج پویا، حداکثر سیگنال به نویز، نویز تاریک، زمان ظهور، نرخ تصویر برداری و اندازه پیکسل دوربین‌ها نیز در نظر گرفته شد تا انتخاب دوربین جدید با حداقل‌های مورد نیاز، تطبیق کیفی داشته باشد.

مهندس نرم‌افزار (داده کاوان خزر پویا- رشت - ۱۳۹۵/۱۰/۱ تا ۱۳۹۸/۳/۳۱)

- تشخیص پلاک: تنظیم دوربین و دریافت تصاویر از طریق کتابخانه‌ی FLIR، انجام شد. در این روش ad-hoc، توسط لبه‌یابی و ریخت‌شناسی، پلاک خودرو پیدا شد که با بهینه‌سازی گوسی در جعبه ابزار Optimization از نرم‌افزار Matlab، فرآپارامترها بهینه شد و دقت روش به ۹۲٪ (تایید شده توسط پلیس راهور) و به سرعت ۵ FPS بر روی Core i7-2670QM رسید.
- خواندن متن داخل پلاک (OCR): روشی دو مرحله‌ای که ترکیبی از thresholding و contour detection بود استفاده شد. بابت thresholding از روش تطبیقی Bradley از کتابخانه‌ی AForge.NET استفاده شد؛ در ادامه، بابت افزایش سرعت اجرا، از پیاده‌سازی dlib، بابت پیاده‌سازی Bradley استفاده شد. بابت پیدا کردن contour ها از کتابخانه‌ی Emgu CV (OpenCV wrapper) استفاده شد. در این مرحله از گشتاورهای هندسی، Hu, Zernike و Tchebichef به عنوان روش استخراج ویژگی و کلاسبند SVM و Perceptron با تحلیل نتایج کتابخانه‌های AnyPredict و AutoKeras انتخاب شد.
- طراحی نرم‌افزار برچسب‌گذاری خودکار و گزارش‌گیری: این نرم‌افزار، بستری برای برچسب‌گذاری محل خودرو، پلاک، حروف و اعداد، نوع پلاک، کلاس، مدل خودرو (مانند سواری/سنگین و پراید/۴۰۵/۲۰۶) و رنگ خودرو فراهم می‌کند و برای تسریع کار، از مدل‌های آموزش‌دیده‌ی قبلی برای برچسب‌گذاری خودکار، بهره می‌گیرد و نتایج آن‌ها را در یک پایگاه داده MS SQL ذخیره می‌کند. بخش گزارش‌گیری نیز عملکرد هر روش را بر اساس دقت پلاک‌یابی، صحت پلاک‌خوانی و نوع پلاک‌های پیدانشده (Recall, Precision, ROC, False Positive) ارزیابی می‌کند.

توسعه دهنده‌ی نرم‌افزار (آوید - رشت - ۱۳۹۲/۲/۱ تا ۱۳۹۳/۲/۳۱)

- سیستم مشتری مداری توسط ردیابی و شناسایی: مشتری مداری در یک فروشگاه زنجیره‌ای در قالب ردیابی مشتری‌ها در داخل فروشگاه انجام شد تا خدماتی مانند راهنمایی مشتری در صورت ایستادن به صورت ساکن در یک نقطه، شناسایی مشتریان خاص، بالقوه و معرفی محل‌های خالی پارکینگ برای مشتریان انجام شود. در این نرم‌افزار از کارت‌های رادیو شناسه باند فوق فرکانس بالا (UHF-RFID) استفاده شد.

تحصیلات

کارشناسی ارشد - مهندسی هوش مصنوعی و رباتیک (دانشگاه آزاد واحد لاهیجان - ۱۳۹۶/۷/۱ تا ۱۳۹۸/۶/۳۰)

پایان نامه: افزایش نرخ تشخیص پلاک خودروهای ایرانی منطبق بر استاندارد اروپا با استفاده از لبه‌یابی کنی

کارشناسی - مهندسی نرم افزار (دانشگاه گیلان - ۱۳۸۷/۷/۱ تا ۱۳۹۲/۱۱/۱)

پروژه: تشخیص پلاک با استفاده از تکنیک‌های پردازش دیجیتالی تصویر و شبکه‌های عصبی مصنوعی

انتشارات

- B. A. Fomani and A. Shahbahrami, "*License plate detection using adaptive morphological closing and local adaptive thresholding*," 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), Shahrekord, 2017, pp. 146-150, doi: 10.1109/IPRIA.2017.7983035.
- Shahbahrami, B. A. Foomani and A. Akoushideh, "*A style-free and high speed algorithm for License Plate Detection*," 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP), Isfahan, Iran, 2017, pp. 76-81, doi: 10.1109/IranianMVIP.2017.8342372.
- T. Najafi, B. A. Foomani and A. Shahbahrami, "*Anxiety and Depression Detection using Statistical Features*," 2017 2nd National Conference on Soft Computing, Roudsar, Guilan, 2017.
- Pourkashani, A. Shahbahrami and B. A. Fomani, "*Copy Move Forgery Detection Using Histogram Quantization of Cross Power Spectrum*," 2017 International Conference on High Performance Computing & Simulation (HPCS), Genoa, 2017, pp. 826-831, doi: 10.1109/HPCS.2017.125.
- B. A. Foomani, A. Shahbahrami and M. Ahmadi, "*Iranian Private Vehicle License Plate Detection using Image Processing Technics*," 2013 8th Iranian Conference on Machine Vision and Image Processing (ICMVIP08), Iran, Zanjan, 2013.