
بسم الله الرحمن الرحيم

پژوهشگاه توسعه فناوری‌های پیشرفته
با مجوز رسمی از وزارت علوم، تحقیقات و فناوری

**عنوان:**

# YOLOv8-Custom: Extending the YOLOv8 Architecture with Custom Modules

# (یولو-سفارشی: گسترش معماری YOLOv8 با ماژول‌های سفارشی)

**نگارنده:**

**مجید خرمگاه**

**به سرپرستی:**

**دکتر میرحسنی**

**تاریخ:**

**مهر 1404**

# چکیده

پروژه YOLOv8-Custom ، یک اثبات مفهوم (Proof-of-Concept) برای سفارشی‌سازی و گسترش معماری شبکه عصبی YOLOv8 است. این پروژه با هدف فراهم آوردن بستری برای انطباق مدل با وظایف خاص(Task-specific Adaptation) ، مانند بهبود تشخیص اشیاء کوچک، طراحی شده است. در این پروژه، با موفقیت یک ماژول کانولوشن سفارشی (MyCustomConv) به جای یکی از لایه‌های استاندارد در بدنه (backbone) مدل yolov8nجایگزین شد. برای این منظور، با ویرایش فایل‌های هسته کتابخانه ultralytics، ماژول جدید به مدل معرفی گردید. پروژه نشان داد که معماری YOLOv8 کاملاً ماژولار و قابل توسعه است و این فرآیند با موفقیت از مرحله تعریف ماژول تا آموزش نهایی مدل سفارشی روی یک دیتاست نمونه، به انجام رسید. این کار، راه را برای پیاده‌سازی تغییرات پیچیده‌تر مانند افزودن هِدهای تشخیص جدید یا مکانیزم‌های توجه هموار می‌سازد.

## فهرست مطالب

# مقدمه

در بسیاری از کاربردهای بینایی ماشین در دنیای واقعی، مانند تشخیص پلاک خودرو از دوربین‌های نظارتی، مدل‌های استاندارد و از پیش آماده شده، عملکرد بهینه‌ای ندارند. توانایی تغییر و انطباق معماری یک مدل پایه مانند YOLOv8 برای مواجهه با چالش‌هایی نظیر تشخیص اشیاء کوچک، بی‌کیفیت یا دارای مانع، یک مزیت کلیدی است. هدف این پروژه، نمایش عملی فرآیند اضافه کردن یک ماژول سفارشی به معماری YOLOv8 و آموزش موفقیت‌آمیز مدل جدید است.

اهداف اولیه پروژه

- یادگیری فرآیند نصب کتابخانه ultralyticsدر حالت قابل ویرایش.(Editable Mode)
- پیاده‌سازی یک ماژول کانولوشن سفارشی در یک فایل پایتون جداگانه.
- ویرایش فایل YAML معماری برای استفاده از ماژول سفارشی.
- اصلاح هسته YOLOv8 برای شناسایی و مدیریت صحیح ماژول جدید.
- آموزش موفقیت‌آمیز مدل تغییریافته به عنوان یک اثبات مفهوم.

# فصل اول

# معماری سیستم

## 1-1. فلسفه طراحی

به جای پذیرش معماری YOLOv8 به عنوان یک جعبه سیاه ثابت، ما رویکردی ماژولار و منعطف را در پیش گرفتیم. فلسفه اصلی این است که هر جزء از شبکه (مانند یک بلوک کانولوشن) باید قابل تعویض با یک جزء سفارشی و بهینه‌تر برای یک کاربرد خاص باشد. این پروژه نشان می‌دهد که با درک سازوکار داخلیYOLOv8 ، می‌توان آن را به یک چارچوب قدرتمند و کاملاً قابل سفارشی‌سازی برای تحقیقات و توسعه تبدیل کرد.

# فصل دوم

# معرفی اجزا (Components Deep Dive)

## 2-1. ماژول سفارشی (MyCustomConv)

**نقش:** یک بلوک کانولوشن پایه که به عنوان جایگزینی برای ماژول Conv استاندارد عمل می‌کند. این کلاس شامل یک لایه Conv2d، یک BatchNorm2d و یک تابع فعال‌سازی SiLU است و ساختار آن کاملاً مشابه بلوک استاندارد YOLOv8 طراحی شده تا سازگاری تضمین شود.

## 2-2. فایل پیکر بندی معماری (my-yolo.yaml)

**نقش:** این فایلYAML ، نقشه ساخت مدل را تعریف می‌کند. ما با کپی کردن yolov8n.yaml، یک نسخه سفارشی ایجاد کردیم و در بخش backbone، دومین لایه Convرا با MyCustomConv جایگزین نمودیم. این فایل به سازنده مدل دستور می‌دهد که در آن نقطه از شبکه، از کلاس سفارشی ما استفاده کند.

## 2-3. فایل tasks.py (هسته سازنده مدل)

**نقش:** این فایل حیاتی، مسئول خواندن فایل YAML و ترجمه آن به یک مدل PyTorch است. مهم‌ترین تغییر در این پروژه، اصلاح این فایل بود. ماژول MyCustomConv به لیست base_modules در تابع parse_model اضافه شد تا سازنده مدل با آن مانند یک ماژول داخلی رفتار کرده و مدیریت ابعاد کانال‌های ورودی و خروجی را به صورت خودکار انجام دهد.

## 2-4. فایل ماژول‌های سفارشی (custom_modules.py)

نقش: برای حفظ پاکیزگی کد و جلوگیری از دستکاری بیش از حد کتابخانه اصلی، تمام ماژول‌های سفارشی ما (در این مورد MyCustomConv) در این فایل جدید که در مسیر ultralytics/nn/ ایجاد شد، قرار گرفتند.

# فصل سوم

# ویژگی‌های کلیدی و نوآوری‌ها

## 3-1. قابلیت گسترش معماری (Architectural Extensibility)

این پروژه به طور عملی ثابت کرد که معماری YOLOv8 یک ساختار بسته نیست. با این روش می‌توان هر نوع لایه یا بلوک جدیدی را به شبکه اضافه کرد، که این یک قابلیت کلیدی برای کارهای تحقیقاتی و بهینه‌سازی‌های پیشرفته است.

## 3-2. سازماندهی کد ماژولار (Modular Code Organization)

با جداسازی کدهای سفارشی در فایل custom_modules.py و معرفی آن به هسته، یک روش تمیز و قابل نگهداری برای توسعه مدل پیاده‌سازی شد که به‌روزرسانی کتابخانه اصلی ultralytics در آینده را آسان‌تر می‌کند.

# فصل چهارم

# فناوری (Technology Stack)

- زبان برنامه‌نویسی: **Python 3.8**
- چارچوب هوش مصنوعی: **PyTorch**
- کتابخانه اصلی: **Ultralytics YOLOv8**
- محیط مجازی: **venv**
- پیکربندی: **YAML**

# فصل پنجم

# راهنمای نصب و اجرا

1. **کلون کردن ریپازیتوری:**
   git clone https://github.com/ultralytics/ultralytics
2. **ایجاد و فعال‌سازی محیط مجازی:**
   python -m venv venv
   .\venv\Scripts\activate
3. **نصب وابستگی‌ها در حالت قابل ویرایش:**
   pip install -e .
4. **اجرای اسکریپت آموزش:**
   python train_custom.py

# فصل ششم

# چالش‌های فنی و راه‌حل‌ها

## 6-1. مشکلات

### 6-1-1. چالش: عدم شناسایی ماژول و خطای ناهماهنگی ابعاد

### 6-1-2. مشاهده

در تلاش‌های اولیه، پس از اجرای اسکریپت آموزش، برنامه با یک RuntimeError متوقف می‌شد. پیام خطا حاکی از آن بود که ماژول سفارشی ما یک ورودی با تعداد کانال مشخصی انتظار داشته، اما ورودی‌ای که از لایه قبلی دریافت کرده، تعداد کانال متفاوتی داشته است (مثلاً انتظار ۱۲۸ کانال، دریافت ۱۶ کانال).

### 6-1-3. راه حل

تحلیل دقیق نشان داد که سازنده مدل در tasks.py، ماژول ناشناخته MyCustomConv را به رسمیت نمی‌شناسد و منطق هوشمند محاسبه خودکار کانال ورودی/خروجی را برای آن اعمال نمی‌کند. راه‌حل، افزودن نام کلاس MyCustomConv به مجموعه base_modules در داخل تابع parse_model بود. این تغییر کوچک به سازنده مدل دستور داد تا با ماژول سفارشی ما دقیقاً مانند یک ماژول استاندارد Conv رفتار کند و مشکل را به طور کامل حل کرد.

# فصل هفتم

# مسیر توسعه آینده(Future Roadmap)

این پروژه موفقیت‌آمیز، زیربنای لازم برای پیاده‌سازی بهبودهای هدفمند زیر را فراهم می‌کند:

**پیاده‌سازی هِد تشخیص P2:** اضافه کردن یک شاخه تشخیص جدید از لایه‌های اولیه شبکه برای بهبود چشمگیر توانایی مدل در یافتن اشیاء بسیار کوچک مانند پلاک‌های دور.

**ادغام مکانیزم توجه (Attention):** افزودن ماژول‌های Attention مانند CBAM برای کمک به مدل جهت تمرکز بر روی ویژگی‌های مهم‌تر در تصاویر شلوغ و بی‌کیفیت.

**آموزش روی دیتاست اختصاصی:** استفاده از این معماری سفارشی برای آموزش روی یک دیتاست بزرگ و واقعی برای ساخت یک مدل کاملاً بهینه.

**توسعه خط لوله دو مرحله‌ای:** تکمیل سیستم با یک مدل OCR (مانند CRNN) برای خواندن نویسه‌های پلاکی که توسط مدل YOLOv8 سفارشی تشخیص داده شده است.