Automated Vehicle Damage Analysis and Reporting System

An End-to-End AI Pipeline for Automotive Inspections

Gabriel Fernandes Carvalho
Fev/2025
Universidade de Brasília

Project Overview

Objective:

- Develop a fully automated system for analyzing vehicle images and generating detailed, humanized reports.
- Applications include towing inspections, claims documentation, and general automotive evaluations.

Key Components:

- Image description
- Visual localization of damages
- License plate extraction and vehicle data retrieval
- Natural language report generation

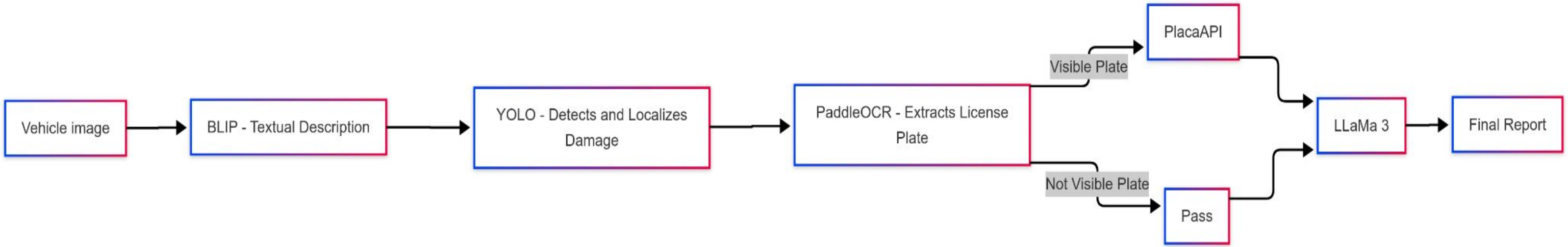
System Architecture & Pipeline

Input: Vehicle image

Processing Steps:

- **BLIP:** Generates detailed textual descriptions of the image
- **YOLO11:** Detects and localizes damage regions
- **OCR (PaddleOCR):** Extracts the vehicle license plate
- **External API:** Retrieves official vehicle details (model, year) using the plate
- **LLaMA 3:** Consolidates all data into a final, humanized report

Output: Annotated image with a comprehensive report



Dataset

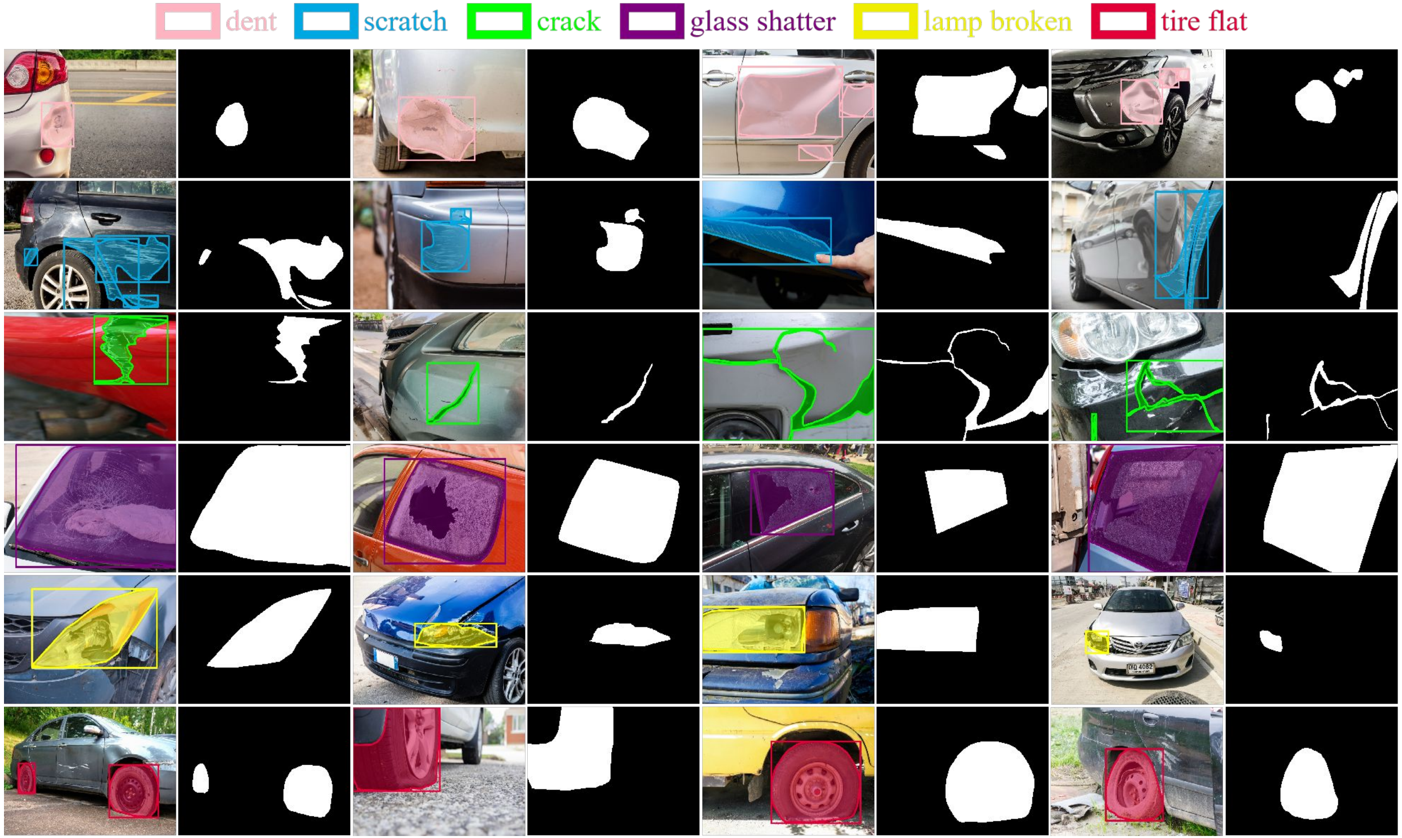
[CarDD: A New Dataset for Vision-based Car Damage Detection]

CarDD Dataset:

- Specialized dataset focused on vehicle damage detection.
- Contains 4000 annotated images highlighting various damages (dent, scratch, crack, glass shatter, tire flat, and lamp broken).
- COCO annotation.
- Split: Training set (2816 images, 70.4%), Validation set (810 images, 20.25%), and Test set (374 images, 9.35%)

Purpose in the Project:

- Serves as the training data for fine-tuning the BLIP and YOLO model.



[CarDD]

BLIP – Description Model

Overview:

- Fine-tuned on the CarDD dataset for detailed description.

Why Not CLIP?

- **CLIP Limitations:**
  - General-purpose image–text matching, not optimized for fine-grained description.
  - Less effective in highlighting specific vehicle problems.

BLIP Advantages:

- Transformer-based architecture designed for generating descriptive captions.
- Better suited for detailed and nuanced descriptions.

Image-Text Retrieval: “The man in blue shirt is wearing glasses.”

https://huggingface.co/Salesforce/blip-image-captioning-base

YOLO11 – Damage Localization

Purpose:

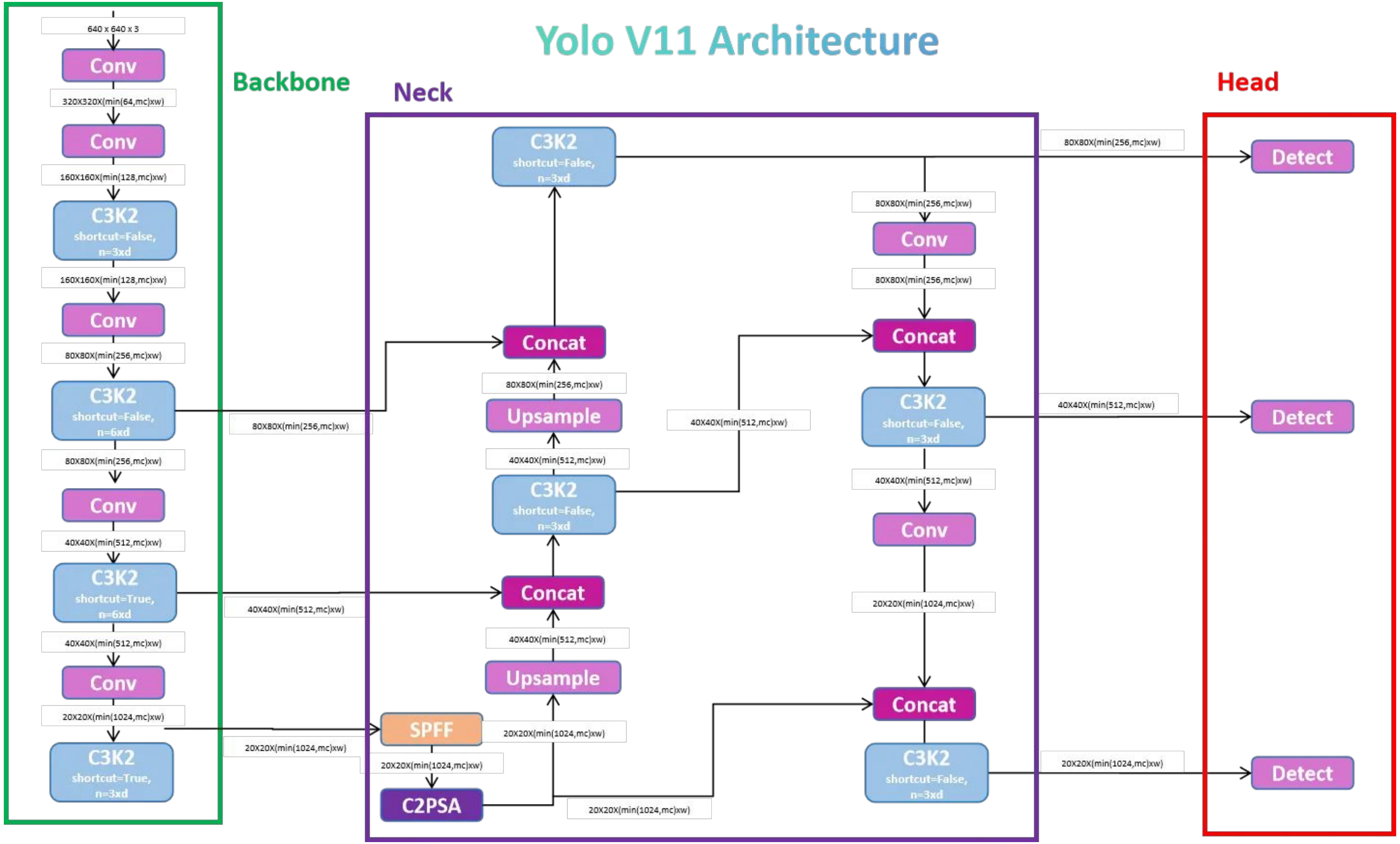
- Detects and localizes damages on the vehicle image.

Key Points:

- Uses the YOLO format for annotations.
- Trained to pinpoint the exact location of damages (scratches, dents, etc.).

https://docs.ultralytics.com/pt/models/yolo11/

Yolo V11 Architecture



[YOLO11 Architecture]

OCR – License Plate Extraction

Tool Used: PaddleOCR

Functionality:

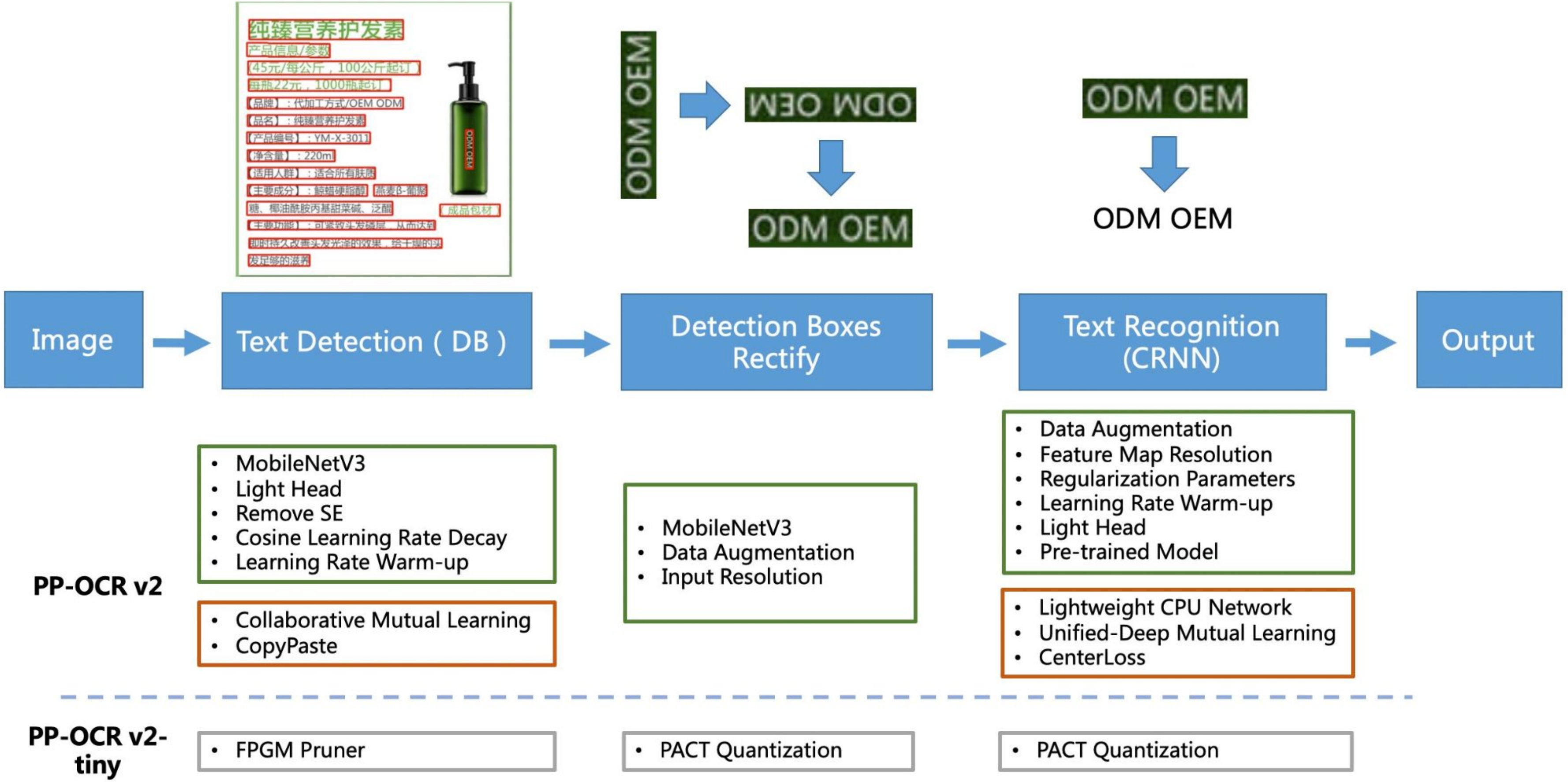
- Extracts license plate information directly from the vehicle image when visible.

Integration:

- The extracted license plate is used to query an external API for additional vehicle data (model, year).

Benefits:

- Enhances the report's reliability with official vehicle details.



https://paddlepaddle.github.io/PaddleOCR/main/en/ppocr/overview.html#pp-ocr_1

LLaMA 3 – Humanized Report Generation

Role in the Pipeline:

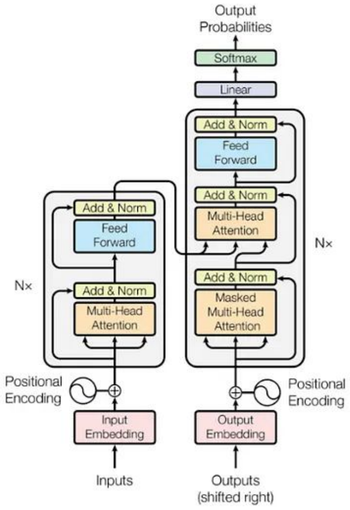
- Aggregates inputs from BLIP, YOLO11, OCR, and the external API.
- Generates a coherent, natural, and detailed report.

Why LLaMA 3?

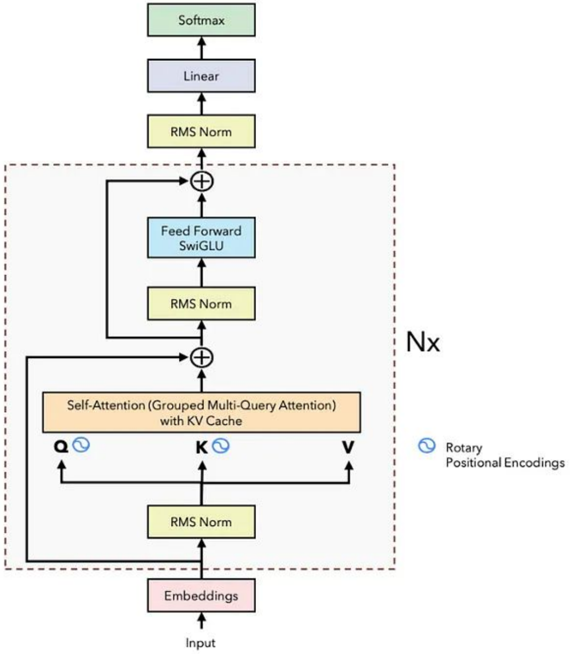
- Excels at natural language generation, ensuring that the final report is clear and standardized.

https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Transformer vs LLaMA



Transformer
("Attention is all you need")



LLaMA

[LLaMa Architecture]

Infrastructure & Tools

Development Environment:

- Google Colab Pro for training and inference.

Frameworks and Libraries:

- PyTorch, Hugging Face Transformers, PaddleOCR.
- Integration with external APIs for vehicle data.

Hardware:

- Utilization of GPU resources to accelerate training and inference.

References & Links

Colab Notebooks:

- BLIP Model: BLIP_Colab
- YOLO11 Model: YOLO_Colab
- OCR & API Integration: OCR_Colab
- Full Pipeline: Report_Colab