**YOLOP: You Only Look Once for Panoptic Driving Perception**

*Dong Wu, et al.*

* Source: <https://arxiv.org/pdf/2108.11250.pdf>
* Performs traffic object detection, drivable area segmentation (DAS), and lane-line segmentation (LLS).
* Compose of one encoder for feature extraction and three decoders.
* State-of-the-art results on BDD100k
* Real-time: 23 FPS on Jetson TX2
* High-precision and real-time are two most critical requirements for autonomous driving.
* Models for specific tasks:
  + RCNN and YOLO series – object detection
  + Enet and PSPNet – semantic segmentation
  + SCNN and SAD-Enet – lane detection
* Using different model for each task means longer time and thus not real-time. Moreover, you have limited computational resources on vehicles.
* Tasks in autonomous driving are related:
  + lanes are often the boundary of drivable area
  + drivable area usually closely surrounds the traffic objects.
* Thus, a multi-task network is more suitable.
* Single-stage detection networks (e.g. YOLO) are faster compared to the two-stage networks. Also, the grid-based approach in single-stage detection is more related to semantic segmentation (based on author’s experiments).
* End-to-end training (meaning training the whole network in one stage) works good enough as compared to alternating optimization (meaning training the network one task after the other).
* Backbone:
  + CSPDarknet (used in YOLOv4) to extract features
* Neck:
  + SPP and FPN modules
  + SPP generates and fuses features of different scales, and FPN fuses features at different semantic levels, making the generated features contain multiple scales and multiple semantic level information.
* Detection Head:
  + anchor-based multi-scale detection similar to YOLOv4
  + First, PAN – a bottom-up feature pyramid network
  + Then, PAN’s conv feature maps for detections at 3 different scales
  + 3 anchors
* FPN – transfers semantic information twop-down
* PAN – transfers positioning information bottom-up
* Drivable area and Lane Line Segment Heads:
  + Same network structure for both heads
  + Output of the last level of FPN to the heads; shape (W/8, H/8, 256) if W and H are input image dimensions
  + Three upsampling; Nearest Interpolation method
  + Head output shape: (W, H, 2)
  + 2 channels in output because we have two classes: drivable area and non-drivable area in one head; and lane-line and non-lane-line in the other head
* Loss Function:









and are focal loss. The first one is for classification and the second is for object detection confidence.

is , i.e. Complete IoU.

is cross-entropy loss with logits.

* are hyper-parameters that need to be tuned.
* BDD100K:
  + For autonomous driving
  + Diversity of geography, weather, environment, lighting conditions (morning, noon, night, etc.)
  + 10 classes
  + 70k training images
  + 10k validation images
  + 20k testing images (not publicly available)
* K-means to find anchors
* Adam optimizer; ,
* Initial learning rate: 0.01
* Warm-up
* Cosine-annealing
* Data Augmentation: HSV distortions, random rotating, scaling, translating, shearing, left-right flipping
* Input images: resized from 1280\*720 to 640\*384