2023 졸업프로젝트 진행 보고서

주제: 3D Object Detection

2023. 02. 07. ~ 2023. 03. 06.

윤성우

진행 상황

* 3DSSD: Point-Based 3D Single Stage Object Detector 논문 1회독

3DSSD 논문 질문

* SA (Set Abstraction), FP (Feature Propagation)의 의미
* point-based의 2 stage의 의미 🡪 proposal(SA downsampling & FP upsampling) – refinement (아래 정리 참조 1)
* FPS의 뜻 = Farthest Point Sampling? = 최대한 멀리 떨어진, 특징이 다른 점들만 남긴다?
* D-FPS (Euclidean distance)와 F-FPS (Feature distance) 의미와 차이점, sampling 과정에서 둘을 이용한다는 것의 의미 (아래 정리 참조 2)
* receptive field의 뜻 = feature를 뽑아내는 input region의 사이즈?  
  (출처: <https://theaisummer.com/receptive-field>)
* anchor free의 뜻
* regression head의 뜻 (regression = bounding box? = prediction?)
* Figure 1에서
  + 맨 처음 N x 4의 4는 (x, y, z, r)?
  + C1, C2의 의미 (채널의 수? 채널이라면 무슨 채널?)
  + SA layer에서 Group은 말 그대로 sampling 후 instance 단위로 점들 묶기, MLP는 그냥 신경망 통과시키기, MaxPool은 최댓값 뽑아내기?
* bin의 뜻 (angle bin, bin value)
* centerness가 regression이 아닌 classification에 해당하는 이유
* 추출한 feature를 neural network에서 사용하는 방법

3DSSD 논문 정리

LiDAR 센서만 사용하는 3D detection에는 voxel-based와 point-based가 있다. 본 논문은 point based. voxel based 방식은 voxel feature를 어떻게 초기화하느냐에 따라 달라진다.

(참조 1) point based는 통상 2가지 과정(stage)을 거친다. 🡪 proposal, refinement. 이때 proposal 과정에서 downsampling과 upsampling이 일어난다)

proposal: object의 bounding box 위치를 찾는 것

refinement (prediction): proposal의 결과에서 더 정밀하게 위치를 찾고, 그 위치를 찾은 object의 classification을 행하는 것

downsampling (SA): 속도를 높이기 위해 도메인의 크기를 줄이는 것 + enlarging receptive field? + point의 feature를 뽑아냄 (🡨 feature 추출이기 때문에 필수 과정)

upsampling (FP): downsampling 과정 중에 잃어버린 point의 feature를 broadcast하기 위해서?

upsampling (FP)와 refinement 과정에서 시간을 많이 쓴다. 🡪 본 논문에선 기존의 SA sampling 방식인 “furthest point sampling based on D-FPS” 대신에 F-FPS (fusion sampling)를 도입함으로써 이 과정을 생략한다. refinement를 없앰으로써 2-stage가 아닌 1-stage 방식을 취하는 것이다.

본 논문에서, downsampling(SA)를 거친 representative points를 이용하여 box prediction을 하는데, 이때 candidate generator layer를 사용한다. candidate point를 instance의 center로 치고, 그 center 주변의 representation points를 찾은 후, feature를 뽑아낸다. (🡨 perception? classification?) 이 feature는 anchor-free regression-head에 들어가 3D bounding box를 찾는 데에 사용된다.

D-FPS만 사용한 채로 FP layer을 없애면 box prediction은 SA layer에서 살아남은 representative point를 이용하게 되는데, D-FPS만 사용하면 point 간의 상대위치만을 고려하기 때문에 representative point의 대부분은 원래 양이 많던 background point가 되고, 상당수의 foreground instance에 해당하는 점을 잃게 된다. 살아남는 representative points의 개수가 적을수록 잃어버리는 foreground instance의 비율은 더 많아진다. (🡨 points recall) 이때 잃어버린 instance를 되살리기 위해 기존 방식들은 FP layer를 도입했던 것.

(참조 2) downsampling에서 살아남은 representative points는 positive point, 즉 background가 아닌 foreground instance의 point가 되도록 해야 한다. 이때 D-FPS을 사용하는 이유, 즉 Euclidean distance가 작은 가까이 붙어 있는 점들을 없애는 이유는, 가까이 붙어있는 점들은 하나의 instance 안의 점들일 가능성이 높고 그 점들이 모두 살아남으면 살아남게 되는 점의 개수가 늘어나거나 detect할 수 있는 instance의 개수가 적어지기 때문이다. 그리고 F-FPS를 사용하는 이유, 즉 feature distance가 작은 (feature가 비슷한) 점들을 없애는 이유는, feature가 비슷한 점들은 곧 background(배경)과 같이 의미 없이 넓게 깔려 있는 점들을 의미하기 때문이다.

negative point (background)가 너무 많이 버려지면 regression에는 도움이 되지만 classification에는 방해가 된다. negative point의 주변 점들이 없어져버려서 receptive field를 키울 수 없기 때문. (?) 이에 따라 모델은 pos point와 neg point를 구분하기 어려워져서 (neg point를 pos point처럼 생각해서) classification 성능이 떨어진다. (?) 따라서 본 논문에서는 pos point를 확보하는 F-FPS 방식과 neg point를 확보하는 D-FPS를 동시에 사용하는 fusion sampling 방식을 사용한다.

최종 prediction 이전에 (prediction head 거치기 전에) feature를 추출하기 위한 SA layer를 한번 거쳐야 한다. 이때 통상적으로 SA layer에서는 1. center point selection 2. surrounding point extraction 3. semantic feature generation 과정을 거친다. 본 논문에서는 이 과정에서의 계산 비용을 줄이기 위해 SA layer를 약간 변형한 candidate generation layer를 사용한다.

D-FPS에서 살아남은 점들은 대부분 neg point이기 때문에 initial center point로써 F-FPS에서 살아남은 점들을 사용한다. 하나의 그룹(instance)으로 묶인 F-FPS의 점들끼리의 상대위치를 이용하여 initial center point를 서로 가까워지도록 옮기고 (이 방식은 VoteNet에서 왔다) 이 옮겨진 점들을 본 논문에서는 candidate point라고 부른다. 본 논문의 candidate generation layer에서 이 candidate point를 center point라고 간주한다. 그 다음 D-FPS의 점과 F-FPS의 점을 이용하여 pre-defined threshold (ex. 떨어진 거리)에 따라 각각의 candidate point의 surrounding point를 찾는다. 그 다음 각 candidate point의 normalized location과 semantic feature를 concatenate한 다음, 이를 MLP layer에 입력으로 집어넣어 feature를 추출한다. 이렇게 추출된 feature는 최종 prediction인 prediction head의 입력으로 들어가, regression과 classification이 수행된다.

최종 regression head가 anchor-based였다면 multi-scale anchor와 multi-orientation anchor를 구축해야 하기 때문에, 본 논문에서는 anchor-free regression head를 사용했다. (본 논문의 취지가 속도가 빠른 point-based 3D detection을 개발하는 것이기 때문)

regression head에서 각각의 candidate point에 대하여 해당 instance와의 거리 (dx, dy, dz)와 그 instance의 크기(dl, dw, dh)와 방향을 예측한다. 이때 각 점들에 대한 방향 정보가 없었기 때문에, classification과 regression을 섞은 formulation을 사용하여 orientation angle regression을 수행한다. (?) 여기서 angle regression을 할 때에 미리 정의한 Na라는 단위로 구분하여 각도를 classification한다. (?)

LiDAR point cloud는 instance의 표면을 나타내기 때문에 centerness label을 활용하는 것은 좋지 않다. 따라서 본 논문에서는 LiDAR point를 그대로 사용하지 않고, 좀 더 instance의 중앙으로 옮겨진 candidate point를 사용하여 centerness label을 활용함으로써 instance들을 구별한다. centerness label은 두가지 값으로 정의된다. 1. lmask : candidate point가 해당 instance 내부에 위치하는지 2. lctrness : (공식은 논문 참고) 1에 가까울수록 중앙에 가까운 것. 최종 classification label은 lmask와 lctrness의 곱이다.

본 논문은 Loss function을 classification loss, regression loss, shifting loss 세가지로 구성한다. classification loss는 classification score와 centerness label로 구성된다. regression loss는 distance, size, angle, corner 네가지에 대한 regression loss로 구성된다. 이 중에서 angle regression loss는 orientation classification loss와 residual prediction loss로 구성된다.