

객체 탐지 및 추적 프로젝트

IT 공학전공 2216362 오유진

깃허브 주소: <https://github.com/yujinoh0103/objectTracking>

이 프로젝트는 객체 탐지 및 객체 추적을 실시간으로 수행할 수 있는 코드를 구현하고, 다양한 모델에 대한 예제 코드를 제공하는 것을 목표로 합니다. 객체 탐지 시스템은 주어진 입력 영상에서 특정 객체를 인식하여 그 객체를 바운딩 박스 형태로 구분하는 기술입니다. YOLO, Faster R-CNN, SSD와 같은 객체 탐지 모델의 예시를 담았습니다. 각 모델은 다양한 정확도, 속도, 성능을 기반으로 사용자가 선택할 수 있도록 지원됩니다. 또한, 객체 추적 시스템은 영상 내에서 인식된 객체의 이동을 추적하여 각 객체에 고유한 ID를 부여하고, 객체의 이동 경로를 추적하는 기술입니다. 객체 추적을 위한 알고리즘으로는 SORT, DeepSORT, ByteTrack, Kalman Filter 등을 사용하는 예제를 제공할 예정입니다. 추적 결과는 바운딩 박스와 고유한 객체 ID를 통해 표시됩니다.

객체 탐지 모델:

- YOLO는 고속의 실시간 탐지 모델로, 속도와 정확도의 균형을 이룹니다.
- Faster R-CNN은 높은 정확도를 자랑하는 모델로, 복잡한 객체나 작은 객체를 탐지하는데 유리합니다.
- SSD는 경량화된 모델로, 실시간 환경에서 빠른 객체 탐지가 가능합니다.

객체 추적 알고리즘

- SORT는 간단하고 효율적인 실시간 추적 알고리즘입니다.
- DeepSORT는 SORT를 확장한 알고리즘으로, 딥러닝 기반으로 더욱 정교한 추적을 지원합니다.
- ByteTrack은 객체 검출의 신뢰도를 높여 추적 성능을 향상시킵니다.
- Kalman Filter는 고전적인 알고리즘으로, 상태 예측 및 추정 기능을 제공합니다.

프로젝트의 의의 및 느낀 점

프로젝트를 통해 객체 탐지 모델에 대한 다양한 코드를 접하고, 수정해볼 수 있었습니다.

이 프로젝트는 깃허브에서 진행된 오픈소스 프로젝트라는 점에서도 의의가 있습니다. 프로젝트 과정에서 Git을 사용하여 개발 중 발생할 수 있는 여러 가지 버전 관리 문제를 체계적으로 해결했습니다. 특히, 각 커밋 메시지는 명확한 목적을 가지고 작성될 수 있도록 노력했으며, main 브랜치가 아닌 test 브랜치에서 작업해 안정성을 높였습니다. 이후

프로젝트가 끝난 후 pull request를 통해 main에 합쳤습니다.

깃을 이용한 프로젝트를 진행하면서 느낀 점은 버전 관리에 있어 중요한 점은 각 커밋이 명확하게 구분되어 있어야 한다는 점입니다. 실제로 저는 잘못 적은 내용이 있어 몇 번 커밋을 되돌렸는데, 이 과정에서 버전 관리가 중요하다는 것을 느꼈습니다. 이 과정에서 커밋 메시지를 명확하게 적어두었기 때문에, 충돌이 거의 없이 프로젝트를 관리하는 것이 가능했습니다.

이 프로젝트는 오픈소스로 진행했기 때문에 다른 참여자들의 기여를 기대하고 있습니다. 여러 개발자들이 Pull Request를 통해 기능 추가, 버그 수정, 성능 개선 등을 제안해 지금의 간단한 코드에서 더 발전해, 같은 모델에 대해서도 더 많은 예제를 다양한 국가의 언어로 주석을 달아 제공할 수 있기를 바라며, 더 명확하고 빠른 코드로 개선될 수 있을 것 같습니다. 그래서 객체 탐지 및 추적을 잘 모르는 사람도 쉽게 접근할 수 있고 객체 탐지 및 추적에 대한 지식이 있는 사람들도 정보를 얻어갈 수 있도록 운영될 수 있을 것입니다.

아쉬운 점 및 추후 개선 사항

실시간 영상 처리에 있어 성능을 높이기 위한 최적화가 부족했습니다. 이는 혼자 다양한 예제를 제공하기가 어렵기 때문에 발생한 문제였습니다. 또, 알고 있는 모델이 적기 때문에 적을 수 있었던 모델의 예시가 적었습니다. 거기에 더해, readme.md파일에 있는 모델에서 일부만 예시를 채울 수 있었습니다. 작성해둔 예제도 너무 간단하고, 정말 사용할 수 있는 방법에 대한 예시만 작성할 수 있었기 때문에 추후에 모델끼리 결합까지 고려해 더 좋은 성능의 모델을 만드는 코드를 제공하고 싶습니다.

또한, 객체 간 상호작용을 추적하거나, 빠르게 움직이는 객체들을 보다 효율적으로 추적할 수 있도록 알고리즘을 개선하고자 합니다. 객체 탐지 모델을 경량화하고 최적화하여 실시간 환경에서 더 높은 프레임 속도를 유지할 수 있도록 개선할 것이며, TensorRT나 ONNX와 같은 도구를 사용해 모델을 최적화하는 예제도 추가로 넣어두고 싶습니다.

또 지금은 간단한 예제만 제공하고 있지만 다양한 모델의 성능을 비교할 수 있는 테스트 코드를 제공해 성능이 가장 우수한 모델을 알 수 있도록 할 것입니다. 현재는 사람 등 이미 잘 알려져 있는 데이터셋을 중심으로 객체를 탐지하고 있지만, 추후에는 새로운 데이터셋을 만드는 것까지 예제가 확장될 것 역시 목표입니다,