안녕하세요 이번에 MLV랩에서 여름 인턴으로 참여한 김유리입니다.

먼저 개인사정으로 인해 발표를 영상으로 녹화해서 보내드리는 점 양해 부탁드리고 관련하여 궁금한 점이 있거나 보완할 수 있는 방향이 떠오르신다면 저에게 slack으로 연락을 주신다면 감사하겠습니다.

이번 랩인턴 생활하면서 저는 면수님과 함께 지환님이 제안해주신 축구 next shot video recognition 프로젝트에 참여를 하게됐습니다.

간단하게 프로젝트 먼저 설명을 드리자면 축구 게임이 진행되는 동안 현재씬에 사용되는 카메라 기법에 기반하여 다음 씬, next shot에서는 어떤 카메라 구도를 사용할 지 예측을 하는 것이 초기 목표였습니다. 비슷한 연구로는 movienet 셋을 이용하여 영화에서 쓰일 카메라 기법을 예측한 바가 있는데 추상적이고 예술적인 면이 담긴 영화와 달리 스포츠에서도 이를 적용할 수 있는지가 궁금해서 진행하게 되었습니다.

(notion) 우선 카메라 레이블은 크게 wide와 closeup 2개로 나누었고 이 안에 더 세분화하여 wide에는 main, drone, offside line, behind the goal, crowd 그리고 closeup에는 closeup corner, beside the goal, close-up side staff, player back, close-up player or referee, inside the goal 이렇게 11개의 label을 설정하게 되었습니다.

(tool)데이터셋은 저희가 사용할 형식에 맞는 데이터를 찾기 힘들어 직접 경기를 돌려보며 annotation을 통해 채워나갔습니다. 그 결과 전반전과 후반전을 합하여 약 1시간 반 정도 되는 실제 축구 영상 4개와 20분 정도 진행되는 fifa와 같은 온라인 축구 게임 영상 26개 정도를 확보했습니다.

실제 축구는 약 500개, 축구 게임은 한 영상당 50개 정도의 shot이 나왔습니다.

모델로는 앞서 발표했듯이 모바일, 즉 경량화가 가능한 movinets를 적용했습니다. 최종적으로 next shot을 예측할 수 있다면 Movinets로는 streaming 중에도 카메라에서 바로 process를 해서 송출을 할 수 있도록 하면 좋을 것 같다는 생각을 했습니다. 또한 2D image 모델을 여러 축으로 expansion을 시켜주어 video inference을 수행함으로써 효율성을 개선한 x3d도 함께 해당 논문에서 비교 대상이었 는데 ~~그리고 기본 모델인 resnet까지 함께 적용을 해보았습니다.

가장 먼저 모든 영상을 구간에 따라 다음 camera label, 즉 next category를 기반으로 나눠 해당 subclip들에 모델을 적용했습니다.

우선 movinet을 이용하여 current category, 즉 현재 game 화면으로부터 사용하고 있는 camera 기법을 인식하는지부터 보았는데 이는 큰 이변 없이 높은 성능을 나타냈습니다. 약 90%를 넘는 accuracy를 확인할 수 있었고 subclip에 해당하는 label들을 이렇게 print해보면 recognition이 잘 이루어지는 것을 확인할 수 있습니다.

다음으로는 이번 프로젝트의 주 목적이었던 현재 scene으로부터 다음 shot의 camera 위치를 예측할 수 있는지를 확인해봤습니다.

resnet과 x3d를 차례로 적용해줬는데 accuracy “~~” 와 “~~”를 달성한 것을 확인했습니다.

하지만 반면에 movinets에서는 성능이 오히려 저하되는 듯하며 저조한 accuracy가 나오는 것을 알 수 있었습니다.

이의 이유를 추측해봤는데 movinets에서는 causality, 즉 인과성을 달성해내기 위해 padding trick을 사용하여 receptive field을 앞으로 보내게 됩니다. 이 때문에 future frames으로부터 절대로 계산하지 않는데 이는 우리가 현재 shot으로부터 다음 장면, 즉 미래 future frame들에 대한 정보가 필요하기 때문에 next shot을 예측한다는 목적에 맞지 않다고 판단을 했습니다.

이를 확인하기 위해 causal Convolution을 모델에서 빼줬더니 50% 정도의 성능이 나왔다. 이는 resnet이나 x3d보다 높거나 전체적으로 비슷한 정확도를 가진 것을 확인할 수 있었습니다. 또한 latency나 총 걸린 시간, memory usage을 확인해보았을 때 모바일 모델인만큼 ~~ 한 결과를 내었습니다.

Movinet은 성능과 latency에 대한 tradeoff을 적절히 지키면서 우수한 성능을 내는 모델인 것을 확인할 수 있었습니다.

(Train\_loss, val\_accuracy 등등 같이 비교)

하지만 물론 resnet이나 x3d에 비해서는 유의미한 차이를 보였지만 이와 같은 accuracy도 만족스러운, 높은 성능은 절대 아니라고 생각을 했습니다. 사실 프로젝트를 시작할 때 잘 될지 불안했던 부분인데 그 이유는 현재 씬을 기반으로 next shot은 예측한다는 아이디어가 좀 불완전하다고 생각했습니다. 현재의 action으로 인해 다음 샷이 정해지는 게 아니라 나오는 어떤 action에 따라서 어떤 구도를 쓸 것인지 선택이 되기 때문이다.

예를 들어 현재 큰 액션 없이 상대 진영에서 패스를 하고 있다고 해서 다음에 바로 골을 넣을 수 있을 지, 아니면 ball out of play로 넘어갈 지는 선수 및 경기 상황에 따라 달린거지 현재 camera “shot”에서 확인할 수 있는 부분이 아니기 때문입니다.

그 다음 movinet으로 실제 축구가 아닌 축구 게임 데이터로 next shot 예측을 해봤는데 ~~성능이 나왔다.

앞선 결과와 약 40%, 거의 2배 정도 차이나는 accuracy를 확인할 수 있었는데 그 이유는 축구 게임 역시 기계로 프로그래밍이 되었다는 이유로 예상해볼 수 있습니다. Fifa와 같은 게임 데이터에서는 인간이 개발한 알고리즘을 이용하여 어느 정도 현상황에 대한 다음 movement가 계획이 되어있다면 실제 축구 게임에서는 이를 통제할 수 없다는 차이점이 성능에서 큰 차이를 불러온 것을 확인해볼 수 있습니다.

정리를 해보자면

또한 시간상 해결하지 못한 다음과 같은 한계를 찾을 수 있었는데 만약 모델을 더 발전시킨다면 다음과 같은 점을 보완하면 좋을 것 같습니다.

한계: 데이터가 부족하다

Input으로 영상만 넣지 않고 다른 정보도 함께 넣는다

Causal conv

우선 데이터를 직접 수집했기 때문에 모델에 적용을 하고 결과를 도출하기에는 턱없이 적은 양이라고 생각했습니다. 실제 축구, 축구 게임 데이터를 모두 합해도 약 3천개 정도의 subclip을 사용할 수 있었는데 이는 딥러닝에 사용하기 부족하기 때문에 추가로 자료를 수집하고 이에 augmentation과 transformation을 더해주어 해당 문제를 보완하면 좋을 것 같습니다.

또한 Input으로 지금 영상이 들어갔는데 이와 함께 player에 대한 정보나 공에 대한 좌표, 또는 annotation을 수행하면서 함께 넣었던 action label(red card/clearance/ceremony)와 같은 추가 정보를 함께 넣어주면 성능을 개선할 수 있다는 생각을 했습니다.

네 제 발표는 여기까지입니다. 들어주셔 감사합니다.