3.3 실험계획법

우리는 object detection을 이용한 Key-Frame 추출방법을 이용하였다.

기존의 추출방법과 새로 정의한 추출방법의 차이를 확인하기위해 대조 비교하였다. (또한 최적의 결과를 위해 최적의 임계 값과 파라미터를 찾는다.)<- 이건 쓸지 말지

실험순서는 다음과 같다.

* Key-frame 정의의 주관성으로 인해 key-frame extraction에 이용 가능한 균일한 평가 방법이 없다. 따라서 실험하기위한 영상을 선택할 때 Key-frame 이라고 생각되는 부분들을 확실히 포함하는 영상을 선택한다.
* 모델을 결과적으로 application에 적재하기 때문에 이를 위한 성능측정의 지표로 삼기 위하여 영상길이에 따른 분류시간의 변화를 관찰하기 위해 여러 경우의 시간대의 영상을 실험한다.
* 기존의 논문에서 나온 방법과 차별을 두기 위해 object-detection방법을 사용하여 영상에서 원하는 정보가 있는 경우에 가중치를 두는 방법을 채택하였다. 이를 평가하기 위해 두 경우의 성능과 결과를 비교한다.

실험에 사용되는 동영상은 30초,1분,2분의 길이를 가지는 영상들이다. 이 영상들은 본 연구의 목적에 맞게 사람이라는 object가 많이 나온다. 최종적으로 3가지의 동영상을 object-detecion 방법을 이용한 결과와 그렇지 않은 결과의 차이를 비교한다. 결과로 나온 사진을 비교하여 이전과의 시간차이를 비교하고 Key-Frame 추출에 영향을 미치는 지를 알아보고 실험을 종료한다.

4.1실험결과 비교

기존의 모델과 object detection을 추가한 모델을 비교하기 위해 각각 세 가지 경우의 영상으로 실험을 진행하고 결과를 비교한다. 다음 표 1,2,3 은 각각의 영상에 대한 시간 측정결과이다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | No Object | | Object majority =3 | | Object majority = 5 | |
| Key-Frame 시간 | 30sec | 2.07s | 30sec | 18.42s | 30sec | 18.21 |
| 60sec | 6.11s | 60sec | 22.93 | 60sec | 23.73 |
| 120sec | 21.91s | 120sec | 93.29s | 120sec | 78.21s |
| Total 시간 | 30sec | 7.4s | 30sec | 23.44s | 30sec | 23.27s |
| 60sec | 24.99s | 60sec | 38.95s | 60sec | 41.25s |
| 120sec | 64.72s | 120sec | 135.39s | 120sec | 133.33s |

표1. 시간이 각각 30s, 60s, 120s 인 영상을 3가지의 모델로 비교해본 결과

위 표는 서로 다른 영상을 3가지의 모델로 비교해 보았다. 첫 번째 모델은 기존의 논문에서 이야기한 모델을 구현한 것이고, 두 번째와 세 번째 모델은 Object-Detection을 추가한 모델에서 가중치를 다르게 두어 실험한 결과이다. 속도의 문제 때문에 Object detection을 Yolo라는 기존의 모델을 이용해 사용한 결과, 기존의 모델보다 적게는 3배에서 많게는 9배까지 나타난 것을 알 수 있다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | No Object | | Object majority =3 | | Object majority = 5 | |
| 사진갯수 | 30sec | 7 | 30sec | 9 | 30sec | 9 |
| 60sec | 11 | 60sec | 12 | 60sec | 13 |
| 120sec | 5 | 120sec | 6 | 120sec | 8 |

표 2. 각각의 모델에서 추출한 사진의 개수

위의 표를 보면 기존의 모델과 가중치를 추가한 모델의 차이를 확인할 수 있다. 가중치를 두어모델을 실험한 결과에서 더 많은 Frame이 뽑힌 것을 확인할 수 있다.

또한 위의 표에서 보듯이 시간의 길이가 더 길다고 해서 뽑힌 key frame의 개수가 더 많은 것이 아님을 알 수 있다.

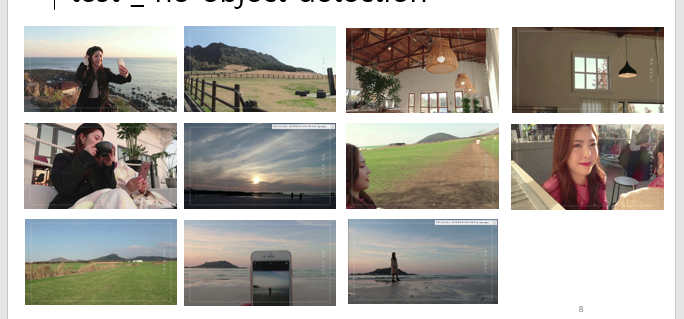


사진1. 1분 영상을 기존의 모델로 뽑은경우

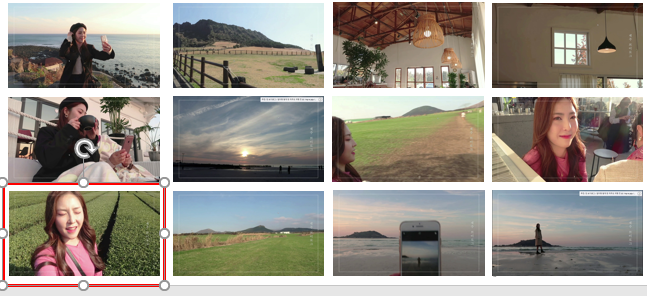


사진 2. Object-detecion weight=3 인 경우의 사진

기존의 모델과의 object-detection 모델을 실험한 결과, 한장의 사진만 다른 것을 확인할 수 있다. 새로 추가된 사진은 사람이 나오는 사진이 추가되었는데, 이는 사람의 label에 가중치를 두었기 때문이다.

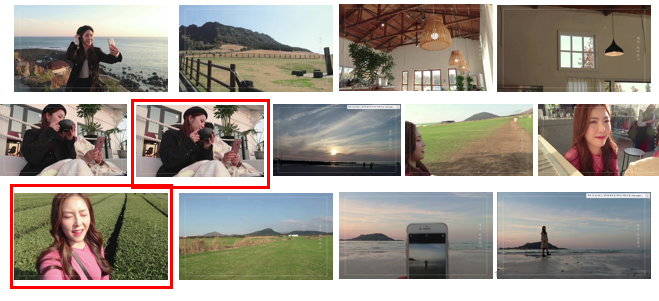


사진 3. Object-detecion weight=3 인 경우의 사진

4.2 평가 및 보완점

실험 결과를 바탕으로 기존의 Key-Frame extraction의 결과보다 Object detection을 적용한 것이 더 좋은 결과가 나온다는 것을 알 수 있다. 이를 활용하여 사용자가 만화에 많이 등장했으면 하는 object를 선택하게 할 수 있다. 사용자의 요구사항에 맞는 object를 추출하여 사용자의 선택에 맞는 Key-Frame을 추출할 수 있지만 120s의 실험에서 보듯이 시간이 너무 오래걸리는 문제가 발생할 수도 있기 때문에 수정이 필요하다.

또한 object detection을 위해 사용한 Yolo 라는 모델의 최신버전을 사용했을 때의 경우와, 실제로 application에 적재되어 실행했을 때의 시간도 비교해 볼 필요성이 있다. 환경에 따라 영상에서 Frame을 추출하는데 소요되는 시간이 다를 수 있기 때문이다. 시간이 오래걸린다면 영상에서 추출하는 Frame의 개수를 줄이는 방법도 존재한다. 현재 영상에서 추출하는 Frame의 수가 많기 때문에 시간이 오래 걸리고 장면의 전환이 없는 영상 같은 경우에는 중복된 Frame이 다수 나옴을 확인했다.

앞으로 최적의 정확도와 시간을 만족하기 위해 최적의 임계 값을 찾는 과정이 필요하다. 필요하다면 영상의 유형마다 다른 임계 값을 설정하고 사용자가 영상의 유형을 선택하면 그 에 맞는 임계 값을 설정하고 추출하는 Frame의 수를 조절하여 최적의 결과값을 얻을 수 있다.