장면 분할 기법을 위한 의미적 유사도의 모델링

정의손[O], 전성준[**], 조동휘[*], 금용호[*], 함동균[*], 김은지[*], 박승보[*]
[O]인하대학교 소프트웨어융합공학과,
[*]인하대학교 소프트웨어융합공학과,
[**]인하대학교 메카트로닉스공학과
e-mail: junsticehand1999@inha.edu[O], tkfkddl0787@naver.com[**], {junejoe97[*], go9149[*]}@inha.edu,
{ejsflqm17[*], hwangto001[*]}@gmail.com, molaal@inha.ac.kr[*]

Modeling of Semantic Similarity for Scene Segmentation

Eui-Son Jung[O], Seong-Jun Jeon[**], Dong-Hwi Cho[*], Yong-Ho Geum[*],
Dong-gyun Ham[*], Eun-Ji Kim[*], Seung-Bo Park[*]
[O]Dept. of Software Convergence Engineering, INHA University,
[*]Dept. of Software Convergence Engineering, INHA University,
[**]Dept. of Mechatronics Engineering INHA University

● 요 약 ●

본 논문에서는 의미적 유사도 기반의 장면 분할 방법을 제안한다. 이 방법은 의미적 접근을 통해 기존 연구에서 가졌던 한계를 극복하고 정확한 장면 분할이 가능할 것으로 기대한다. 의미적 유사도 비교를 Class 종류 비교, Class별 객체의 개수 비교, 샷 간의 Histogram비교, 객체의 관심영역(ROI) Histogram비교 총 4가지 규칙으로 정의했고 이때 도출된 4가지 유사도는 전처리를 거쳐 종합 유사도를 계산한다. 또한 의미적 접근을 통해 연속되는 Shot의 유사도를 비교하고 기준값에 따라 Shot을 묶어서 최종적으로 의미적 유사도(Semantic Similarity)에 기반한 장면의 경계(Scene Boundary) 분할 방법을 제시한다.

키워드: 의미적 유사도(Semantic Similarity), 장면 분할(Scene Segmentation)

I. Introduction

최근 인공지능 및 딥러닝의 발전으로 동영상 활용 분야가 매우 다양해지고 있다. 특히 GPU 발전으로 과거에 불가능했던 실시간 영상처리가 가능한 단계에 접어들면서 영상 분석, 영상 제작 등 사용자에 따라 다양한 목적으로 장면 분할이 적용되고 있다. 장면 분할을 위해서는 영상 속에서 다양한 정보를 추출할 수 있어야 하기에 객체 추출과 객체의 정보를 판단하는 연구 또한 활발하다. 본 논문에서는 Object Detection을 위한 YOLO 알고리즘을 사용하여 영상의 속성을 정의할 수 있는 메타데이터(Meta Data)를 생성하고 추출된 정보들을 활용해 의미적 유사도 기반의 장면 분할 방법을 제안한다[1].

II. Related works

2.1 장면 분할 기법

장면 분할 기법은 비디오 요약, 비디오 분석, 내용(Story)에 기반한 장면 분할과 같은 분야에서 많이 활용된다. 기존 연구에서의 장면 분할방식은 밝기, 히스토그램과 같은 방법을 사용해 왔다[2]. 히스토그램과 같이 픽셀 기반의 유사도 비교는 간단하고 장면 분할에서 꼭 고려해야 할 요소 중 하나지만 객체의 이동, 교차편집 등 잡음에 매우 민감하게 반응해 정확도가 떨어지는 단점을 내포하고 있다. 또한 유사도 비교를 위한 방법을 독립적으로 적용해 오차와 한계성이 뚜렷하다. 가장 큰 이유는 의미적 유사도(Semantic Similarity)에 기반한 접근을 하지 못했기 때문이다. 따라서 본 연구에서는 의미적 요소들을 적용하여 종합적인 의미적 유사도 접근을 통한 장면 분할(Scene Segmentation)방식을 제안한다[2][3].

2.2 객체 검출 및 인식 방법

객체 검출(Object Detection) 방식에서 가장 많이 이용되어 온 방식은 YOLO를 통한 방식이다. YOLO는 지금도 꾸준히 새로운 버전을 내놓으며 발전 중인 딥러닝 API로 Sementic Segmentation, Instance Segmentation을 가능하게 한다. 본 논문에서는 장면 분할의

첫 단계인 $Shot$에 대한 구분은 HSV Channel에 대한 Histogram Compare 기법을 적용해 이미 완료한 상태에서 이후 단계인 장면 분할을 주로 다룬다. 따라서 $Shot$을 입력으로 사용했고 YOLO를 활용한 객체 검출 및 Identity를 부여를 통해 Instance Segmentation을 하였고 이를 바탕으로 의미적 유사도 기반의 장면 분할에 활용했다.

III. Scene segmentation

3.1 의미적 유사도의 판단

$Shot$간의 의미적 유사도란 $Shot$과 $Shot$에 포함된 객체들의 유사도로 정의한다.

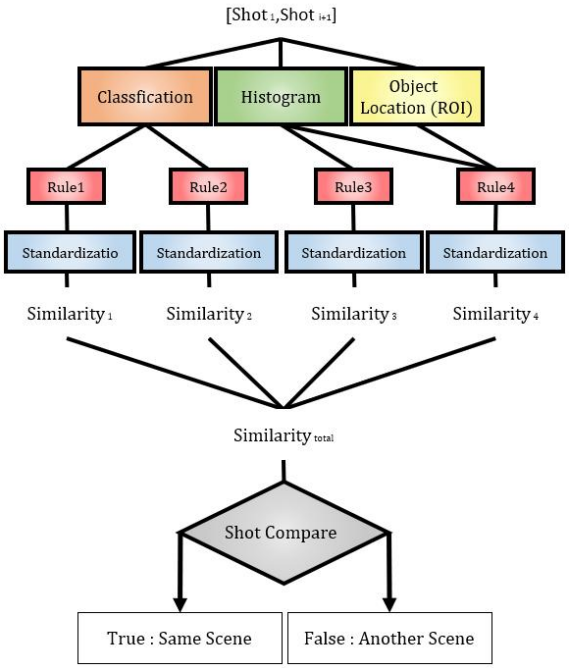

Fig. 1. Decision Flowchart for Semantic Similarity

$Shot$간의 의미적 유사도에 대한 계산과정은 Fig. 1과 같이 이루어진다. 본 논문에서는 의미적 유사도에 기반한 장면 분할의 방법론을 주로 다룰 것이고 객체의 Instance segmentation은 2.2절의 설명과 같이 모두 완료했다. 먼저 입력되는 $Shot$에 대한 모든 객체에 관한 정보는 YOLO를 통해 추출했고 객체가 속하는 Class(Classification), 프레임($Frame$)당 검출된 객체의 수, ROI 기반의 객체의 좌표 정보를 검출할 수 있었다. 의미적 유사도에 기반한 장면 검출에 활용하기 위해 총 4가지 Rule을 조합해 유사도 검출에 적용하였고 3장에서 관련 내용을 자세히 다룬다. 또한 각 Rule 적용을 통해 얻은 유사도의 Scale 차이로 인한 오차를 방지하기 적절한 전처리($Standardization$)를 거쳐 최종 유사도인 $Similarity_{total}$을 도출한다.

3.2 장면 분할을 위한 Rule 적용

이해를 돕기 위한 부호 규약은 다음과 같다.

$Sc = Scence \quad , \quad Sh = Shot$

$F = Frame \quad , \quad Ob = Object$

$id = identity \quad , \quad D = denominator$

$count([0,1,2,3,4]) = 5$

Sh_{index}^{name}

id는 객체의 고유한 특성을 나타내었고 유사도를 0~1의 값으로 정규화하기 위해 D를 사용하였다. 의미적 유사도 기반의 장면 분할을 위해 4가지 Rule을 적용해 유사도($Similarity$)를 구하고 Rule을 조합해 $Similarity_{total}$을 도출한다.

Rule 1. Similarity of Shot Object Classes

$$Sh_n = \{Ob_0, Ob_1, Ob_2, \cdots Ob_n\} \tag{1}$$

YOLO를 통해 $Shot$별로 검출된 모든 객체의 Class를 비교해 각 $Shot$별로 몇 개의 Class가 일치하는지를 계산해 $Similarity_1$을 도출한다. YOLO는 $Shot$에 속하는 모든 객체(Object)를 식(1)과 같이 검출한다. 이때 검출된 객체들은 YOLO가 검출하는 총 80개의 Class 중 하나이다. $Shot$이 시간의 흐름에 따라 $Shot_1, Shot_2 \cdots Shot_n$으로 이어질 때 각 $Shot$에서 검출된 모든 객체를 리스트에 담아 중복되는 Class는 하나만 남도록 정리해주면 각 $Shot$별로 검출된 Class의 종류를 식(2)와 같이 집합에 담을 수 있다.

$$Sh_1 = [person, tie, car]$$
$$Sh_2 = [person, tie, bicycle, dog] \tag{2}$$

검출된 Class의 집합 식(2)와 같다고 가정하고 Sh_1, Sh_2에 대한 Class 유사도 비교를 Table. 1에 나타냈다. 본 논문에 적용한 YOLO가 분류하는 Class의 개수는 80개이다. 각 $Shot$에서 검출된 객체는 정의된 80개의 Class에 Index Number로 접근할 수 있다. 따라서 각 $Shot$에서 검출한 객체의 Class는 80개의 1차원 배열로 나타낼 수 있고 [1×80] 1차원 배열 2개를 AND 연산하여 Class가 일치할 때만 1의 값을 가지게 한다.

Table 1. Comparation of Object Classes in Two Shots

	index length = 80											
	index(1) person	...	index(5) tie	...	index(15) car	...	index(50) cell phone	...	index(62) bicycle	...	index(80) dog
Shot1	1	...	1	...	1	...	0	...	0	...	0
Shot2	1		1		0		0		1		1
AND	1		1		0		0		0		0

AND 연산을 적용하면 Class가 동일한 경우에만 '1'의 결과를 얻을 수 있다. 이를 모두 합해 $Similarity_1$ 도출 식(4)의 분자로 적용한다. 분모는 식(3)과 같이 Sh_1과 Sh_2의 총 객체 수를 비교하여 더 큰 값을 식(3)과 같이 D_1 에 저장한다.

$$D_1 = \max(count(Sh_1), count(Sh_2)) \quad (3)$$

$$Similarity_1 = \frac{count(Sh_1 \wedge Sh_2)}{D_1} \quad (4)$$

따라서 식 4와 같이 0~1 범위를 가지는 Rule 1.에 대한 의미적 유사도($Similarity_1$)을 도출할 수 있다. 이때 유사도의 값이 1에 가까울수록 유사도가 높다고 정의한다.

Rule 2. Similarity of Object Count between Classes

$$Sh_i = \{count(class(Ob_0)), count(class(Ob_1)) \cdots count(class(Ob_n))\} \quad (5)$$

YOLO를 통해 $Shot$별로 검출된 모든 객체의 Class의 수를 저장해 각 $Shot$안에 Class별 수를 비교해 $Similarity_2$을 도출한다. YOLO로 얻어낸 $Shot$의 Class 집합은 식(5)의 형태를 딴다고 할 때 $Shot$ 간의 Class 별 객체의 개수 변화는 장면의 의미적 접근에 유용한 규칙이 될 수 있다. 연속성을 고려해 Sh_1과 Sh_2의 Class 개수를 리스트에 담아보면 식(6) 와 같이 나타낼 수 있다.

$$Sh_i = ['person' : 3, 'tie' : 2, \cdots \ 'car' : 1]$$
$$Sh_2 = ['person' : 2, 'tie' : 1, \ \cdots \ 'bicycle' : 2, 'car' : 3] \quad (6)$$

$$D_2 = \max(count(Sh_1^{id}), count(Sh_2^{id})) \quad (7)$$

$$ClassSimilarity_{ij} = 1 - \frac{|count(class(Sh_i)) - count(class(Sh_j))|}{D_2} \quad (8)$$

Sh_1이 소유하고 있는 객체의 Class별 개수에서 Sh_2의 개수를 뺀 값을 분자로 한다. 식(7)과 같이 두$Shot$ 중 더 큰 값을 선택해 분모로 적용한다. 또한, 1에서 빼주어 한 Class에 대한 $ClassSimilarity_{ij}$를 구한다. Sh_1와 Sh_2에 속한 $person$ Class로 예시를 들면 아래와 같이 나타낼 수 있다.

$$\frac{count(Sh_1^{person}) - count(Sh_2^{person})}{\max(count(Sh_1^{person}), count(person(Sh_2)))} = 1 - \frac{|3-2|}{3} = 0.666$$

$$Similarity_2 = \frac{\sum_{id=0}^{D_1} ClassCountSimilarity_{id}}{D_1} \quad (9)$$

그런 뒤 다른 identity를 가진 모든 Class를 식(9)와 같이 합하고 D_1으로 나누어 $Similarity_2$를 구한다.

Rule 3. Frame Histogram-based Similarity

$$Sh_i = [hist(F_0), hist(F_1) \cdots \ hist(F_n)] \quad (10)$$

모든 $Shot$에 대한 Frame은 HSV (H:색상, S:채도, V:명도) 형식 중 H값을 사용한다. 이는 영상의 밝기와 채도 값으로 인한 오류를 제거하기 위함이다.

i는 256 범위이며 H(색상)의 히스토그램 인덱스를 나타낸다. 여기서 Sh_1은 마지막 프레임으로 계산하고 Sh_2는 첫 번째 프레임으로 계산한다.

$$D_3 = \max(hist(Sh_1)_i, hist(Sh_2)_i) \quad (11)$$

$$histSimilarity_i = 1 - \frac{|hist(Sh_1)_i - hist(Sh_2)_i|}{D_3} \quad (12)$$

동일한 장면(Scene)에 속하는 $Shot$의 경우 각 $Shot$의 히스토그램(Histogram)끼리의 유사도가 높다. 따라서 유사도를 구하기 위해 Sh_1의 히스토그램과 Sh_2의 히스토그램을 인덱스별로 빼준 뒤 절대값으로 변환하여 합산한다. 그리고 그 값을 max 연산을 통해 더 큰 값을 분모로 가지게 하여 $Shot$들 간의 히스토그램 유사도를 계산한다.

$$Similarity_3 = \frac{\sum^{256} histSimilarity_i}{256} \quad (13)$$

위에서 구한 히스토그램의 모든 범위(i : 0~255)를 합하고 범위의 최대값 255으로 나누어 rule 3.의 $Similarity_3$를 구한다.

Rule 4. Object-based Histogram Similarity

$$Sh_i = \left\{\sum^{256} hist(Ob_0), \sum^{256} hist(Ob_1), \cdots \ \sum^{256} hist(Ob_n)\right\} \quad (13)$$

Sh_i는 $Shot$ 에서 등장하는 객체에 대한 히스토그램 합의 집합이다. Sh_i를 토대로 Sh_1과 Sh_2가 만들어지고 이를 비교해 적용한다. 여기서 Rule 3. 와는 다르게 히스토그램 합을 먼저 계산한 이유는 객체상태의 변화로 인한 오검출을 막기 위함이다.

$$D_4 = \max(id(Sh_i)) \quad (14)$$

동일한 id를 가진 객체에 대해 히스토그램 합의 최대값을 D_4로 받는다.

$$IDSimilarity_k = 1 - \frac{|id(Sh_1) - id(Sh_2)|}{D_4} \quad (15)$$

식(15)와 같이 Sh_1과 Sh_2에서 동일한 id를 가진 객체에 대해서만 히스토그램 비교를 한다. 동일한 id를 가진 객체의 갯수의 최대값을 D_5로 받는다.

$$D_5 = count(id(Sh_i)) \quad (16)$$

$$Similarity_4 = \frac{\sum_{k=0}^{D_5} IDSimilarity_k}{D_5} \quad (17)$$

Sh_1에서 Sh_2를 빼준 값의 절대값을 분자로 취하고, D_5를 적용하여 $Similarity_4$를 도출한다.

3.3 Standardization

3.2의 과정을 통해 4가지 의미적 유사도(Semantic similarity)를 도출했다. 또한 각각에 대한 수학적 모델링을 통해 Class 종류 기반의 유사도 비교, 동일 Class에 속하는 객체의 개수 기반의 유사도 비교, Shot의 Frame간 Histogram 비교, 객체의 관심영역(ROI) 기반의 Histogram 비교 총 4가지 Rule에 대해 수학적으로 정의했다. 4가지 Rule에 의한 의미적 유사도는 (0~1)사이의 범위를 갖지만 Scale의 차이로 인한 오차가 있을 수밖에 없다. 따라서 적절한 표준화 전처리를 통해 오차를 줄이고 $Similarity_{total}$의 도출에 적용한다.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{4}(Similarity_i - \mu)}{n}} \quad (1.1)$$

$$Standardization_i = \frac{Similarity_i - \mu}{\sigma} \quad (1.2)$$

$$Similarity_{total} = \frac{\sum_{i=1}^{4} Standardization_i}{4} \quad (1.3)$$

먼저 4가지 Rule을 통해 얻어진 유사도의 표준편차를 (1.1)로 구하고 이를 (1.2)로 표준화한 유사도 결과를 얻는다. 이렇게 전처리를 거친 4개의 유사도 평균값을 (1.3)으로 최종 유사도($Similarity_{total}$)을 도출한다. 최종 유사도의 결과가 기준값($Hyperparameter$) 이상이면 동일한 장면($Scene$)으로 규정하고 기준값 이하이면 다른 장면($Scene$)에 속한다고 규정한다.

IV. Conclusions

본 논문에서는 장면 분할 위한 의미적 유사도 계산을 위해 4가지 Rule을 모델링 하였다. 또한, 각각의 유사도를 전처리하여 Scale을 조정하고 종합적인 유사도를 도출하는 방식을 제안했다. 본 논문에서는 객체 인식을 통해 추출한 정보를 다양하게 장면 분할에 적용하여 내용 기반의 영상 분석을 위한 토대가 되게 하였다. 4가지 Rule과 최종 유사도 단계에서 장면 분할의 기준이 되는 기준값은 사용자가 임의로 적용해야 하는 $Hyperparameter$이기에 본 논문에서 제안한 방법론을 토대로 실험을 통해 기준값을 얻고 적용할 것이다.

REFERENCES

[1] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[2] Dgkang, cyjo, mhkang, hrbae, jjjung.(2021). “Domain Dependent Object Relationship Generation Technique based onScene Graph Generation Model for Understanding of Video with Story,” KIEE, 1812 - 1813, July 2021.

[3] Sbyun, skjung, “Semantic Segmentation Using Depth Information inIndoor Illumination Changes Environment,” KIISE, 134-138, Feb 2022.