고정 카메라 기반 비디오 모니터링 환경에서 딥러닝 객체 탐지기 결과를 활용한 실시간 전경 및 시설물 추출

이나연, 손승욱, 유승현, 정용화, 박대희
고려대학교 컴퓨터융합소프트웨어학과
email: zpo07050@korea.ac.kr

Real-Time Foreground and Facility Extraction with Deep Learning-based Object Detection Results under Static Camera-based Video Monitoring

Nayeon Lee, Seungwook Son, Seunghyun Yu, Yongwha Chung, Daihee Park
Dept. of Computer Convergence Software, Korea University

요 약

고정 카메라 환경에서 전경과 배경 간 픽셀값의 차를 이용하여 전경을 추출하기 위해서는 정확한 배경 영상이 필요하다. 또한, 프레임마다 변화하는 실제 배경과 맞추기 위해 배경 영상을 지속해서 갱신할 필요가 있다. 본 논문에서는 정확한 배경 영상을 생성하기 위해 실시간 처리가 가능한 딥러닝 기반 객체 탐지기의 결과를 입력받아 영상 처리에 활용함으로써 배경을 생성 및 지속적으로 갱신하고, 획득한 배경 정보를 이용해 전경을 추출하는 방법을 제안한다. 먼저, 고정 카메라에서 획득되는 비디오 데이터에 딥러닝 기반 객체 탐지기를 적용한 박스 단위 객체 탐지 결과를 지속적으로 입력받아 픽셀 단위의 배경 영상을 갱신하고 개선된 배경 영상을 도출한다. 이후, 획득한 배경 영상을 이용하여 더 정확한 전경 영상을 획득한다. 또한, 본 논문에서는 시설물에 가려진 객체를 더 정확히 탐지하기 위해서 전경 영상을 이용하여 시설물 영상을 추출하는 방법을 제안한다. 실제 돈사에 설치된 카메라로부터 획득된 12시간 분량의 비디오를 이용하여 실험한 결과, 제안 방법을 이용한 전경과 시설물 추출이 효과적임을 확인하였다.

1. 서론

주어진 영상에서 배경과 전경을 구분하는 전경 추출(Foreground Extraction)[1]은 가장 기본적인 영상 처리 응용이다. CNN 기반 딥러닝 기술의 발전으로 전경 추출이 가능하나 이는 탐지한 결과인 박스 형태로 나타날 뿐 픽셀단위로 전경을 정확하고 실시간으로 추출하는 방법은 아니다. 하지만 카메라 기반의 비디오 모니터링 환경을 가정할 때 배경 영상만 빠르고 정확히 생성 및 갱신할 수 있다면 실제 전경과 비슷하게 전경 추출을 수행할 수 있다. 또한, 시설물에 의해 가려진 객체를 탐지하지 못하는 오 탐지가 종종 발생하기 때문에, 시설물에 가려진 객체를 더 정확하게 탐지하기 위해 시설물의 위치를 알 수 있는 시설물 영상 정보가 필요하다.

기본적으로 CNN 기반 딥러닝 기술에 의한 전경 추출은 기본적으로 픽셀 단위의 출력을 생성해야 하기 때문에 많은 파라미터가 필요하다. 이에 따라 많은 학습 데이터가 필요하며, 처리 속도가 오래 걸리는 한계가 있다[2]. 본 논문에서는 CNN 기반 객체 탐지기의 결과를 입력받아 영상 처리에 활용함으로써 더 정확한 전경 영상을 획득하고 이를 통해 시설물 정보를 추출하는 것을 제안한다. 즉, 제안 방법은 CNN 기반 객체 탐지기의 박스 단위 결과를 이용하기 때문에, 기존 객체(전경)의 픽셀 단위 출력을 생성하는 CNN 기반 영상 분할기(Image Segmentor) 보다 학습 데이터 생성이 용이하고 처리 속도 측면에서 더 빠르다는 장점이 있다.

본 논문에서는 현재까지 발표된 CNN 기반 객체 탐지기 중 처리 속도 대비 우수한 정확도를 제공하는 것으로 알려진 YOLOv4[3]의 박스 단위 객체 탐지 결과를 활용한다. 구체적으로는 고정 카메라에서 획득되는 비디오 데이터에 YOLOv4를 적용하여 획득한 박스 단위 객체 탐지 결과를 지속적으로 입력받아 픽셀 단위의 배경 영상을 갱신함으로써 개선된 배경 영상을 도출한다. 이후, 이를 활용하여 더 정확한 전경 영상을 획득한다. 또한, 박스 정보와 전경 영상을 이용하여 생성 및 갱신된 시설물 영상을 획득한다. 본 논문에서는 실제 돈사에 설치된 고정 카메라에서 획득된 비디오 데이터를 이용하여 실험한 결과, 제안

방법에 따른 전경 및 시설물 추출이 매우 효과적임을 확인하였다.

2. 제안 방법

본 논문에서는 돼지가 있는 돈방에 tilted-down-view 카메라를 설치하여 영상을 획득하고, 이를 YOLOv4 객체 탐지기에 적용하여 객체 탐지 박스들을 획득한다고 가정한다. 보고된 바에 따르면, YOLOv4는 돈방 내 돼지들을 빠르고 정확히 탐지할 수 있다[4]. 따라서, 고정 카메라에서 획득되는 비디오 데이터에 YOLOv4를 적용한 박스 단위의 객체 탐지 결과를 지속적으로 입력받아 개선된 픽셀 단위의 배경 영상을 획득하고, 이를 활용하여 더 정확한 전경 및 시설물 추출 영상을 획득하는 방법을 제안한다. 그림 1은 본 논문에서 제안하는 방법의 전체 구성도를 나타낸 것으로, YOLOv4에서 얻은 이전 프레임에 대한 탐지 박스 정보를 현재 프레임에 적용하여 배경 영상을 생성하고 배경 차를 이용해 전경 및 시설물 영상을 생성한다.

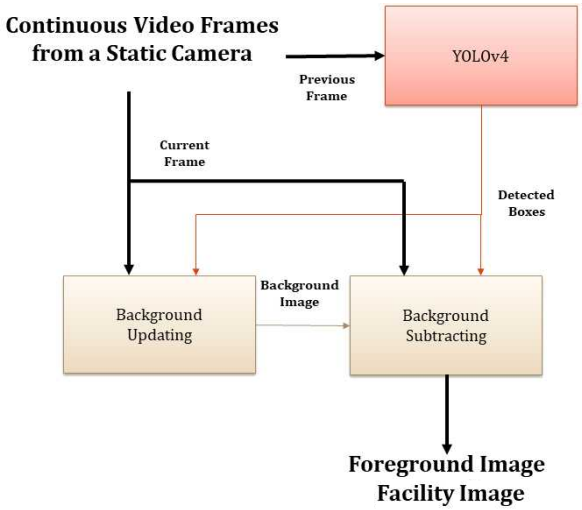


그림 1. 제안 방법의 구성도

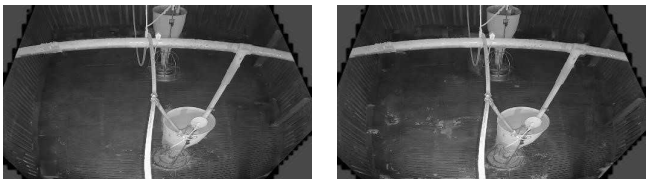
2.1 탐지 박스 정보를 이용한 배경 생성 및 갱신과 전경 영상 추출

먼저 YOLOv4 탐지 결과로 얻은 탐지 박스 정보를 이용하여 현재 프레임의 원본 영상에서 탐지 박스 영역을 제외한 모든 영역의 픽셀 평균값을 구한다. 그 후, 영상의 픽셀 중 현재까지 배경 영상을 갱신하면서 계속해서 탐지 박스 영역에 속했던 픽셀은 배경 영상에 한번도 영향을 주지 않은 것으로 취급하여, 배경 영상의 해당 위치는 현재 영상의 픽셀 평균값으로 대체한다. 또한, 실제 배경이 항상 동일하게 유지되지 않기 때문에 현재 실제 배경에 맞추어 배경 영상을 점차적으로 갱신해야 한다. 따라서, 현재 프레임에서 탐지 박스 영역 외는 각 픽셀마다 이전 프레임에서 생성된 배경 영상과 비교하여 현재 영상의 같은 위치에 있는 픽셀의 값이 더 크면 배경 영상의 픽셀값을 1 증가, 작으면 1 감소시킨다. 그러나 같은 위치에 지속적으로 발생한 오 탐지로 인하여 그림 2처럼 배경 영상에 노이즈가 생성될 수 있다. 이를 방지하기 위해서 일정 프레임 동안 배경과 현재 영상의 같은 위치의 픽셀값이 10 퍼센트 미만으로 차이가 나면 배경의 해당 픽셀값을 해당 위치 픽셀의 배경 기준값으로 설정한다. 즉, 배경과 현재 영상의 픽셀값이 10 퍼센트 이상 차이가 나게 되면 배경 영상을 갱신을 하지 않도록 한다. 본 논문에서는 배경 기준값을 위한 프레임 수를 100으로 설정하여 진행하였다. 마지막으로 배경 차(현재 프레임 영상의 픽셀값과 같은 위치에 있는 현재 배경 영상의 픽셀값 차의 결과)를 통해 전경 영상을 추출한다. 본 논문의 탐지 박스 정보를 이용한 배경 생성 및 갱신과 전경 영상 추출 알고리즘은 Algorithm 1과 같다.

Algorithm 1. Update Background & Extract Foreground

Input : Current_frame = Current video frame
Box_list = Box list for previous frame from YOLOv4
BG = Background image for previous frame
Output : New_bg = Updated background image for current frame
FG = Extracted foreground image for current frame

```
count = accumulated values for each pixel (array form)
acc_value = value for accumulation
img_avg = Current_frame's pixels average
          excluding Box_list position
for( all Current_frame's pixels = Cur_pix ) {
  if( Cur_pix included in Box_list in all frames )
    New_bg's pixel value at same position = img_avg
  else if( Cur_pix included in Box_list position )
    New_bg's pixel value = BG's pixel value(both at same position)
  else {
    if( count[ Cur_pix ] > acc_value )
      if( diff BG's pixel and Cur_pix value < 10% )
        if( Cur_pix > BG's pixel value )
          New_bg's pixel value = BG's pixel value + 1
        else
          New_bg's pixel value = BG's pixel value - 1
    else
      if( diff BG's pixel and Cur_pix value < 10% )
        count[ Cur_pix ] = count[ Cur_pix ] + 1
      else
        count[ Cur_pix ] = 0
      if( Cur_pix value > BG's pixel value )
        New_bg's pixel value = BG's pixel value + 1
      else
        New_bg's pixel value = BG's pixel value - 1 }}
FG = sub New_bg from Current_frame for all pixels
Return New_bg and FG
```



70000th Key Frame　　80000th Key Frame

그림 2. 배경 영상 갱신 중 발생되는 노이즈 예시

2.3 전경 영상을 이용한 시설물 생성 및 갱신

YOLOv4를 사용한 객체 탐지에서 그림 3과 같이 시설물 뒤에 가려진 객체들에서 오 탐지가 발생할 수 있다. 따라서, 본 논문에서는 앞서 제안한 전배경 추출 방법을

이용하여 시설물의 위치를 확인하는데 도움을 주는 시설물 영상 추출 방법을 제안한다.

객체의 전체적인 모습이 시각적으로 보이는 영역은 객체를 가리는 시설물이 있다고 판단하기 힘들기에, 배경차로 구한 전경 영상에서 객체가 존재하는 부분의 정보를 이용한다. 먼저, 전경 영상의 전체 픽셀 평균값과 전경 영상에서 각각의 탐지 박스 영역에 대한 픽셀 평균값을 구한다. 대부분의 전경 영상에서 객체 영역의 픽셀값이 배경 영역보다 높기에 전체 영상과 탐지 박스 영역을 비교했을 때 탐지 박스 영역의 픽셀 평균값이 더 높게 측정된다. 따라서, 탐지 박스 영역의 픽셀 평균값이 전체 영상의 픽셀 평균값보다 낮으면 해당 탐지 박스 영역은 False Positive(객체가 없는데 있다고 판단)로 간주하여 갱신에서 제외한다. 갱신이 진행되는 전경 영상의 탐지 박스 영역에 존재하는 픽셀 중 탐지 박스 영역의 픽셀 평균값보다 높은 곳(객체가 존재하는 곳)에 대해 시설물 영상에서 같은 위치의 픽셀값을 1 감소한다. (모든 부분이 겹친 시설물 영역이 될 가능성이 있기에 겹친 시설물 영상의 초깃값은 모두 255로 설정한다.) 또한, 획득한 배경 영상이 현재 영상의 배경과 완벽히 일치할 수 없어 전경 영상과 시설물 영상에 노이즈를 발생시키기 때문에 일정 프레임마다 보정 작업을 한다. 본 실험에서는 1만 프레임으로 설정하여 진행하였다. 너무 작은 값의 픽셀 변화가 이루어지면 노이즈로 인한 일시적인 결과라고 판단할 수 있기 때문에, 만약 픽셀값이 245 이상(일정 프레임 동안 10 미만 감소)이면 초기값(255)으로 변경한다. 또한, 그림 4와 같이 시설물 내 노이즈로 생긴 구멍이나 외곽선 깎임을 보정하기 위하여 픽셀값이 255인 부분의 좌우와 위아래의 픽셀들의 값을 255로 변경한다. 본 논문의 전경 영상을 이용한 시설물 생성 및 갱신 알고리즘은 Algorithm 2와 같다

Algorithm 2. Update Facility Image

Input : FG = Extracted foreground image for current frame
Box_list = Box list for previous frame from YOLOv4
Facility = Facility image for previous frame
Output : New_facility = Extracted Facility image for current frame

```
img_count = current number of frames
edit_value = value to edit every few frame
img_avg = FG's pixels average
all pixels of Facility copy to New_facility
for( all box in Box_list = Cur_box ) {
  box_avg = Cur_box position's pixels average in FG
  if( box_avg > img_avg ) {
    for( Cur_box's all pixels in FG = Cur_pix ) {
      if( Cur_pix value> box_avg )
        New_facility's pixel value = Facility's pixel value - 1 }}}
if( remainder obtained by dividing img_count by edit_value = 0 ) {
  for( all New_facility's pixels = Cur_pix ) {
    if( Cur_pix's value > 245)
      Cur_pix's value = 255 }
  for( all New_facility's pixels = Cur_pix ) {
    if( Cur_pix = 255 )
      all pixel's value around Cur_pix = 255 }}
Return New_facility
```

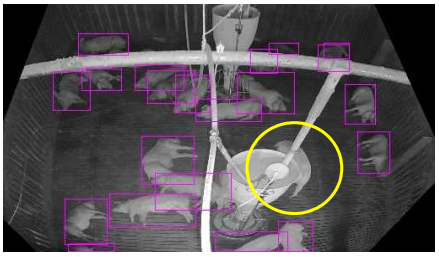


그림 3. 시설물에 가려져 탐지가 되지 않는 예시

그림 4. 시설물 영상 갱신 중 발생되는 노이즈 예시

3. 실험 결과

본 실험은 경상남도 하동군에 위치한 바른양돈 돈사에서, 절반의 영역만 모니터링하는 카메라에서 획득된 비디오로 수행되었다. 카메라는 돈사의 중앙을 기준으로 2.1 m 높이의 기둥에 약 45도 각도로 설치되어있으며, 이를 통해 12시간 분량의 1920×1080 해상도의 돼지들 영상 데이터를 획득하였고, 약 9만장의 키 프레임을 추출하였다[5]. 또한, YOLOv4의 처리 속도를 고려하여 영상의 해상도를 512×288 해상도로 변경하였다. 또한, tilted-down-view로 촬영된 영상으로 인한 앞과 뒤쪽 돼지들의 크기 차이를 완화하기 위해 perspective 변환을 적용하여 top-view 형태로 변경하였다. 마지막으로, 본 실험은 Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, 32GB RAM, GeForce RTX 2080 Ti GPU, CUDA 10.0 환경에서 수행되었다.

그림 5는 1만 번째, 3만 번째, 5만 번째, 7만 번째, 9만 번째 키 프레임에 대한 입력 영상과 배경 갱신 결과이고, 그림 6은 전경 추출 영상과 겹친 시설물 영상 갱신 결과이다. 그림 5에서 1만 번째 키 프레임에 대한 배경 영상을 보면 움직이지 않는 돼지로 인해 배경이 생성되지 못한 부분이 확인된다. 그러나, 시간이 지남에 따라, 돼지가 이동하면서 배경 영상의 대부분이 생성 및 갱신되었고, 점점 노이즈가 감소되는 것을 확인하였다. 즉, 제안한 배경 생성 알고리즘을 이용해 점차적으로 갱신되는 배경 영상을 생성할 수 있는 것을 확인하였다. 또한, 그림 6에서 1만 번째 키 프레임에 대한 전경 추출 영상을 보면 생성되지 못한 배경 부분에서 박스 모양의 노이즈가 발생되는 것을 확인하였으나, 배경영상의 갱신이 진행될수록 이러한 현상이 감소되어 더 선명하고 실제 전경에 근접한 전경 추출 결과를 획득할 수 있었다. 또한, 1만 번째 키 프레임에서는 영상의 대부분이 시설물 영역으로 설정되어 있으나, 시간이 지날수록 시설물의 형태가 잡히는 것이 확인되었다. 그리고 일정 프레임마다 영상을 보정 처리함으로써 처음에 끊겨있던 시설물이 조금씩 이어지는 것을 확인하

였다.

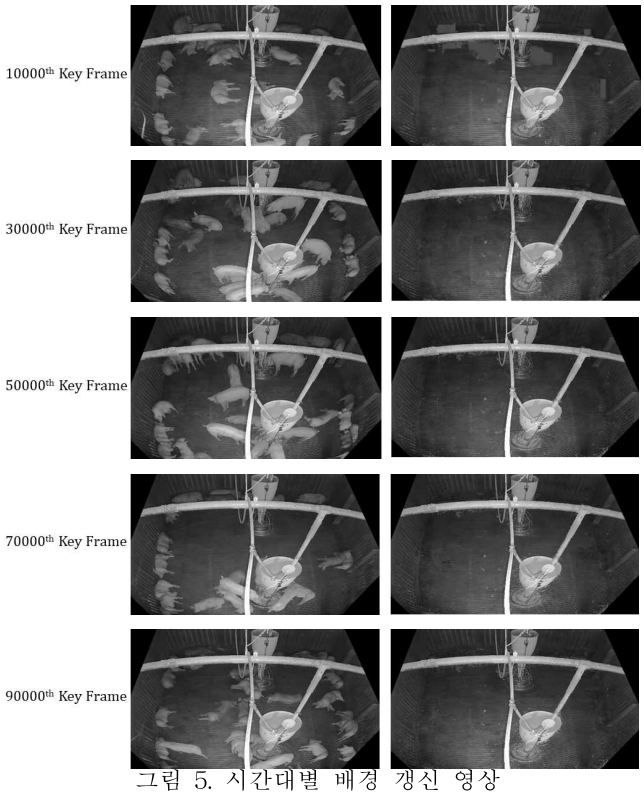


그림 5. 시간대별 배경 갱신 영상

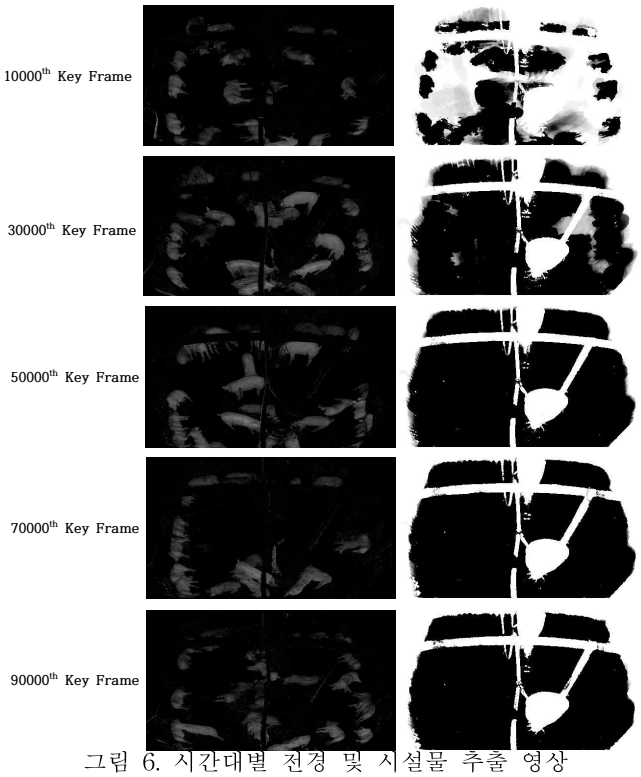


그림 6. 시간대별 전경 및 시설물 추출 영상

마지막으로 동일한 비디오에 대해서 제안 방법의 수행시간을 측정하였다. 기존 YOLOv4에 배경 및 전경 생성 방법을 추가하여 측정하였으며, 시설물 추출에 대한 영상처리 부분도 추가하여 측정하였다. 측정 결과, 두 방법 모두 평균 FPS(1초당 처리하는 프레임 수)가 72.5로 동일하게 측정됨을 확인하였다. 즉, 제안 방법이 처리 속도에 거의 영향을 주지 않으며, 일반 PC 환경에서 실시간 처리가 가능함을 확인하였다.

4. 결론

실시간 비디오 모니터링 응용을 위한 전경 추출을 위해서는 정확도-처리 속도 간의 트레이드오프를 고려해야 한다. 본 논문에서는 고정 카메라에서 획득되는 비디오 데이터에 실시간 처리 속도의 딥러닝 기반 객체 탐지기를 적용한 박스 단위 객체 탐지 결과를 지속적으로 입력받아 픽셀 단위의 배경 영상을 갱신하는 전경 추출 방법과 시설물 추출 방법을 제안하였다. 제안 방법의 효과를 확인하기 위하여 실제 돈사에 설치된 고정 카메라에서 획득된 12시간 분량의 비디오에서 추출한 데이터로 실험한 결과, 제안 방법은 돈사 내 돼지 모니터링을 위한 매우 효과적인 방법이 될 수 있음을 확인하였다. 이후, 본 연구의 제안 방법을 활용하여 전경 영상을 이용한 객체의 탐지 정확도 개선 및 시설물에 가려진 객체에 대한 탐지에 대한 개선 방법을 연구할 계획이다.

감사의 글

본 논문은 2020년도 과학기술정보통신부의 재원으로 연구개발특구진흥재단의 과학벨트성과확산지원사업(1711123920)과 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-현장맞춤형 이공계 인재양성 지원사업의 지원을 받아 수행된 연구임 (No. 2019H1D8A1109907) 지원으로 수행된 연구결과임.

참고 문헌

[1] I. Setitra and S. Larabi, “Background Subtraction Algorithms with Post Processing: A Review,” *Proc. of ICPR*, 2014.

[2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, W. Liu, and M. Piekikäinen, “Deep Learning for Generic Object Detection: A Survey,” *International Journal of Computer Vision*, Vol. 128, pp. 261-318, 2020.

[3] A. Bochkovskiy, C. Wang, and H. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 1*arXiv preprint*, arXiv:2004.10934, 2020.

[4] H. Ahn, S. Son, H. Kim, S. Lee, Y. Chung, and D. Park, “EnsemblePigDet: Ensemble Deep Learning for Accurate Pig Detection,” *Applied Sciences*, Vol. 11, pp. 5577, 2021.

[5] S. Yu, S. Son, H. Ahn, Y. Chung, and D. Park, “비디오에서 키프레임 추출을 위한 파라미터 설정,” *Proc. of IPIU*, 2021.