

# Image Processing and Deep Learning Techniques for Fast Pig's Posture Determining and Head Removal

Hanse Ahn<sup>†</sup> · Wonseok Choi<sup>\*\*</sup> · Sunhwa Park<sup>\*\*\*</sup> · Yongwha Chung<sup>\*\*\*\*</sup> · Daihee Park<sup>\*\*\*\*\*</sup>

## ABSTRACT

The weight of pig is one of the main factors in determining the health and growth state of pigs, their shipment, the breeding environment, and the ration of feed, and thus measuring the pig's weight is an important issue in productivity perspective. In order to estimate the pig's weight by using the number of pig's pixels from images, acquired from a Top-view camera, the posture determining and the head removal from images are necessary to measure the accurate number of pixels. In this research, we propose the fast and accurate method to determine the pig's posture by using a fast image processing technique, find the head location by using a fast deep learning technique, and remove pig's head by using light weighted image processing technique. First, we determine the pig's posture by comparing the length from the center of the pig's body to the outline of the pig in the binary image. Then, we train the location of pig's head, body, and hip in images using YOLO(one of the fast deep learning based object detector), and then we obtain the location of pig's head and remove an outside area of head by using head location. Finally, we find the boundary of head and body by using Convex-hull, and we remove pig's head. In the Experiment result, we confirmed that the pig's posture was determined with an accuracy of 0.98 and a processing speed of 250.00fps, and the pig's head was removed with an accuracy of 0.96 and a processing speed of 48.97fps.

**Keywords :** Real-Time Pig Monitoring, Posture Determining, Head Removal, Image Processing, Deep Learning, YOLO

# 돼지의 빠른 자세 결정과 머리 제거를 위한 영상처리 및 딥러닝 기법

안 한 세<sup>†</sup> · 최 원 석<sup>\*\*</sup> · 박 선 화<sup>\*\*\*</sup> · 정 옹 화<sup>\*\*\*\*</sup> · 박 대 희<sup>\*\*\*\*\*</sup>

## 요 약

양돈 업계에서 돼지의 무게는 돼지의 건강이나 성장 상태, 출하 여부, 사육 환경, 사료 배급을 결정하는 주요 요인 중 하나이며, 따라서 돼지의 무게를 측정하는 것은 돼지의 생산성 측면에서 중요한 문제이다. Top-view 카메라에서 획득한 영상으로부터 돼지의 픽셀 수를 이용하여 돼지의 무게를 추정하고자 할 때, 정확한 픽셀 수 측정에 영향을 주는 돼지의 자세를 결정할 필요가 있으며, 픽셀 수 측정에 영향을 주는 머리 부분을 제거할 필요가 있다. 본 논문에서는 빠른 영상처리 기법을 이용하여 돼지의 자세를 빠르게 결정하고, 딥러닝 기반의 빠른 객체탐지 기법인 YOLO를 이용하여 돼지 머리 위치를 파악한 후, 경량화된 영상처리 기법을 이용하여 돼지의 머리와 몸통 경계를 획득하고 머리를 제거하는 방법을 제안한다. 즉, 빠른 영상처리 기법으로 이진화된 돼지의 영상 데이터에서 돼지의 몸통 중심점으로부터 돼지의 외곽선까지의 길이를 비교하여 돼지의 자세를 결정한다. 또한, 돼지의 머리 위치를 탐지하기 위하여 YOLO를 이용하여 영상 데이터 내의 돼지의 머리, 몸통, 엉덩이의 위치를 학습시킨 후, 끝은 자세의 돼지 머리 위치를 획득하고 머리 바깥 영역을 제거한다. 마지막으로 Convex-hull을 이용하여 돼지의 머리와 몸통 경계를 추정한 후, 머리를 제거한다. 실험 결과, 0.98의 정확도와 250.00fps의 수행속도로 돼지의 자세를 결정하였으며, 0.96의 정확도와 48.97fps의 수행속도로 돼지의 머리탐지 및 제거가 가능함을 확인하였다.

**키워드 :** 실시간 돼지 모니터링, 자세 결정, 머리 제거, 영상처리, 딥러닝, YOLO

※ 본 연구는 2018년도 교육부의 지원으로 한국연구재단의 기초연구사업(2018R1D1A1A09081924)으로 수행된 연구결과임.  
※ 이 논문은 2019년도 한국정보처리학회 춘계학술발표대회에서 “돼지 머리 제거를 위한 영상처리”의 제목으로 발표된 논문을 확장한 것임.  
† 준 회 원 : 고려대학교 컴퓨터정보학과 석사과정  
\*\* 준 회 원 : 고려대학교 컴퓨터융합소프트웨어학과 학사과정  
\*\*\* 비 회 원 : 고려대학교 컴퓨터융합소프트웨어학과 학사과정  
\*\*\*\* 중신회원 : 고려대학교 컴퓨터융합소프트웨어학과 교수  
\*\*\*\*\* 정 회 원 : 고려대학교 컴퓨터융합소프트웨어학과 교수  
Manuscript Received : July 8, 2019  
Accepted : August 20, 2019  
\* Corresponding Author : Yongwha Chung(ychungy@korea.ac.kr)

## 1. 서 론

한국농촌경제연구원에 따르면 한국의 양돈 생산액은 2018년 기준으로 약 7조 원을 기록하였다[1]. 이는 쌀의 생산액인 약 8조 원에 이어 농업 생산액 2위에 이를 정도로 대한민국의 중요한 산업이며, 따라서 양돈 업계에서는 돼지의 생산성을 증대시키기 위하여 많은 노력을 기울이고 있다. 돼지의 생산성에 영향을 주는 여러 요인 중 돼지의 무게는 돼지의 건

강 및 성장 상태, 적정 무게 도달 여부를 통한 돼지의 출하 여부 및 사료 배급량을 결정하며[2, 3], 특히 가성비 기준 적정 출하 시점과 저성장 돈을 결정하는 중요한 요인이다. 일반적으로 돼지의 무게는 저울을 이용하여 측정하나, 저울을 이용한 무게 측정은 노동력을 많이 소모할 뿐만 아니라 많은 시간을 소요하며, 돼지와 접촉하면서 부상이 생길 수 있는 위험한 작업이기도 하다. 또한, 국내 양돈 농가의 환경은 관리인 1명당 약 1,800마리에 이르는 규모의 돼지들을 관리하기 때문에, 약 100kg에 육박하는 모든 돼지의 무게를 직접 측정하는 것은 현실적으로 불가능하다.

앞서 언급한 문제점을 해결하기 위하여 관리인이 직접 돼지의 무게를 측정하는 방법 대신, 돈사 천장에 Top-view 카메라를 설치하고 카메라로부터 획득한 영상정보에 영상처리 기법을 이용하여 돼지의 무게를 추정하는 다양한 방법들이 제안되었다[4-11]. 예를 들어, 돈사 천장에서 획득된 비디오로부터 돼지의 픽셀 수를 측정하고 선형회귀함수를 이용하여 돼지의 무게를 추정한 결과, 실제 돼지 무게의 2kg 내외 오차로 무게추정이 가능함이 보고되었다[4, 5]. 그러나 대부분의 영상처리 논문처럼 이러한 돼지 무게추정 연구에서는 수행시간에 대한 언급이 없다는 문제가 있다. 따라서, 본 연구진이 Morphological 연산을 이용한 돼지 머리 제거 방법[2]과 Medial Axis Transform[12]을 이용한 돼지 머리 제거 방법[9]을 영상처리 라이브러리인 OpenCV[13]로 구현하여 수행시간을 측정된 결과, 비디오 프레임 한 장당 약 1초 이상의 시간이 필요함을 확인하였다.

또한, 최근의 딥러닝 기법을 적용하여 돼지의 자세를 결정하여 곧은 자세의 돼지를 획득한 후, 머리를 제거하여 각 돼지의 무게를 추정할 수 있다. 예를 들어, 돼지의 자세를 결정하기 위하여 OpenPose[14], PoseNet[15] 등 자세 결정을 위한 딥 네트워크나 Mask R-CNN[16]과 같은 영상 분할/자세 결정 등 다양한 영상처리 문제에 적용할 수 있는 딥 네트워크들을 이용하여 돼지의 자세를 결정할 수 있다. 그러나 이러한 End-to-End 방식의 딥 네트워크들은 수백만에서 수천만 개에 이르는 가중치 값들을 가지고 있어, GPGPU를 이용하여 수행시간을 최소화한 경우에도 비디오 프레임 한 장당 최소 수십에서 수백 msec가 소요되는 문제가 있다.

앞서 언급한 기존 방법들의 수행시간 문제를 해결하기 위하여, 본 논문에서는 실시간 돼지 모니터링 환경[17-20]에 적합한 돼지 자세 결정 및 머리 제거 방법을 제안한다. 즉, 영상처리 기법인 연결요소 레이블링[21]을 이용하여 돼지 몸통의 중심점을 획득한 후, 중심점과 돼지의 외곽선까지의 길이를 비교하여 돼지의 자세가 곧은 자세인지, 굽은 자세인지를 결정한다. 또한, 돼지의 머리 위치를 획득하기 위하여 딥러닝 기반의 빠른 객체탐지 기법인 YOLO(You Only Look Once)[22]를 이용하여 돼지의 머리, 몸통, 엉덩이 위치를 학습시킨 후, 곧은 자세의 돼지에 대하여 머리 위치를 획득하고 머리 위치를 기준으로 일정 길이 이상의 바깥 영역을 제거한다. 마지막으로 Convex-hull[23]을 경량화하여 돼지 머리 영역의 오목 점 위치를 탐지한 후, 중심점으로부터 각 오목 점

까지의 거리 정보를 이용하여 머리와 몸통의 경계를 추정한 후 돼지의 머리를 제거한다.

요약하면, 본 연구의 목표는 빠른 영상처리 기법과 딥러닝 기법을 적절히 활용하여, 실시간 돼지 모니터링 환경에서 동작할 수 있는 돼지 자세 결정 및 머리 제거 방법을 제안하는 것이다. 특히, 각 돼지별 정확한 픽셀 수를 측정하기 위해, 본 연구진이 수년간 진행한 실시간 돼지 모니터링 결과[17-20]를 활용한다. 즉, 입력된 비디오 프레임에서 전·배경 분리를 수행한 후, 돼지들이 근접 및 겹침 상태가 아닌 경우에 한하여 본 연구에서 제안하는 자세 결정 및 머리 제거 방법을 적용하는 것으로 가정한다. 또한, 약 100kg에 육박하는 실제 돼지들의 무게 측정에 여러 어려움이 있어, 본 연구에서는 각 비디오 프레임 내 개별 돼지들이 픽셀 수 측정에 적합한 자세인지 결정하고 적합한 자세인 돼지들에 대한 머리 제거 방법의 정확도와 수행속도를 평가한다.

본 논문의 구성은 다음과 같다. 2장에서는 돼지의 자세 결정과 머리 제거에 관련된 기존 연구들을 간략하게 소개하고, 3장에서는 본 연구에서 제안하는 돼지의 자세 결정과 머리탐지 및 제거 방법에 관하여 기술한다. 4장에서는 제안한 방법으로 실험한 결과를 객관적인 지표로 평가하고, 마지막으로 5장에서 결론 및 향후 연구에 대하여 언급한다.

## 2. 관련 연구

직접적인 돼지 무게 측정의 어려움을 극복하기 위하여 영상 데이터를 이용하여 간접적으로 무게를 추정하는 제안 방법 중 Top-view 카메라로부터 획득한 영상정보에서 돼지의 픽셀 수를 측정된 후, 이를 이용하여 추정한 돼지의 무게와 실제 측정된 무게와의 관계를 보여주는 제안 방법이 있다[4, 5]. 돼지의 무게추정을 위해 픽셀 수를 이용하기 위해서는 정확한 픽셀 수 측정에 영향을 줄 수 있는 요인들을 제거할 필요가 있으며, 이러한 요인 중에는 돼지의 자세와 머리 위치가 있다. 돼지는 생활하면서 곧은 자세 또는 굽은 자세를 취하며, 자세별 픽셀 수 측정 결과, 굽은 자세를 취한 돼지는 선 자세를 취한 돼지에 비하여 적게 측정이 되었다. 또한, 일반적으로 돼지는 머리를 숙이거나 들고 있으며, 머리 위치에 따른 픽셀 수 측정 결과 머리를 숙인 상태의 돼지는 머리를 든 상태의 돼지에 비하여 적게 측정이 되었다. 따라서 정확한 픽셀 수 측정을 위해서는 영상 데이터에서 돼지의 자세를 결정하여 곧은 자세의 돼지 영상 데이터를 획득한 후, 머리를 제거할 필요가 있다.

영상 데이터에서 돼지의 머리를 제거하기 위하여 영상처리 기법인 Morphological 연산 기법을 이용하는 방법이 제안되었다[2]. 이 방법은  $s \times s$  크기의 원형 커널을 사용하여 영상처리 기법인 Morphological Open 연산을 수행하는 방법으로 이진화 기법[24]을 이용한 전·배경 분리 및 잡음 제거를 제외한 별도의 전처리 과정이 필요 없다는 장점이 있다. 그러나 이 방법은 돼지의 크기와 커널의 크기에 따라 정확도에 큰 영향을 주기 때문에 돼지의 크기에 따라서 커널의 크기를

개별적으로 설정할 필요가 있다. 또한, 앞서 언급한 커널 크기에 따라 발생하는 정확도 문제를 최소화하기 위해 복수의 커널을 설정하여 머리를 제거하는 경우, 커널의 수와 크기가 증가할수록 연산량이 증가하며, 이에 따라 수행시간이 많이 소요되는 문제가 있다. 다른 연구로는, 영상처리 기법인 Medial Axis Transform 기법을 이용하여 돼지의 머리를 제거하는 방법[9]이 제안되었으나, Medial Axis Transform 기법을 수행하는 과정에서 연산량이 증가하여 수행시간이 많이 소요되는 문제가 있다.

또한, 카메라로부터 획득한 영상 데이터를 이용하여 영상 데이터 내 객체들의 자세를 결정하기 위하여 OpenPose[14], PoseNet[15], Mask R-CNN[16] 등이 제안되었다. 이러한 End-to-End 딥러닝 기법은 별도의 전처리 과정이 필요 없다는 장점이 있으나, 자세 추정을 위하여 픽셀 각각에 대한 결정을 해야 하기 때문에 연산량이 많다는 문제를 가지고 있다.

본 논문에서는 이러한 돼지의 자세 결정과 머리 제거 문제의 어려움을 극복하기 위하여, End-to-End 딥러닝 기반의 자세 추정 대신 빠른 영상처리 기법을 제안하며, 연산량이 많은 기존의 영상처리 기반 머리 제거 대신 빠른 딥러닝 기반 객체탐지 및 후처리 기법을 제안한다.

### 3. 제안 방법

#### 3.1 전처리

본 논문에서는 1280 × 720 해상도의 비디오(초당 30프레임)에서 기준으로 하며, 3.2 m 높이에서 Top-view 카메라가 설치된 환경을 가정한다. 또한, 영상 데이터 내 돼지의 위치를 파악하기 위하여 시공간 보강 기법을 적용하여 전·배경 분리를 한 영상 데이터를 이용한다[18]. 입력받은 영상 데이터에 이진화 기법과[24] 미디언 블러 기법을 적용하여 영상 데이터를 이진화하고 잡음 신호를 최소화한다. 이후 영상 데이터 내 돼지들을 객체화시키기 위하여 연결요소 레이블링 기법[21]을 적용하여 돼지들을 객체화하고 각 돼지의 중심점 위치 및 픽셀 수를 획득한다. 각 레이블링 된 돼지 객체들에 대하여 객체의 픽셀 수가 일정 크기 이상(약 30,000픽셀)인 객체들은 근접 및 겹침 돼지로 판단하며, 일정 크기 이하(약 15,000픽셀)의 객체들은 잡음 신호 또는 조명 상태에 따라 이진화 기법 수행 결과가 픽셀 수를 측정하기 좋지 않은 돼지로 판단한다. 이후 앞서 획득한 레이블링 결과를 이용하여 이진화 기법 수행 결과가 양호하며, 근접 및 겹침 돼지가 아닌 개별 돼지들을 획득한다. Fig. 1은 전처리 과정 수행 후 획득한 개별 돼지의 영상 데이터를 보여준다.

#### 3.2 돼지의 자세 결정

돼지의 자세 결정은 중심점으로부터 외곽선까지의 길이를 비교하는 방법을 사용한다[10]. 즉, 중심점으로부터 수직 및 수평인 일직선을 생성하였을 때, 곧은 자세의 돼지는 중심점을 기준으로 외곽선까지 둘로 나뉜 직선의 비율이 약 1 : 1로 수렴하는 반면, 굽은 자세의 돼지는 그렇지 않은 결과를

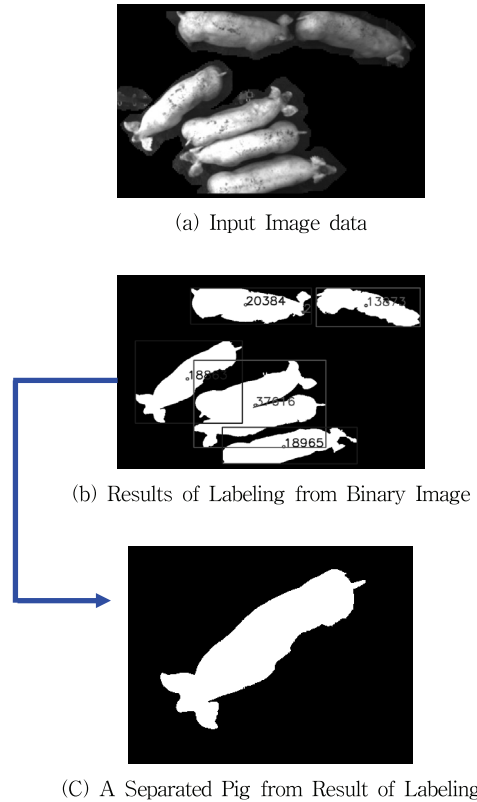


Fig. 1. Results of Preprocessing

보인다는 사실을 이용한다. 돼지의 자세를 결정하기 위하여 앞서 연결요소 레이블링 기법을 이용하여 획득한 개별 돼지의 중심점으로부터 x축 및 y축과 평행한 방향으로 돼지를 지나는 두 직선을 생성한다. 이후 생성된 각 직선에 대하여 중심점을 기준으로 나뉜 두 직선의 길이를 비교하여 나뉜 직선의 길이의 차이가 생성된 직선 길이의 약 0.2 배 이상일 경우 굽은 자세로 판단하며, 그 이하일 경우 곧은 자세로 판단한다. Fig. 2는 각각 굽은 자세의 돼지와 곧은 자세의 돼지에 대하여 자세 결정 알고리즘을 적용하여 자세를 결정한 결과를 보여주며, Fig. 3은 돼지의 자세 결정 알고리즘을 의사 코드로 나타낸 것이다.

#### 3.3 돼지의 머리 위치 탐지

돼지의 머리 위치를 탐지하기 위하여 본 논문에서는 딥러닝 기술 중 실시간 처리가 가능한 객체탐지 기법인 YOLO [22]를 이용한다. YOLO는 네트워크의 최종 출력단에서 경계박스(Bounding Box)의 위치 찾기 및 클래스 분류가 동시에 수행된다. 하나의 네트워크로 특징 추출 및 경계박스 생성과 동시에 클래스를 분류하므로 구조가 간단하며, 수행시간이 빠르다는 장점이 있다. 본 논문에서는 돼지의 머리 위치를 탐지하기 위하여 영상 데이터 내 돼지의 머리 위치와 정확도를 높이기 위하여 몸통 및 엉덩이 위치를 동시에 학습시켰다. 머리 위치 경계박스를 획득하는 모델을 생성하기 위하여 1,000장의 영상 데이터를 Image Augmentation을 사용하여 상하, 좌우, 상하좌우가 반전된 영상 데이터를 만들어 총 4,000장의

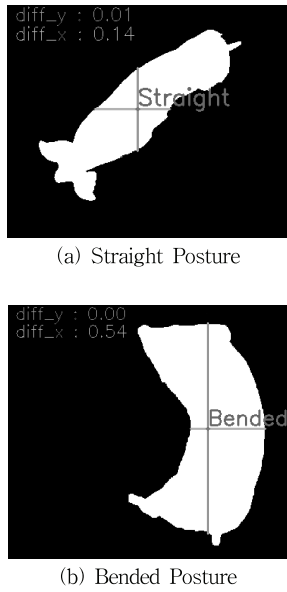


Fig. 2. Results of Posture Determining

Algorithm 1. Pig's Posture Determining
<p><b>Input</b>                      y = y coordinate of center point                      x = x coordinate of center point                      lableTop = most top y coordinate of label                      lableBottom = most bottom y coordinate of label                      lableLeft = most left x coordinate of label                      lableRight = most right x coordinate of label                      colour[y][x] = brightness of image at (y, x);</p> <p><b>Output</b>                      result : pig posture estimation result</p>
<p><b>Algorithm</b>                      set startPointX, endPointX to 0;                      set startPointY, endPointY to 0;                      for p = lableLeft to lableRight                        if (colour[y][p] != 0 &amp; startPointX = 0)                          set startPointX to p;                        if (colour[y][p] = 0 &amp; startPointX != 0)                          set endPointX to p;                        break;                      for p = lableTop to lableBottom                        if (colour[p][x] != 0 &amp; startPointY = 0)                          set startPointY to p;                        if (colour[p][x] = 0 &amp; startPointY != 0)                          set endPointY to p;                        break;                      top = (y - startPointY);                      bottom = (endPointY - y);                      diff_y =   top - bottom   /                                (endPointY - startPointY);                      left = (x - startPointX);                      right = (endPointX - x);                      diff_x =   left - right   /                                (endPointX - startPointX);                      if diff_y &gt; 0.2    diff_x &gt; 0.2                        set result to 'bended';                      else                        set result to 'straight';                      return result;</p>

Fig. 3. An Algorithm for Pig's Posture Determining

영상 데이터를 학습한다. 그 후, 영상 데이터에서의 각 돼지 머리 위치 경계박스를 생성한 후, 경계박스의 좌표를 텍스트 파일로 생성하였으며, 경계박스의 중심점을 이용하여 곧은 자세의 각 돼지의 머리 위치를 획득한다. 이후 정확한 머리경계 추정 및 연산량을 최소화하기 위하여 곧은 자세로 결정된 돼지에 대하여 앞서 획득한 머리 위치를 기준으로 생성된 경계박스 영역 외의 바깥 영역을 제거한다. Fig. 4는 YOLO를 통해 획득한 머리 영역의 경계박스와 머리 바깥 영역을 제거한 결과를 보여준다.

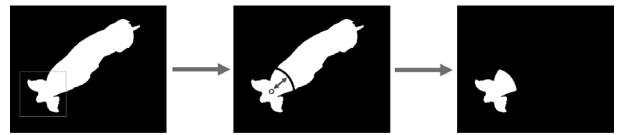


Fig. 4. Results of Outside Area Removing

### 3.4 돼지의 머리 경계추정 및 제거

앞서 획득한 돼지의 머리 영역 영상 데이터에 에지 검출 [25]을 수행하여 에지 정보를 도출한 후, 이를 이용하여 돼지의 외곽선 정보를 획득한다. 그 후 앞서 획득한 외곽선 정보를 이용하여 Convex-hull을 수행하며, 일정 크기 이상의 Convex-hull 다각형의 불록함을 계산하여 오목 점의 위치를 획득한다[26]. Fig. 5는 머리 영역 영상 데이터에 Convex-hull을 적용하여 획득한 머리 영역의 오목 점과 Convex-hull 다각형의 불록함을 픽셀 수로 수치화하여 보여준 것이며, Fig. 6은 Convex-hull을 이용한 오목 점 탐지 알고리즘을 의사 코드로 나타낸 것이다.

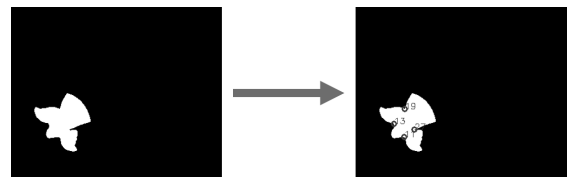


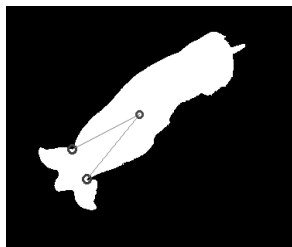
Fig. 5. Results of Pig's Head Convexity Detection

앞서 획득한 Convex-hull 다각형의 오목 점 중 돼지의 머리와 몸통의 경계로 활용할 수 있는 오목 점은 돼지의 귀 부분에 존재하며, 이러한 오목 점들은 돼지의 몸통 중심점에서 가장 가까운 두 오목 점이라는 특징을 가진다. 먼저, 돼지의 머리와 몸통의 경계를 획득하기 위해서 일정 불록함(약 9픽셀) 이상의 Convex-hull 다각형의 오목 점 중 중심점으로부터 가장 가까운 오목 점을 획득한다. 그 후 일정 불록함(약 7 픽셀) 이상의 Convex-hull 다각형의 오목 점 중 앞서 획득한 점과 중심점과 일정 각도(약 45°~105°)를 이루는 가장 가까운 오목 점을 획득한다. 앞서 획득한 두 오목 점을 이진화된 원본 돼지 영상 데이터에 적용하여 돼지의 머리와 몸통을 분리한다. 그 후, 연결요소 레이블링 기법을 이용하여 영상 데이터 내 가장 작은 영역인 돼지의 머리를 제거한다. Fig. 7은 머리 제거 알고리즘을 적용하여 머리와 몸통의 경계를 추정

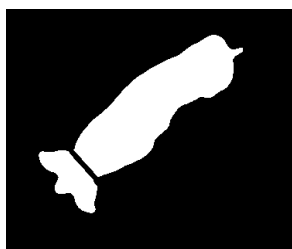
하고 분리한 후 머리가 제거된 돼지를 보여준 것이며, Fig. 8은 돼지 머리 제거 알고리즘을 의사 코드로 나타낸 것이다.

Algorithm 2. Pig's Head Convexity Detection
<p><b>Input</b> binary image of pig's head area</p> <p><b>Output</b> contours : point Vector Array of contours defects : array of convexity defects index : index where contour area is maximum</p>
<p><b>Algorithm</b> find edge from input image; find contour vector using edge; maxarea = 0; for i = 0 to i &lt; size of contours do   if contourArea(contours[i]) &gt; maxarea then     maxarea = contourArea(contours[i]);     index = i; for i = 0 to i &lt; size of contours do   find convex Hull in contours[i];   if size of hull[i] &gt; 3 then     convexity detection; return contours, defects, index;</p>

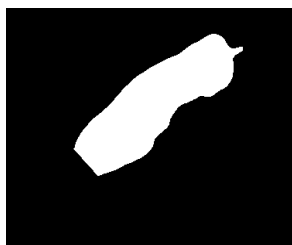
Fig. 6. An Algorithm for Pig's Head Convexity Detection



(a) Estimated Boundary



(b) Separated Head from Body



(c) Head Removed Pig

Fig. 7. Result of Head Removal

Algorithm 3. Pig's Head Removal
<p><b>Input</b> binary image of single pig center : coordinate of pig's center contours : point vector array of contours defects : array of convexity defects index : index where contour area is maximum</p> <p><b>Output</b> frame : binary image of single pig's body</p>
<p><b>Algorithm</b> set min_dist to enough large integer value; for j = 0 to size of defects[index] do   Vec4i v = defects[index][j];   depth = v[3] / 256.0;   if 9 &lt;= depth then     faridx = v[2]     ptFar = contours[index][faridx]     dist = distance from center to faridx;     if min_dist &gt; dist then       min_dist = dist;       Point A = ptFar; set min_dist to enough large integer value; for j = 0 to j &lt; size of defects[index] do   Vec4i v = defects[index][j];   depth = v[3] / 256.0;   if 7 &lt;= depth then     if contours[index][v[2]] == Point A then       continue;     faridx = v[2]     ptFar = contours[index][faridx]     dist = distance from center to faridx;     angle = angle from center, Point A, ptFar;     if min_dist &gt; dist &amp;&amp; 45 &lt; angle &lt; 105 then       min_dist = dist;       Point B = contours[index][v[2]]; draw line between Point A to Point B in input image; find smallest area using connect component method; smallest area removing; return frame;</p>

Fig. 8. An Algorithm for Pig's Head Removal

#### 4. 실험 결과

본 실험은 충청북도 청주시에 위치한 충북대학교 내에 9마리 돼지(110~115일령, 90~98kg, Duroc × Landrace × Yorkshire)가 활동하는 돈사에서 수행되었다. 영상 데이터 촬영을 위해 돈사의 중앙을 기준으로 3.2m 높이의 천장에 Intel® RealSense D435 카메라를 설치하였고 이를 통해 1280 × 720 해상도의 돼지들 영상 데이터를 획득하였다. 또한, 본 실험은 Ubuntu 16.04 LTS OS, Intel Core i7-7700K 4.20 GHz CPU, NVIDIA GeForce GTX 1080 Ti(3584 CUDA cores, 11 GB VRAM) GPU, 32 GB RAM, Visual Studio 2017, 영상처리 라이브러리 OpenCV 3.4[13] 환경에서 수행되었다.

본 논문에서는 전처리 과정을 수행하기 위하여 시공간 보강 기법[18]을 적용하여 전·배경 분리를 한 150장의 곁은 자세의 돼지 영상 데이터와 150장의 곁은 자세의 돼지 영상 데이터, 합계 300장의 영상 데이터를 이용하였다. 300장의 영상

데이터에 이진화 기법과 미디언 블러 기법을 적용하여 영상 데이터를 이진화하고 잡음 신호를 제거하였으며, 연결요소 레이블링 기법을 사용하여 각 돼지를 객체화하고 중심점의 좌표를 획득하였다. 이후 돼지의 자세를 결정하기 위하여 중심점으로부터 x축과 y축에 평행한 두 직선을 생성하여 중심점을 기준으로 외곽선까지 둘로 나뉜 직선의 길이 차이와 생성된 직선 길이의 비율을 측정하여 돼지의 자세를 결정하였다[10].

돼지의 머리탐지를 위한 YOLO[22]의 학습데이터는 수집된 원본 영상 데이터 중 1,000장의 영상 데이터를 Image Augmentation을 사용하여 상하, 좌우, 상하좌우가 반전된 영상 데이터를 만들어 총 4,000장의 영상 데이터를 생성하여 사용하였다. 또한, 돼지머리 탐지의 정확도를 높이기 위하여 돼지의 머리와 몸통 및 엉덩이 위치를 동시에 학습시켰다. 테스트 데이터는 학습데이터로 사용되지 않은 앞서 전처리 과정을 수행한 300장의 영상 데이터를 사용하여 테스트를 진행하여 돼지의 머리 위치 경계박스 좌표를 획득하였으며, 경계박스의 중심점을 이용하여 각 돼지의 머리 위치를 획득하였다. 이후 곧은 자세로 결정된 돼지들에 대하여 앞서 획득한 머리 위치를 기준으로 생성된 경계상자 영역 외의 바깥 영역을 제거하였다.

앞서 획득한 돼지 머리 영역에 Convex-hull을 수행하여 Convex-hull 다각형의 오목 점의 좌표와 볼록함 값을 획득하였다[11]. 이후 Convex-hull을 통해 획득한 오목 점과 중심점과의 거리 정보를 이용하여 머리와 몸통의 경계를 탐지하고 분리하였다. 마지막으로 연결요소 레이블링 기법을 통해 영상 데이터 내 가장 작은 영역인 돼지의 머리를 제거하였다. Fig. 9는 본 논문의 제안 방법을 적용하여 돼지의 자세를 결정하고 머리를 제거한 결과를 보여준다.

기존 방법[2, 14]과 제안 방법 간 성능 평가를 위해 돼지 영상 데이터 300장에 대한 자세 결정과 머리 제거 Ground-truth를 생성하였다. 자세 결정의 경우 참 긍정(TP)은 예측된 자세가 돼지의 실제 자세와 같을 경우, 거짓 긍정(FP)은 돼지

의 자세와 예측된 자세가 다를 경우로 정의하였다. 머리 제거의 경우 참 긍정(TP)은 돼지의 머리 부분을 제거하였을 경우, 거짓 긍정(FP)은 돼지의 머리 부분이 아닌 곳을 제거하였을 경우, 거짓 부정(FN)은 머리 부분을 제거하지 못한 경우로 정의하였으며, TP, FP, FN을 측정하기 위하여 제안 방법을 적용하여 머리를 제거한 돼지 영상 데이터의 픽셀 수와 앞서 생성한 Ground-truth의 픽셀 수를 비교하여 측정하였다. 본 논문의 자세 결정 방법과 머리 제거 방법의 정확도(ACC)는 (1)과 같이 계산하였으며, 수행속도를 계산하기 위하여 측정된 수행시간을 fps(frame per second)로 계산하였다. 또한, 정확도와 수행시간을 합한 성능 지표를 획득하기 위하여 정확도와 수행속도를 곱한 값을 계산하였다.

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

실험 결과, 돼지 영상 데이터 300장에 대하여 자세 결정의 경우 영상 데이터 한 장당 평균 0.98의 정확도로 돼지의 자세를 결정할 수 있었으며, 수행속도는 평균 250.00fps로 측정되었다. 반면, 기존 딥러닝 방법[14]은 300장에 대하여 영상 데이터 한 장당 평균 0.89의 정확도로 자세 결정에 성공하였으며, 수행속도는 평균 12.99fps로 측정되었다. 서론에서 설명하였듯이 일반적인 End-to-End 딥러닝 기법으로 자세를 추정하려면 많은 계산량이 필요하지만, 본 연구에서는 매우 빠르고 효과적인 영상처리 기법을 적용하여 딥러닝 기법에 비하여 약 20배 빠른 처리속도를 제공할 수 있음을 확인하였다.

또한, 본 논문이 제안한 자세 결정 방법 수행 결과, 영상 데이터 1장당 평균 약 2.40 마리의 곧은 자세 돼지를 획득하였다. 이를 이용하여 머리탐지 및 제거를 수행한 결과 영상 데이터 한 장당 평균 0.96의 정확도로 돼지의 머리를 탐지 및 제거할 수 있었으며, 수행속도는 평균 48.97fps로 측정되었다. 반면, 기존 영상처리 방법[2]은 영상 데이터 한 장당 평균 0.94의 정확도로 머리탐지 및 제거에 성공하였으며, 수행속도는 평균 0.72fps로 측정되었다. 서론에서 설명하였듯이 Morphological 연산을 이용하거나 Medial Axis Transform을 이용하여 돼지 머리를 제거하려면 많은 계산량이 필요하지만, 본 연구에서는 매우 빠른 경계박스 기반 딥러닝 기법(YOLO)과 빠른 후처리 기법을 적용하여 일반적인 영상처리 기법에 비하여 약 70배 빠른 처리속도를 제공할 수 있음을 확인하였다.

Table 1은 돼지의 자세 결정, Table 2는 돼지의 머리탐지 및 제거에 대하여 각각 기존 방법과 제안 방법의 정확도, 수행속도, 정확도와 수행속도를 곱한 값을 비교한 표이며, 수치가 높을수록 성능이 높음을 의미한다.

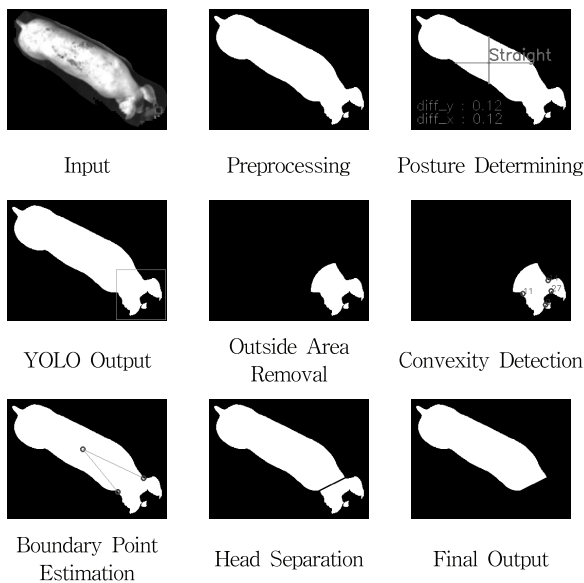


Fig. 9. Results of Posture Determining and Head Removal

Table 1. Performance Comparison of Posture Determining

Method	Accuracy (ACC) ↑	Processing Speed(fps) ↑	Accuracy × Processing Speed ↑
Deep Learning[14]	0.89	12.99	11.56
Proposed	0.98	250.00	245.00

Table 2. Performance Comparison of Head Removal

Method	Accuracy (ACC) ↑	Processing Speed(fps) ↑	Accuracy × Processing Speed ↑
Image Processing [2]	0.94	0.72	0.68
Proposed	0.96	48.97	47.01

## 5. 결론 및 향후 연구

양돈 업계에서 돼지의 무게는 다양한 변수를 결정하는 중요한 요소 중 하나이며, 이에 따라 돼지의 무게 측정은 필수적이다. 그러나 돼지 수 대비 관리인 부족으로 인하여 관리인이 직접 무게를 측정하는 대신 카메라로부터 획득한 영상 데이터로부터 픽셀 수를 이용하여 돼지의 무게를 추정하는 방법이 제안되었다. 픽셀 수를 이용하여 돼지의 무게를 추정하기 위해서는 픽셀 수 측정에 영향을 줄 수 있는 요인들을 제거할 필요가 있으며, 이러한 요인 중에는 돼지의 자세와 머리 위치가 있다. 즉, 돼지의 자세와 머리 위치로 인한 영향을 제거하기 위해서, 돼지의 자세를 결정한 후 돼지의 머리를 제거할 필요가 있다.

본 논문에서는 영상처리 기법을 이용하여 중심점으로부터 외곽선까지의 거리 비교를 통한 돼지의 자세 결정과 딥러닝 기반의 빠른 객체탐지 기법인 YOLO를 이용한 돼지의 머리 위치를 탐지하고 Convex-hull을 이용하여 탐지된 오목 점과 중심점과의 거리 정보를 이용한 돼지 머리 제거 방법을 제안하였다. 즉, 중심점으로부터 외곽선까지의 거리를 비교하여 빠르게 돼지의 자세를 결정한 후, YOLO를 이용하여 머리, 몸통과 엉덩이 위치를 학습시킨 후 머리 위치를 획득하여 불필요한 영역을 제거하였다. 이후 경량화 Convex-hull을 이용하여 머리 영역의 오목 점을 탐지하였으며, 이를 이용하여 머리와 몸통의 경계를 찾은 후 머리를 제거하였다.

실험 결과, 영상 데이터 1장당 약 0.98의 정확도와 250.00fps의 수행시간으로 돼지의 바른 자세를 결정하였으며, 약 0.96의 정확도와 48.97fps의 빠른 수행속도로 돼지 머리를 탐지하고 제거할 수 있음을 확인하였다. 즉, 기존 영상처리 및 End-to-End 딥러닝 기법을 적용한 경우에 비하여, 정확도 감소 없이 수행시간을 수십 배 개선함으로써 실시간 돼지 모니터링이 가능함을 확인하였다. 향후 연구로 돈사 내의 돼지들을 추적하여 각 돼지의 지속적인 무게추정이 가능하도록 확장할 예정이며, 픽셀 수 측정 시 영향을 줄 수 있는 다른 요인들을 검토 및 최소화하여 균일한 측정값을 획득할 수 있도록 할 예정이다.

## References

[1] Korea Rural Economic Institute [Internet], <https://www.krei.re.kr>, accessed July 2019.  
 [2] K. Jun, S. Kim, and H. Ji, "Estimating Pig Weights from Images Without Constraint on Posture and Illumination,"

*Computers and Electronics in Agriculture*, Vol.153, pp.169-176, 2018.  
 [3] N. Brandl and E. Jørgensen, "Determination of Live Weight of Pigs from Dimensions Measured Using Image Analysis," *Computers and Electronics in Agriculture*, Vol.15, No.1, pp.57-72, 1996.  
 [4] M. Kashiha, C. Bahr, S. Ott, C. Moons, T. Niewold, F. Ödberg, and D. Berckmans, "Weight Estimation of Pigs Using Top-view Image Processing," in *Proceedings International Conference Image Analysis and Recognition*, pp.496-503, 2014.  
 [5] M. Kashiha, C. Bahr, S. Ott, C. Moons, T. Niewold, F. Ödberg, and D. Berckmans, "Automatic Weight Estimation of Individual Pigs Using Image Analysis," *Computers and Electronics in Agriculture*, Vol.107, pp.38-44, 2014.  
 [6] H. Guo, X. Ma, Q. Ma, K. Wang, W. Su, and D. H. Zhu, "LSSA-CAU: An Interactive 3d Point Clouds Analysis Software for Body Measurement of Livestock with Similar Forms of Cows or Pigs," *Computers and Electronics in Agriculture*, Vol.138, pp.60-68, 2017.  
 [7] C. Shi, G. Teng, and Z. Li, "An Approach of Pig Weight Estimation Using Binocular Stereo System Based on LabVIEW," *Computers and Electronics in Agriculture*, Vol.129, pp.37-43, 2016.  
 [8] A. Pezzuolo, M. Guarinob, L. Sartoria, L. González, and F. Marinello, "On-barn Pig Weight Estimation Based on Body Measurements by a Kinect v1 Depth Camera," *Computers and Electronics in Agriculture*, Vol.148, pp.29-36, 2018.  
 [9] M. Lu, T. Norton, A. Youssef, N. Radojkovic, A. Fernández, and D. Berckmans, "Extracting Body Surface Dimensions from Top-view Images of Pigs," *International Journal of Agricultural and Biological Engineering*, Vol.11, No.5, pp.182-191, 2018.  
 [10] W. Choi, H. Ahn, H. Lee, Y. Chung, and D. Park, "Detection of Pig's Posture for Pig's Head Removal," in *Proceedings of the KIPS Spring Conference*, Vol.26, No.1, pp.625-628, 2018.  
 [11] H. Ahn, W. Choi, H. Lee, Y. Chung, and D. Park, "Image Processing for Pig's Head Removal," in *Proceedings of the KIPS Spring Conference*, Vol.26, No.1, pp.621-624, 2019.  
 [12] H. Blum, "A Transformation for Extracting New Descriptors of Shape," *Models for the Perception of Speech and Visual Form*, Vol.19, No.5, pp.362-380, 1967.  
 [13] Open Source Computer Vision: 'OpenCV'. Available at <http://opencv.org>, accessed December 2018.  
 [14] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields," *arXiv preprint arXiv:1812.08008*, 2018.  
 [15] Y. Chen, C. Shen, X. S. Wei, L. Liu, and J. Yang, "Adversarial PoseNet: A Structure-aware Convolutional Network for Human Pose Estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp.1212-1221, 2017.  
 [16] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask

R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp.2961-2969, 2017.

[17] J. Lee, L. Jin, D. Park, and Y. Chung, “Automatic Recognition of Aggressive Pig Behaviors Using Kinect Depth Sensor,” *Sensors*, Vol.16, No.5, pp.631, 2016.

[18] J. Kim, Y. Chung, Y. Choi, J. Sa, H. Kim, Y. Chung, D. Park, and H. Kim, “Depth-based Detection of Standing-pigs in Moving Noise Environments,” *Sensors*, Vol.17, No.12, pp.2757, 2017.

[19] M. Ju, Y. Choi, J. Seo, J. Sa, S. Lee, Y. Chung, and D. Park, “A Kinect-based Segmentation of Touching-pigs for Real-time Monitoring,” *Sensors*, Vol.18, No.6, pp.1746, 2018.

[20] J. Sa, Y. Choi, H. Lee, Y. Chung, D. and Park, J. Cho, “Fast Pig Detection with a Top-view Camera Under Various Illumination Conditions,” *Symmetry*, Vol.11, No.2, pp.266, 2019.

[21] M. Dillencourt, H. Samet, and M. Tamminen, “A General Approach to Connected-component Labeling for Arbitrary Image Representations,” *Journal of the ACM*, Vol.39, No.8, pp.253-280, 1992.

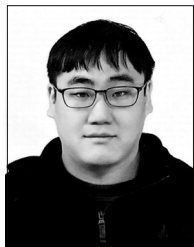
[22] J. Redmon, and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7263-7271, 2017.

[23] D. McCallum, and D. Avis, “A Linear Algorithm for Finding the Convex Hull of a Simple Polygon,” *Information Processing Letters*, Vol.9, No.5, pp.201-206, 1979.

[24] N. Otsu, “A Threshold Selection Method from Gray-level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.9, No.1, pp.62-66, 1979.

[25] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.679-698, 1986.

[26] J. Zunic, and P. Rosin, “A New Convexity Measure for Polygons,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.26, No.7, pp.923-934, 2004.



**안 한 세**

<https://orcid.org/0000-0003-3969-2879>  
 e-mail : hansahn@korea.ac.kr  
 2019년 고려대학교 컴퓨터정보학과(학사)  
 2019년~현 재 고려대학교  
 컴퓨터정보학과 석사과정  
 관심분야 : 영상처리, 병렬처리, 딥러닝



**최 원 석**

<https://orcid.org/0000-0002-2438-8834>  
 e-mail : carews315@korea.ac.kr  
 2018년~현 재 고려대학교 컴퓨터융합  
 소프트웨어학과 학사과정  
 관심분야 : 딥러닝, 최적화, 운영체제, H/W



**박 선 화**

<https://orcid.org/0000-0002-9130-9331>  
 e-mail : barksum@korea.ac.kr  
 2019년~현 재 고려대학교 컴퓨터융합  
 소프트웨어학과 학사과정  
 관심분야 : 머신러닝, 영상인식



**정 용 화**

<https://orcid.org/0000-0001-6539-167X>  
 e-mail : ychungy@korea.ac.kr  
 1984년 한양대학교 전자통신공학과(학사)  
 1986년 한양대학교 전자통신공학과(석사)  
 1997년 U. of Southern California(박사)  
 1986년~2003년 한국전자통신연구원  
 생체인식기술연구팀(팀장)  
 2003년~현 재 고려대학교 컴퓨터융합소프트웨어학과 교수  
 관심분야 : 병렬처리, 영상처리, 융합 IT



**박 대 희**

<https://orcid.org/0000-0003-4726-4508>  
 e-mail : dhpark@korea.ac.kr  
 1982년 고려대학교 수학과(학사)  
 1984년 고려대학교 수학과(석사)  
 1989년 플로리다 주립대학 전산학과(석사)  
 1992년 플로리다 주립대학 전산학과(박사)  
 1993년~현 재 고려대학교 컴퓨터융합소프트웨어학과 교수  
 관심분야 : 빅데이터, 데이터마이닝, 인공지능, 융합 IT