

Article

Fast Pig Detection with a Top-View Camera under Various Illumination Conditions

Jaewon Sa ¹, Yunchang Choi ¹, Hanhaesol Lee ¹, Yongwha Chung ^{1,*}, Daihee Park ¹ and Jinho Cho ²

¹ Department of Computer Convergence Software, Korea University, Sejong 30019, Korea; sjwon92@korea.ac.kr (J.S.); ycc4477@korea.ac.kr (Y.C.); maxsoribada@korea.ac.kr (H.L.); dhpark@korea.ac.kr (D.P.)

² Division of Food and Animal Science, Chungbuk National University, Cheongju 28644, Korea; jinhcho@cbnu.ac.kr

* Correspondence: ychungy@korea.ac.kr; Tel.: +82-44-860-1343

Received: 25 January 2019; Accepted: 18 February 2019; Published: 20 February 2019



Abstract: The fast detection of pigs is a crucial aspect for a surveillance environment intended for the ultimate purpose of the 24 h tracking of individual pigs. Particularly, in a realistic pig farm environment, one should consider various illumination conditions such as sunlight, but such consideration has not been reported yet. We propose a fast method to detect pigs under various illumination conditions by exploiting the complementary information from depth and infrared images. By applying spatiotemporal interpolation, we first remove the noises caused by sunlight. Then, we carefully analyze the characteristics of both the depth and infrared information and detect pigs using only simple image processing techniques. Rather than exploiting highly time-consuming techniques, such as frequency-, optimization-, or deep learning-based detections, our image processing-based method can guarantee a fast execution time for the final goal, i.e., intelligent pig monitoring applications. In the experimental results, pigs could be detected effectively through the proposed method for both accuracy (i.e., 0.79) and execution time (i.e., 8.71 ms), even with various illumination conditions.

Keywords: agriculture IT; computer vision; pig detection; depth information; infrared information

1. Introduction

Caring for group-housed pigs is an important issue that can be resolved by detecting or managing problems early with regards to their health and welfare [1–6]. Especially, it is required to minimize the potential damage for individual pigs from infectious diseases or other health problem. Because of the small number of farm workers, however, it is very challenging to care for individual pigs in a large pig farm.

Recently, several researches have been reported using surveillance techniques for an automatic pig monitoring system [7–42]. In this study, we focus on pig monitoring systems with a top-view camera under various illumination conditions in a realistic pig farm environment. The illumination problem has been considered by either applying image processing techniques that are time-consuming or by using thermal/depth cameras known to be less sensitive to the illumination problem. Indeed, we reported results previously for pig detection with Kinect-based depth information [38,39]. The depth information obtained from low-cost sensors, however, are susceptible to sunlight, and a fast solution to the illumination problem caused by sunlight has not yet been reported.

In this study, we propose not only a low cost but also a fast method to detect pigs through a top-view camera under various illumination conditions. First, we exploit the infrared and depth

information which is concurrently obtained from a low-cost camera, such as Intel RealSense [43]. The accuracy of the depth information measured from the RealSense camera is degraded significantly when covering a large area (i.e., pig room). Furthermore, sunlight through a window at daytime generates many noises on both depth and infrared information. Thus, we integrate both information complementarily to resolve the low-pixel accuracy and illumination noises such as sunlight. Second, we apply simple but effective image processing techniques only for satisfying the real-time execution for pig detection. By decreasing the computational workload of the detection task through the simple image processing, it allows to complete intermediate-level vision tasks, such as pig tracking, and high-level vision tasks, such as behavior analysis for pigs.

The rest of the paper is organized as follows: Section 2 describes previous pig detection methods. Section 3 explains the proposed method to detect pigs under various illumination conditions. The experimental results for pig detection are presented in Section 4, and Section 5 finally concludes the proposed method.

2. Background

The contribution of this study is to achieve our ultimate goal of an automatic analysis for pig behavior during 24 h by individually recognizing each pig through pig detection. The previous researches performed segmenting touching pigs and tracking individual pigs [41,42], but the most important task for the ultimate goal of a 24 h pig monitoring is accurate pig detection. For example, Figure 1 shows various illumination conditions in a realistic pig farm environment. With an infrared camera, the gray values of pigs located at four corners are generally darker than those of pigs located at center locations (see Figure 1a). In addition, the accuracy of the depth information obtained from a low-cost depth camera decreases quadratically as the distance increases [44]. Thus, we can apparently confirm the differences between infrared and depth information images. Sunlight through a window at daytime makes it especially difficult to separate pigs from the neighboring wall and floor (see Figure 1b) with both infrared and depth information. Clearly, the critical problem in consistently separating and tracking pigs for the automatic behavior analysis is to precisely detect pigs under various illumination conditions.

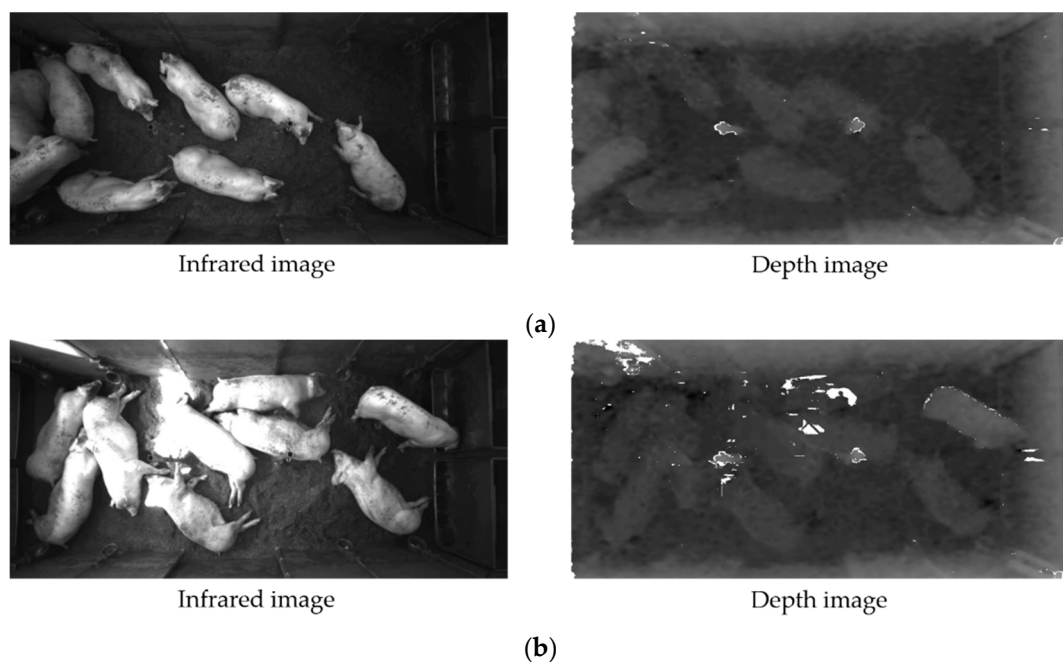


Figure 1. Various illumination conditions: (a) At 7 a.m. (for the purpose of explanation, we denote this kind of image as a low-contrast image) and (b) at 9 a.m. (for the purpose of explanation, we denote this kind of image as a sunlight image).

In order to enhance the low-contrast image as shown in Figure 1a or the sunlight image as shown in Figure 1b, we adopted contrast limited adaptive histogram equalization (CLAHE) [45], which is one of the most widely used techniques, to enhance the low contrast, as in bio/medical applications (e.g., CT/MRI imaging). Note that, histogram equalization (HE) [46] is also one of the most employed techniques for improving image contrast, but it may cause a problem that foreground cannot be detected because of the excessive change in brightness. Then, we adopted the Otsu algorithm [47] to detect objects from the gray images based on thresholding.

Figure 2 shows the results of Otsu after CLAHE. From the infrared images, it is difficult to detect the dark pig from the low-contrast image (see the red box shown in Figure 2a) or the possible boundary lines between the pig and the neighboring wall and floor from the sunlight image (see the red box shown in Figure 2b). The illumination problems with infrared images may be solved by using depth images. From depth images, however, it is difficult to detect the pig completely, owing to the inaccurate pixel values of the depth images (see the green area shown in Figure 2, and we call this problem the “missing pig-pixel problem” for the purpose of explanation). If we can exploit this complementary information from infrared and depth images, we can detect pigs more accurately under various illumination conditions.

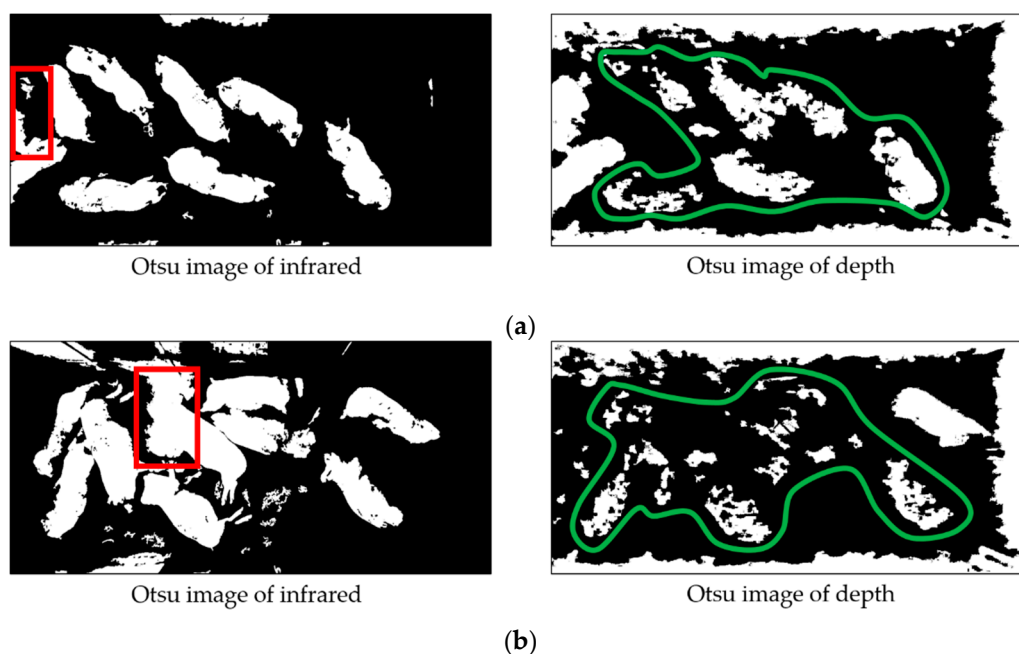


Figure 2. The difficulties of pig detection under various illumination conditions: (a) the results of Otsu [47] after contrast limited adaptive histogram equalization (CLAHE) [45] with a low-contrast image and (b) the results of Otsu [47] after CLAHE [45] with a sunlight image.

We summarize some of the previous approaches used for pig monitoring as shown in Table 1. Even if online monitoring applications are required to satisfy the real-time requirements, the processing speed was not described or the real-time requirements were not satisfied in many previous studies. Furthermore, some of the methods considered the illumination problem by applying time-consuming image processing techniques (i.e., “management of various illumination = Yes” shown in Table 1), whereas some others did not (i.e., “management of various illumination = No” shown in Table 1). Also, some of the methods tried to avoid the illumination problem by using thermal/depth cameras that were known to be less sensitive to the illumination problem. However, none of the previous methods have reported the results of pig detection with sunlight images (i.e., “management of sunlight = No” shown in Table 1). For example, we try to extend the previous research of detecting standing pigs [38,39] in order to additionally detect lying pigs with sunlight images, which is very difficult to solve with only depth information. In addition to pig detection, some studies for detecting objects have been

reported by using data modalities (i.e., multi-sensor fusion) through various information [48–52]. For example, References [48] and [49] proposed an object detection method by using both color and infrared information. In Reference [50], the fusion information between grayscale and thermal information was employed for foreground detection. As another example of data modality, Reference [52] proposed a background subtraction method for detecting moving objects by using color and depth information.

Table 1. Some of the pig detection results (published during 2009–2018).

Data Type	Data Size	Pig Detection Algorithm	Management of Various Illumination (Sunlight)	No. of Pigs in a Pen	Execution Time (seconds)	Reference
Gray/Color	Not Specified	Thresholding	No (No)	Not Specified	Not Specified	[7]
	720 × 540	CMA-ES	Yes (No)	12	0.220	[8]
	768 × 576	Wavelet	Yes (No)	Not Specified	1.000	[9]
	768 × 576	GMM	Yes (No)	Not Specified	0.500	[10]
	150 × 113	Texture	Yes (No)	Not Specified	0.250	[11]
	640 × 480	Learning	Yes (No)	9	Not Specified	[12]
	720 × 576	Thresholding (Otsu)	No (No)	10	Not Specified	[13]
	1280 × 720	Thresholding	No (No)	7–13	Not Specified	[14]
	Not Specified	GMM	Yes (No)	3	Not Specified	[15]
	352 × 288	ANN	No (No)	Not Specified	0.236	[16]
	640 × 480	Thresholding (Otsu)	No (No)	22–23	Not Specified	[17]
	640 × 480	Thresholding (Otsu)	No (No)	22	Not Specified	[18]
	Not Specified	Thresholding (Otsu)	No (No)	17–20	Not Specified	[19]
	574 × 567	Color	No (No)	9	Not Specified	[20]
	256 × 256	GMM/Thresholding	Yes (No)	Not Specified	Not Specified	[21]
	1760 × 1840	Global + Local Thresholding	Yes (No)	Not Specified	Not Specified	[22]
	1280 × 720	Global + Local Thresholding	Yes (No)	23	0.971	[23]
	Not Specified	Thresholding (Otsu)	No (No)	2–12	Not Specified	[24]
	320 × 240	Thresholding (Otsu)	No (No)	Not Specified	Not Specified	[25]
	512 × 424	Thresholding (Otsu)	Yes (No)	Not Specified	Not Specified	[26]
	1440 × 1440	Thresholding	Yes (No)	Not Specified	1.606	[27]
960 × 540	Deep Learning	No (No)	1	Not Specified	[28]	
2560 × 1440	Deep Learning	No (No)	4	Not Specified	[29]	
Depth	Not Specified	Depth Thresholding	No (No)	1	Not Specified	[30]
	640 × 480	Depth Thresholding	No (No)	Not Specified	Not Specified	[31]
	512 × 424	Depth Thresholding	No (No)	1	Not Specified	[32]
	512 × 424	Thresholding (Otsu)	No (No)	Not Specified	Not Specified	[33]
	512 × 424	Depth Thresholding	No (No)	1	Not Specified	[34]
	1294 × 964	Depth Thresholding	No (No)	1	Not Specified	[35]
	512 × 424	GMM	No (No)	19	0.142	[36]
	512 × 424	Deep Learning	Yes (No)	1	0.050	[37]
	512 × 424	Depth Thresholding	No (No)	22	0.056	[38]
	512 × 424	Depth Thresholding	No (No)	13	0.002	[39]
Gray + Depth	1280 × 720	Infrared + Depth Fusion	Yes (Yes)	9	0.008	Proposed Method

To the best of our knowledge, this is the first report on detecting pigs in real time by exploiting the complementary information (i.e., without any time-consuming techniques, such as frequency-, optimization-, or deep learning-based detections) with sunlight images obtained from a low-cost camera. That is, we propose a fast pig detection method with a reasonable accuracy with the ultimate goal of achieving a “complete” real-time vision application from low-level vision tasks to intermediate- and high-level vision tasks by carefully balancing the tradeoffs between the computational workload and detection accuracy as well as exploiting both depth and infrared information.

3. Proposed Method

As in Reference [39], detecting pigs in a pen can be achieved by analyzing the depth information between the background and foreground (e.g., the floor, wall, and pigs) because the depth information

is less sensitive to various illuminations. However, it is challenging to precisely detect the pigs by analyzing depth information because the depth information obtained from the low-cost camera is measured inaccurately. That is, pigs such as lying pigs cannot be detected according to a certain threshold that is obtained from the inaccurate depth information. Meanwhile, infrared information has the advantage of accurate pixel values, so that the pigs can be detected from the background by using simple image processing techniques. If some of pigs are located at the four corners of the pen, however, the pigs cannot be detected accurately because the gray values at the corners are darker than those of the center. Furthermore, if sunlight appears in the pig pen during daytime, many noises seriously affect both the depth and infrared information due to the sunlight and, thus, make it difficult to detect the pigs accurately. Thus, it is required to exploit the complementary information from both the depth and infrared images under the low-contrast and sunlight environment.

In this study, we propose a fast pig detection method, which is denoted as 'FastPigDetect' under various illumination conditions by using the advantages of both depth (i.e., less sensitive to illumination) and infrared (i.e., more accurate pixel values) information. First, the region of interest (ROI) is set to exclude unnecessary regions, such as a feeder or another pig pen. Then, we remove not only noises generated by the sunlight at daytime but also other noises according to the environment of the pen by using spatiotemporal interpolation on both depth and infrared information. In the next step, the neighboring background (i.e., the floor and wall) is segmented from the pigs by analyzing the depth information, and the contrast of the infrared information is improved with a contrast enhancement technique to "roughly" detect pigs. Finally, the pigs in the pen can be "precisely" detected by integrating both the depth and infrared information with simple image processing techniques. With the advantages of the two information, the pigs can be detected effectively in the low-contrast as well as the sunlight environment. Figure 3 shows the overall procedure for the pig detection under various illumination conditions.

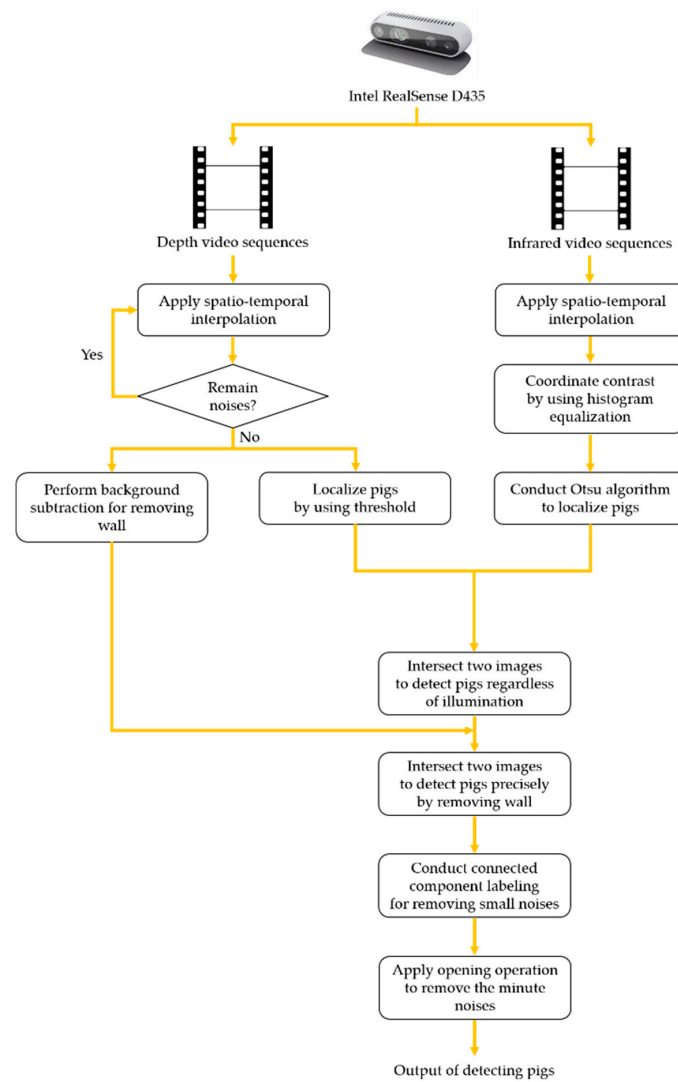


Figure 3. The overall procedure of the proposed method.

We define some terminologies described in the proposed method for increased readability. Table 2 describes the terminologies for each procedure of the pig detection method.

Table 2. Definitions of the key terminologies of the proposed method.

Category	Definition	Description
Depth	D_{input}	Depth input image
	$D_{interpolate}$	Depth background image through modeling during 24 h videos
	$D_{background}$	Depth interpolated image through spatiotemporal interpolation
	$D_{localize}$	Depth image where pigs are localized through threshold
	$D'I_{localize}$	Depth image where pigs are localized through background subtraction and Otsu
Infrared	I_{input}	Infrared input image
	$I_{interpolate}$	Infrared interpolated image with spatiotemporal interpolation
	$I_{contrast}$	Infrared image where the contrast is coordinated by histogram equalization
	$I_{localize}$	Infrared image where pigs are localized by Otsu algorithm
Depth + Infrared	DI_1	Intersection image between $D_{localize}$ and $I_{localize}$
	DI_2	Intersection image between DI_1 and $D'I_{localize}$

3.1. Removing Noises and Localizing Pigs

3.1.1. Procedure with Depth Information

In Section 3.1.1, we localize the pigs in the pig pen by removing noises from the depth information. From D_{input} , we first set the ROI to exclude the unnecessary regions (i.e., a feeder or another pen). Then, we apply 4×4 window spatiotemporal interpolation [39] as a preprocessing step for removing noises (i.e., undefined pixels) that may occur, such as those caused by sunlight at daytime. Note that because the noises generated from intense sunlight are similar to the large moving noises described in Reference [39], the spatiotemporal interpolation technique is iteratively conducted until the noises that are removed. For understanding the locations of each pig, the pixel frequencies from both foreground (i.e., the pigs) and background (i.e., the floor and wall) are calculated through histogram analysis using $D_{interpolate}$. We note that the background area is larger than that of each pig in the pen. Accordingly, the most frequent pixel can be selected as the background pixel used to segment the depth image into the background and the foreground, i.e., the pixel used to separate the foreground and background from each other. By setting the most frequent pixel to the threshold for background segmentation, the roughly localized pigs in $D_{localize}$ can be obtained by using the threshold. However, the depth values of the wall may be the same as those of the pigs in accordance with their size, so that the floor can be relatively removed, but the wall cannot be removed using the threshold.

In order to resolve the problem, we conduct the background modeling for producing $D_{background}$ and apply the frame difference between $D_{background}$ and D_{input} for removing the wall. Before modeling $D_{background}$, it is necessary to realize the characteristic of depth information. The depth information is likely to be measured inaccurately according to the distance between the sensor and the pigs/background. For example, even if the size of the pig located at the corner in the pen is the same with the size of the pig located at the center, the depth values of each pig are subtly different because the pig in the corner is farther away from the sensor than the pig in the center. In case of the background, depth values at any locations of the background may be also obtained differently according to the distance from the camera. Thus, the background modeling is required to calibrate the depth values at any location, which are then used to conduct background subtraction for calibrating the depth values of each pig and the background.

For modeling the background of the pig pen, we exploit all depth information videos recorded during a 24 h period. First, the floor and other parts (i.e., the wall and pigs) for every frame are respectively divided by using the threshold for background segmentation that is selected by histogram analysis. Because the depth values of the wall among other parts may be the same as with those of the pigs, the pigs and the wall are considered to be in the same category. Here, we define the floor and the other parts as the floor background and the other background, respectively. In the next step, the floor background and the other background are independently updated with the depth values of the floor and the other parts during the 24 h videos. After updating each background, the depth values of the other background are overlapped to the not updated regions of the floor background because the depth values of the floor are only updated from D_{input} through the threshold.

Every $D_{interpolate}$ is applied with frame difference using the modeled $D_{background}$ and by applying histogram equalization and Otsu algorithm [47] on the image, a $D'_{localize}$, where the pigs are localized with removing the wall, can be obtained. Applying the two localized images, $D_{localize}$ and $D'_{localize}$, to the infrared images allows for the robust detection of the pigs in the sunlight and low-contrast conditions. Figure 4 shows the results of localization for the pigs through the depth information in low-contrast and sunlight conditions, respectively. The missing pig-pixel problem could be solved effectively (compare the green areas shown in Figures 2 and 4).

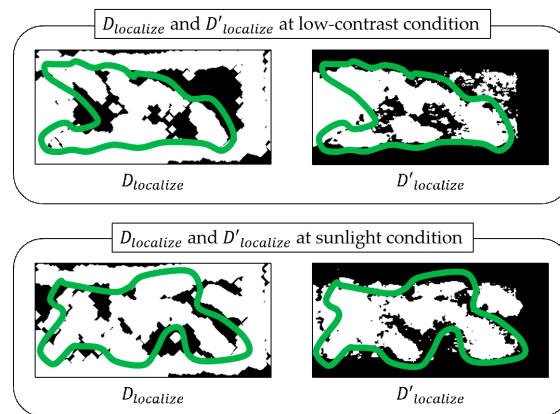


Figure 4. The localization of the pigs with a low contrast (at 7 a.m.) and a sunlight image (at 9 a.m.). $D_{localize}$ and $D'_{localize}$ are generated through the threshold from a histogram analysis about the depth values and background subtraction using modeled $D_{background}$, respectively.

3.1.2. Procedure with Infrared Information

In fact, the pigs in the pen can be detected using characteristics of the infrared information I_{input} , such as accurate pixel values. However, there is a problem in that pigs may not be detected in various illumination conditions such as sunlight and low contrast in I_{input} . In other words, because the infrared image is affected by various illuminations, the localized images obtained from the depth information, $D_{localize}$ and $D'_{localize}$, should be exploited for accurately detecting the pigs.

In the same way as the depth images, the ROI of I_{input} is set in order to exclude the unnecessary region in the pen, and then the 4×4 spatiotemporal interpolation technique is performed for removing noises such as sunlight. Note that the spatiotemporal interpolation technique is performed only once, as the pixel values in I_{input} are not correctly interpolated, owing to the characteristic sensitivity to illumination conditions. Then, the histogram equalization (HE) [46] is performed to resolve the low contrast of $I_{interpolate}$, which makes the contrast in $I_{interpolate}$ consistent. Through the procedure of HE, the Otsu algorithm is applied to roughly localize the pigs in $I_{contrast}$. Figure 5 shows the results of the localization of the pigs in $I_{localize}$ gained from $I_{contrast}$ at low-contrast and sunlight conditions. However, the pigs in $I_{localize}$ cannot be accurately detected because the contrast of all pixels in the floor, wall, and sunlight are also coordinated consistently by applying HE. That is, even though all the pigs in the pen can be confirmed, the noises are not totally removed. These noises can be removed by exploiting the complementary information from the infrared images (i.e., $I_{localize}$) and depth images (i.e., $D_{localize}$ and $D'_{localize}$) simultaneously.

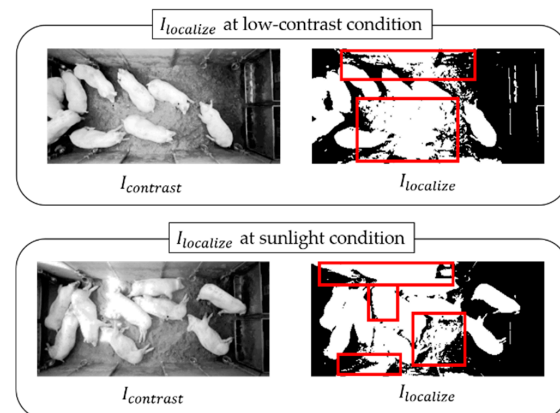


Figure 5. The localization of the pigs in the infrared information at low-contrast and sunlight conditions. The bold box indicates the noises detected by histogram equalization (HE) in $I_{localize}$. Although the pigs can be identified at low-contrast and sunlight conditions, some pigs are unidentified by the detected background because all of the pixels in $I_{interpolate}$ are consistently coordinated by HE.

3.2. Detecting Pigs Using both Depth and Infrared Information

In order to detect only the pigs, DI_1 is produced by conducting an intersection operation between $D_{localize}$ and $I_{localize}$ where HE and the Otsu algorithm are applied. Figure 6 shows the result after an intersection operation to detect the pigs by removing the noises generated from the floor, wall, and sunlight.

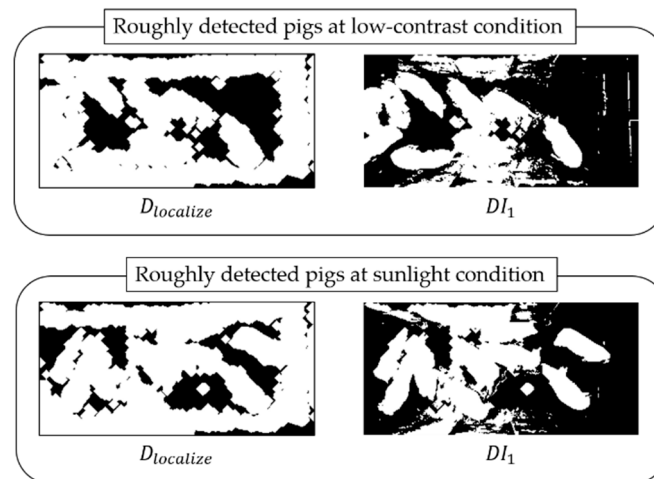


Figure 6. The result of the intersection operation between $D_{localize}$ and $I_{localize}$ to detect the pigs.

Nevertheless, there is a problem in that the wall and floor are still detected in DI_1 : first, the wall is not removed in not only $I_{localize}$ with HE and Otsu algorithm but also in $D_{localize}$; second, the center of the floor is also detected, largely because of all of the coordinated contrast pixels caused by HE. For only detecting pigs from the wall and floor, $DI_{localize}$, i.e., the frame difference image between D_{input} and $D_{background}$, is used. Because the wall in $DI_{localize}$ is mostly removed through background subtraction and the pigs are roughly localized in the image, it is able to detect the pigs by performing an intersection between DI_1 and $DI_{localize}$, where most of the wall and floor are removed in DI_2 . Given DI_2 , the post-processing using some image processing techniques is performed to accurately detect the pigs. In order to remove the noise remaining in DI_2 , an erosion operation is conducted to remove and minimize small noises that are adjacent to the objects or generated from the intersection operation in DI_2 . Then, all of the objects are labeled through connected component analysis (CCA), where each area of the objects is calculated, along with whether the objects should be removed or not according to their sizes. After removing the noises according their sizes, the pigs can be precisely detected by using a dilation operation to recover the shapes of the pigs. Figure 7 shows that the pigs are finally detected by applying the proposed method through D_{input} and I_{input} in various illumination conditions.

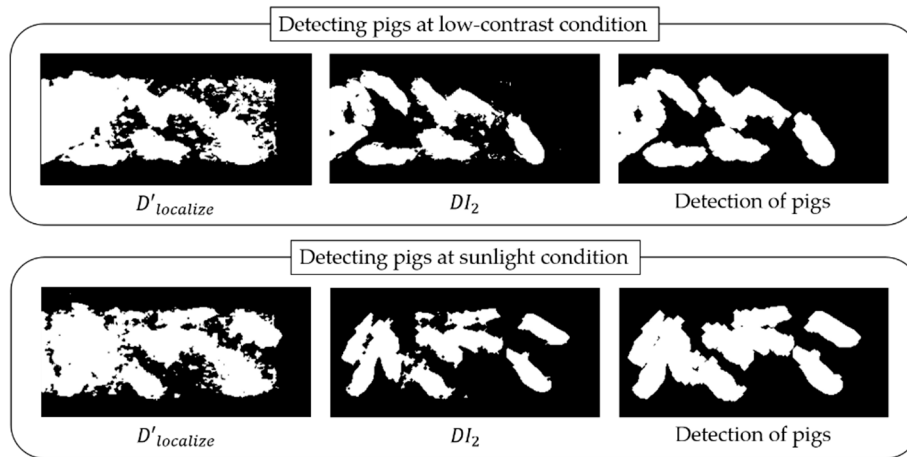


Figure 7. The detection result of all pigs by using depth information and infrared information.

At last, the proposed method is described in Algorithm 1 as follows.

Algorithm 1. Pig detection algorithm under various illumination conditions

Input: Depth and infrared images

Output: Detected pig image

Step 1: Removing noises and localizing pigs with depth and infrared information individually

Procedure with depth information:

Generate $D_{background}$ from modeling background during 24 h videos;

$D_{interpolate} = SpatTempIntp(D_{input})$;

$threshold = HistAnalysis(D_{interpolate})$;

for $y = 0$ to $height$:

for $x = 0$ to $width$:

if $D_{interpolate}(x, y) > threshold$:

$D_{localize}(x, y) = 255$;

else:

$D_{localize}(x, y) = 0$;

$DI_{localize} = BackgroundSubtract(D_{interpolate}, D_{background})$;

$DI_{localize} = Otsu(DI_{localize})$;

Procedure with infrared information:

$I_{interpolate} = SpatTempIntp(I_{input})$;

$I_{contrast} = HistEqualization(I_{interpolate})$;

$I_{localize} = Otsu(I_{contrast})$;

Step 2: Detecting pigs with depth and infrared information collectively

$DI_1 = Intersect(I_{localize}, DI_{localize})$;

$DI_2 = Intersect(DI_1, DI_{localize})$;

Erode DI_2 to remove and minimize noises;

Conduct CCA to the minute noises in DI_2 ;

Dilate DI_2 to recover shapes of the pigs;

4. Experimental Results

4.1. Experimental Setup and Resources for the Experiment

The following experimental setup was used to conduct our pig detection method: Intel Core i7-7700K 4.20 GHz (Intel, Santa Clara, CA, USA), NVIDIA GeForce GTX1080 Ti 11 GB VRAM (NVIDIA, Santa Clara, CA, USA), 32 GB RAM, Ubuntu 16.04.2 LTS (Canonical Ltd, London, UK), and OpenCV 3.4 [53] for image processing. We installed an Intel RealSense low-cost camera (D435 model, Intel,

Santa Clara, CA, USA) [43] on a ceiling at a height of 3.2 m in a 2.0 m \times 4.9 m pig pen located in Chungbuk National University, Korea.

In the pig pen, a total of nine pigs (Duroc \times Landrace \times Yorkshire) were raised, with an average initial body weight (BW) of 92.5 ± 5.9 kg. We simultaneously obtained infrared and depth videos from the installed camera, which had a resolution of $1,280 \times 720$ and 30 frames per second (FPS). Figure 8 displays the whole monitoring setup with the camera in the pig pen.

We used the depth and infrared images obtained from the camera during a 24 h period. Because it was extremely difficult to create the ground-truth image 24 h videos (i.e., 2,592,000 frames were obtained from 24 h videos of 30 FPS), our method for detecting the pigs was applied to three frames per ten minutes (i.e., total of 432 frames) selected from each video. Meanwhile, as explained in Section 2, various illumination conditions in the depth and infrared videos were confirmed, such as low contrast and sunlight. In particular, the illumination issues of low-contrast and sunlight conditions were evidently found when the pigs were located at the corners in the pen or when sunlight appeared at the specific time (08:00–10:00 a.m.). Thus, we detected the pigs while considering the issues for the illumination conditions.

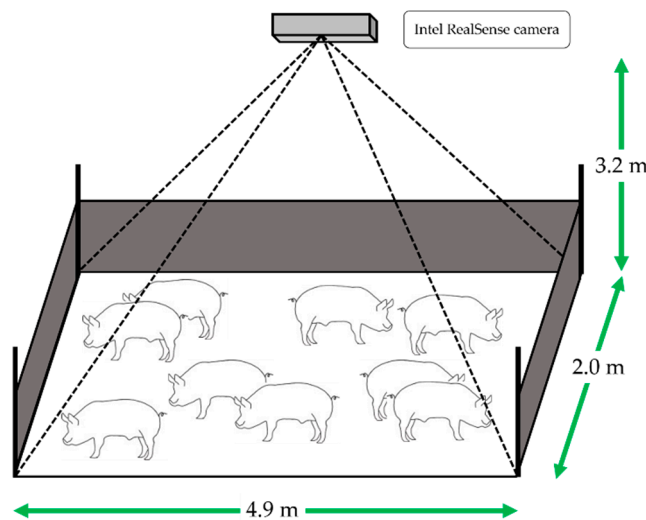


Figure 8. The experimental setup with a RealSense low-cost camera.

4.2. Detection of Pigs under Various Illumination Conditions

Initially, we modeled $D_{background}$ as an independent procedure for conducting the frame difference between $D_{background}$ and $D_{interpolate}$. To remove and minimize the noises caused by the illumination conditions in the depth and infrared information, a spatiotemporal interpolation technique was applied to the 1296 frames extracted from each video. Note that because the spatiotemporal interpolation technique was interpolated from three frames to one frame, 1296 frames were needed to detect the pigs in 432 frames. In the $D_{interpolate}$ and $I_{interpolate}$ derived from the interpolation technique, simple image processing techniques were conducted to each domain.

In the case of the procedure of depth information, a histogram analysis was performed to gain $D_{localize}$ from $D_{interpolate}$. Here, the frequency of the depth value corresponding to the background converged to 53, and the threshold for segmenting the background was defined as 53. $D_{localize}$ was then derived by binarizing $D_{interpolate}$, using the threshold defined through histogram analysis. In the second step, the frame difference between $D_{background}$ and $D_{interpolate}$ was carried out to derive $D_{localize}$, where the Otsu algorithm was applied to $D_{localize}$. Note that because the parameter for localizing the pigs may be changed continuously according to the inaccurate depth values, the Otsu algorithm should be used to automatically determine the parameter for every image. In the case of the procedure of infrared information, $I_{contrast}$ could be obtained by applying HE to localize the pigs with $I_{interpolate}$. Similar to the procedure of the depth information, the Otsu algorithm was used to define the parameter

for segmenting the background so that $I_{localize}$, where the pigs were localized, was obtained from $I_{contrast}$ with the Otsu algorithm. With the attributes from these procedures, DI_1 and DI_2 were obtained by intersecting among these localized images, where the noises resulted in the illumination conditions were removed. Finally, a morphology operation and CCA were conducted to DI_2 as the post-processing steps for refining the detected pigs. As the size of each noise calculated by CCA was less than 100, the noises were simply removed with the threshold defined as 100. After that, a dilation operation was conducted three times to sufficiently recover the shape of the pigs, and as a result, all of the pigs in the pen could be accurately detected. Figure 9 illustrates the detected pigs by using the proposed method from the 24 h recorded videos. In Figure 9, only one detection result per hour is displayed because of the large number of the frames in the 24 h videos.

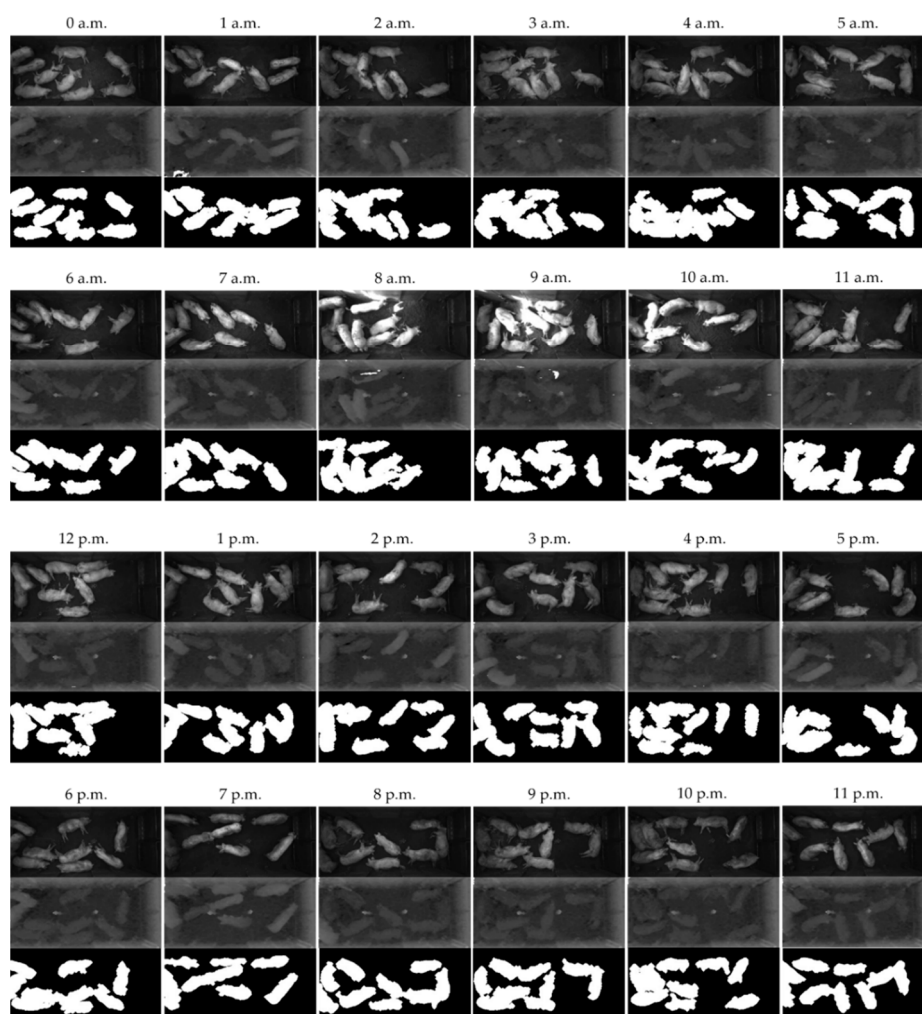


Figure 9. The results of pig detection under various illumination conditions.

4.3. Evaluation of Detection Performance

For evaluating the performance of detecting the pigs from the proposed method, we compared the detection result of the proposed method with those of state-of-the-art deep learning-based methods, including YOLO9000 [54] (i.e., a bounding box-based object detection method) and DeepLab [55] (i.e., a pixel-level semantic segmentation method). In particular, YOLO9000 was selected among many bounding box-based object detectors because it is known to be very fast and reasonably accurate (due to its “you only look once”). DeepLab was also selected among many pixel-level semantic segmentors because it is known to be fast and accurate (due to its “Atrous convolution”). Because YOLO9000 is a bounding box-based object detector and DeepLab is a pixel-level semantic segmentor,

YOLO9000 is expected to be faster but less accurate than DeepLab. Note that because the depth information was inaccurate as described in Section 2, we only used the infrared information for training and testing the data (i.e., detecting the pigs) with the deep learning-based methods. Before executing the deep learning-based methods, we realized that it was hard to generate the ground-truth images, and that the ground-truth was not enough to train for deep learning-based methods. Thus, we generated 2592 ground-truth data through data augmentation by flipping the input data vertically and horizontally. Note that the input image resolution could be increased to detect more objects [56]. However, the dataset for training and testing both YOLO and DeepLab was composed of the same resolution as the data which was used in the proposed method for fair comparison.

In the case of YOLO9000, we produced a model through the training data, which was composed of 2592 infrared frames. We defined the hyperparameters that were utilized in YOLO9000 for training as follows: 0.001 for learning rate, 0.0005 for decay, default anchor parameter, 0.9 for momentum, leaky ReLU as the activation function, and 10,000 for the epoch. In the case of DeepLab, we also produced a model through the training data, which was composed of the same dataset as YOLO. In addition, we defined the hyperparameters that were also utilized in DeepLab for training as follows: 0.006 for learning rate, 0.0005 for decay, 0.9 for momentum, ReLU as the activation function, and 30,000 for the epoch. In the training step of each method, a pretrained model through ResNet with COCO dataset was exploited. We then used 432 test frames, consisting of sunlight and low-contrast conditions in the pen, as well as normal conditions. From the test step from each method, YOLO9000 generated bounding boxes on the pigs and DeepLab conducted semantic segmentation between foreground (i.e., pigs) and background (i.e., floor and wall). However, both of the methods could not detect some pigs located at the corner or in the area of sunlight, as compared to the proposed method. Figure 10 shows the results of the detected pigs for each method in the various illumination conditions.

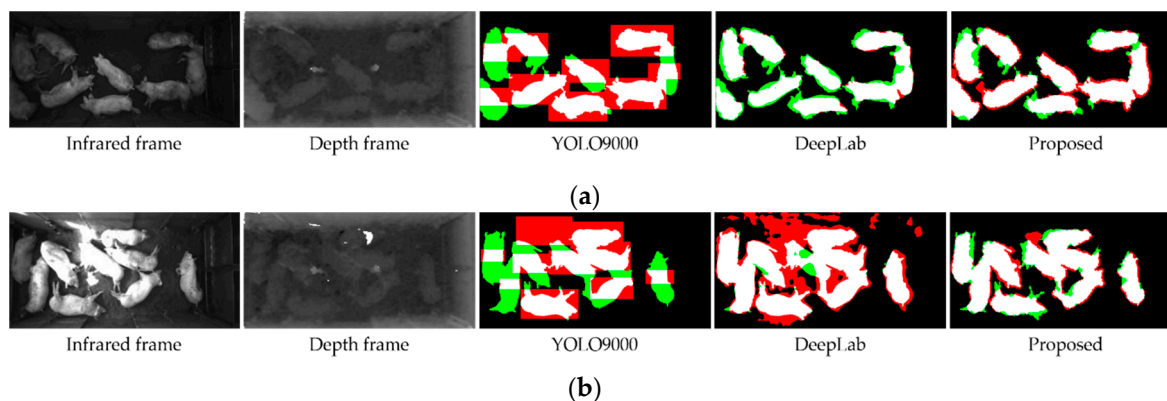


Figure 10. The results of each method for pig detection: (a) the results with a low-contrast image and (b) the results with a sunlight image.

In the experimental results for detecting the pigs through the proposed method and the deep learning-based methods, we calculated the pig detection accuracy for comparing the performance of each method. We calculated the precision, recall, and the detection accuracy (denoted as ACC) as the intersection-over-union [57] for each method using the following equations:

$$Precision = \frac{TP}{TP + FP'} \quad (1)$$

$$Recall = \frac{TP}{TP + FN'} \quad (2)$$

$$ACC = \frac{TP}{TP + FN' + FP'} \quad (3)$$

where true positive (*TP*) means a pixel on the pig predicted as the pig, false positive (*FP*) means a pixel on the background predicted as the pig, and false negative (*FN*) means a pixel on the pig predicted as the background, respectively. As shown in Figure 10, for example, we represented the false detected pixels for pigs (i.e., *FP* as false pig and *FN* as false background) as the red and green colors, respectively, from the results of each detection method. In the experimental results, the precision of each method was respectively measured as 0.79 (YOLO9000 method), 0.91 (DeepLab method), and 0.92 (proposed method). Also, the recall of each method was derived as 0.64 (YOLO9000 method), 0.88 (DeepLab method), and 0.86 (proposed method). Lastly, the detection accuracy (i.e., *ACC*) was measured as 0.54 (YOLO9000 method), 0.79 (DeepLab method), and 0.79 (proposed method), as shown in Table 3. By carefully fusing the depth and infrared information, the proposed method could also provide a higher accuracy than the deep learning-based methods.

Table 3. A comparison of the average performance.

Method	Precision	Accuracy Recall	ACC	Execution Time	Real-Time Accuracy ($ACC_{RealTime}$)
YOLO [54]	0.79	0.64	0.54	14.65 ms	0.80
DeepLab [55]	0.91	0.88	0.79	264.83 ms	Undefined
FastPigDetect (Proposed)	0.92	0.86	0.79	8.71 ms	0.95

In addition, the execution time for each method was measured in order to verify the real-time requirements on pig detection. As shown in Table 3, YOLO9000 could provide faster results than DeepLab. By applying simple but effective image processing techniques without any time-consuming techniques, the proposed method could provide much faster results than YOLO9000. Note that the deep learning-based methods have a huge number of weights to be computed and thus required tens or hundreds of milliseconds to detect the pigs from one image, even with a powerful GPU. On the contrary, the execution time of the proposed method was measured with a single CPU core. If we parallelize the simple pixel-level operations of the proposed method, then we can improve the execution speed of the proposed method further.

For real-time video stream applications such as 24 h pig monitoring or autonomous driving [58], we need to maximize the accuracy while satisfying the real-time constraint. Generally, there is a tradeoff between accuracy and the computational resources required. That is, a higher accuracy requires more computational resources, whereas less computational resources drive a lower accuracy. Thus, the tradeoff between accuracy (i.e., *ACC*) and processing speed (i.e., *FPS*) should be analyzed for the 24 h pig monitoring application. Similar cases have been analyzed by the video compression community to control the power consumption of an embedded computer and to maximize the compressed video quality [59,60]. For the purpose of explanation, we define “real-time accuracy” (denoted as $ACC_{RealTime}$) as follows.

To derive the collective (i.e., *ACC* vs *FPS*) performance of a method *X*, we first represent the performance of method *X* in the two-dimensional domain of *FPS* (i.e., *x* axis) and *ACC* (i.e., *y* axis) as shown in Figure 11a:

$$Performance = (X_{FPS}, X_{ACC}), \text{ where } 0 < X_{ACC} < 1. \quad (4)$$

Then, we assume two hypotheses: first, the upper limit of X_{ACC} with an unlimited computational resource (i.e., $X_{FPS} = 0$) is 1 (see the black point at (0, 1) shown in Figure 11a); second, each computational step of method *X* contributes to the accuracy equally. In addition, the real-time criterion for a video stream application such as 24 h pig monitoring is set to 30 FPS (see the dashed

line shown in Figure 11a). For the detection accuracy of method X at 30 FPS, we estimate the real-time accuracy of method X by using the two points $(0, 1)$ and (X_{FPS}, X_{ACC}) by using Equation (5) and (6):

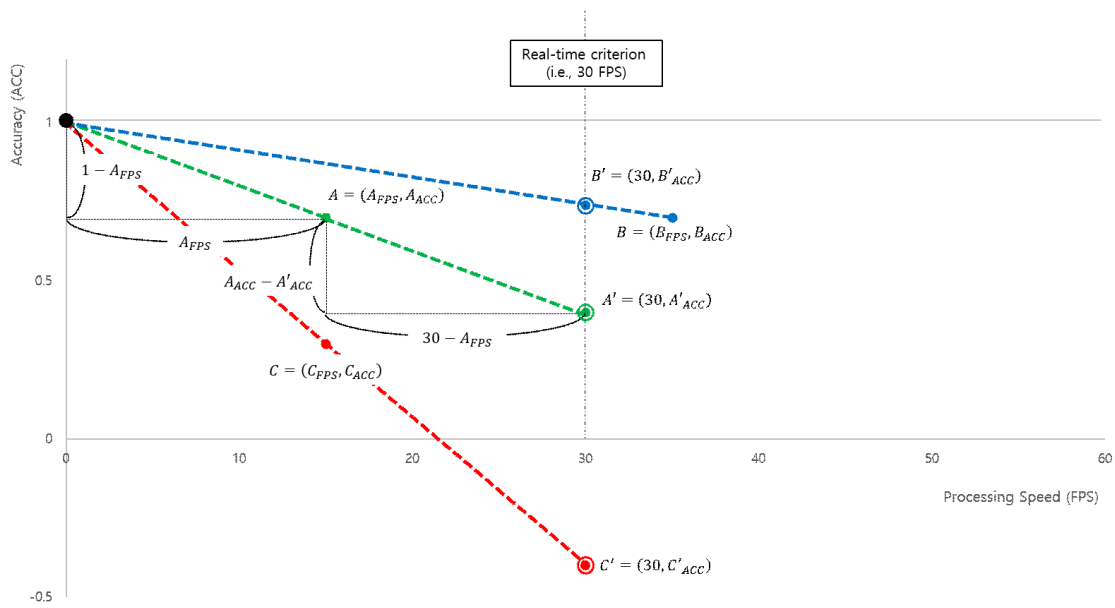
$$ACC_{RealTime} = \begin{cases} X'_{ACC} & \text{if } X'_{ACC} \geq 0, \\ \text{undefined} & \text{otherwise.} \end{cases} \quad (5)$$

$$\text{Where } X'_{ACC} = \frac{30}{X_{FPS}} \cdot X_{ACC} + \left(1 - \frac{30}{X_{FPS}}\right). \quad (6)$$

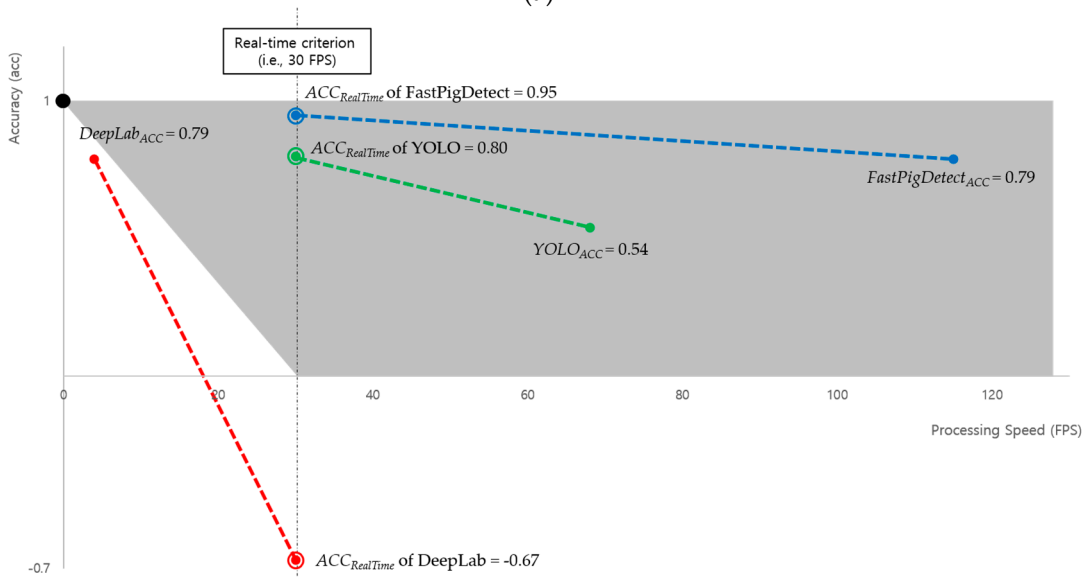
For example, we can represent the performance of three methods (i.e., A , B , and C) in the two-dimensional domain of FPS and ACC, as shown in Figure 11a, as $A_{FPS} < 30$ and $B_{FPS} > 30$, $A'_{ACC} < A_{ACC}$ and $B'_{ACC} > B_{ACC}$. However, the real-time accuracy of method C is undefined because $C'_{ACC} < 0$. It means that method C cannot satisfy the real-time requirement due to the relatively low accuracy in terms of its resource consumption. Figure 11b shows the possible area where $ACC_{RealTime}$ can be defined. That is, the real-time accuracies of the proposed and YOLO9000 methods could be defined sufficiently, whereas the DeepLab method could be defined marginally with the very low real-time accuracy on our experimental setup. Especially, the precise detection of the pigs using DeepLab was very difficult in low-contrast and sunlight environments, and the huge computational workload required for semantic segmentation was also burdened for real-time pig detection. On the contrary, the proposed method (i.e., FastPigDetect) could provide a reasonable accuracy with much less computational workload. As described in Section 1, it is necessary to establish a complete and automatic monitoring application in real-time for our final goal involving both intermediate- and high-level vision tasks. That is, detecting pigs should be performed as fast as possible by considering the further procedures of both intermediate- and high-level vision tasks. With less time-consuming techniques, it is able to establish a real-time monitoring application for pig involving both intermediate- and high-level vision tasks.

Although the FastPigDetect method could detect the pigs in real time by applying simple image processing through data modality between infrared and depth information, it is necessary to develop a parameter-optimized pig detection method. It means that the generalization for other modality images or other pig room data may be impossible with the current proposed method because the parameter of the proposed method by using both depth and infrared information were optimized for our experimental pig room only. In fact, the contribution of the proposed method is to detect the pigs in real time from various illumination conditions including intense sunlight in a pig room. It is important to develop a general pig detection algorithm whose parameters are determined automatically for commercial products. However, other parameters for the RoI setting or morphological operation (e.g., dilation/erosion) should be optimized according to the structure of a pig room (e.g., shape of floor/wall or size of a pig room) or the camera installation environment (e.g., installation height). Even though the generalization capability is out of scope of this study, the capability is required for commercial products, and it will be an interesting future work.

Furthermore, our proposed method exploited both infrared and depth information, but the deep learning-based methods only used infrared information. In the previous study [39], the YOLO model trained with depth information was used to detect only standing pigs in a pig room. When training the YOLO model with only depth information, however, the detection performance for standing and lying pigs through YOLO was not acceptable. Since detecting lying pigs was much more difficult than standing pig detection due to inaccurate depth values, for accomplishing our goal in this study, we conducted training and testing with YOLO by only using infrared information. As shown in Figure 1, the infrared information has more accurate pixel values than depth information. However, multimodal learning has several research issues. Thus, we will explore how the pig detection accuracy through the deep learning-based method is improved by using depth information as an additional channel with infrared information, as well as fine-tuning the deep learning architecture.



(a)



(b)

Figure 11. A comparison of each method by using various performance metrics: (a) the illustration of the real-time accuracy in the two-dimensional domain of processing speed (i.e., FPS) and accuracy (i.e., ACC) and (b) the comparison of the real-time accuracy between the proposed method and deep learning-based methods. The shadow area shows the range of ACC of which the $ACC_{RealTime}$ can be defined.

5. Conclusions

In a surveillance environment on a realistic pig farm, fast pig detection is important to efficiently manage the pigs for their health care. Nevertheless, there is a problem that pigs could not be accurately detected because of various illumination conditions in a realistic pig farm. With an infrared camera, for example, the gray values of pigs located at the four corners are generally darker than those of pigs located at center locations. In particular, sunlight through a window at daytime makes it difficult to separate pigs from the neighboring wall and floor.

In this study, we concentrated on detecting pigs in real time under various illumination conditions to analyze the behaviors of individual pigs with the final goal of the consistent monitoring during 24 h. In other words, we proposed a pig detection method at daytime and nighttime with less time-consuming techniques. As an initial step for preprocessing, a spatiotemporal interpolation technique was applied to remove the noise caused by sunlight. Then, we detected pigs by carefully fusing the depth and infrared information and applying image processing techniques. In particular, we applied simple but effective image processing techniques only (i.e., without any time-consuming techniques, such as frequency- or optimization- or deep learning-based detections) with both previous and current frame information in order to make the final goal of intelligent pig monitoring run in real time.

Based on the experimental results for 432 video frames (including 3888 pigs) over 24 h, we confirmed that all 3888 pigs could be detected correctly (while the accuracy with ground-truth was 0.79) in real time (i.e., 114 FPS). Compared with the state-of-the-art deep learning-based methods, the proposed method could detect pigs more accurately and more quickly. We will extend this study to develop a real-time tracking system for individual pigs over 24 h for the management of individual pigs as the final goal.

Author Contributions: Y.C., D.P. and J.C. conceptualized and designed the experiments; J.S., Y.C., and H.L. collected the video data; Y.C. and J.S. designed and implemented the detection system; H.L. and J.S. validated the proposed method; J.S., H.L., and Y.C. wrote the paper.

Funding: This research was supported by a Korea University Grant.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Banhazi, T.; Lehr, H.; Black, J.; Crabtree, H.; Schofield, P.; Tschärke, M.; Berckmans, D. Precision Livestock Farming: An International Review of Scientific and Commercial Aspects. *Int. J. Agric. Biol.* **2012**, *5*, 1–9.
2. Neethirajan, S. Recent Advances in Wearable Sensors for Animal Health Management. *Sens. Bio-Sens. Res.* **2017**, *12*, 15–29. [[CrossRef](#)]
3. Tullo, E.; Fontana, I.; Guarino, M. Precision Livestock Farming: An Overview of Image and Sound Labelling. In Proceedings of the 6th European Conference on Precision Livestock Farming (EC-PLF 2013), Leuven, Belgium, 10–12 September 2013; pp. 30–38.
4. Matthews, S.; Miller, A.; Clapp, J.; Plötz, T.; Kyriazakis, I. Early Detection of Health and Welfare Compromises through Automated Detection of Behavioural Changes in Pigs. *Vet. J.* **2016**, *217*, 43–51. [[CrossRef](#)] [[PubMed](#)]
5. Tschärke, M.; Banhazi, T. A Brief Review of the Application of Machine Vision in Livestock Behaviour Analysis. *J. Agric. Inform.* **2016**, *7*, 23–42.
6. Han, S.; Zhang, J.; Zhu, M.; Wu, J.; Kong, F. Review of Automatic Detection of Pig Behaviours by using Image Analysis. In Proceedings of the International Conference on AEECE, Chengdu, China, 26–28 May 2017; pp. 1–6. [[CrossRef](#)]
7. Cook, N.; Bench, C.; Liu, T.; Chabot, B.; Schaefer, A. The Automated Analysis of Clustering Behaviour of Piglets from Thermal Images in response to Immune Challenge by Vaccination. *Animal* **2018**, *12*, 122–133. [[CrossRef](#)] [[PubMed](#)]
8. Brunger, J.; Traulsen, I.; Koch, R. Model-based Detection of Pigs in Images under Sub-Optimal Conditions. *Comput. Electron. Agric.* **2018**, *152*, 59–63. [[CrossRef](#)]
9. Tu, G.; Karstoft, H.; Pedersen, L.; Jorgensen, E. Illumination and Reflectance Estimation with its Application in Foreground. *Sensors* **2015**, *15*, 12407–12426. [[CrossRef](#)]
10. Tu, G.; Karstoft, H.; Pedersen, L.; Jorgensen, E. Segmentation of Sows in Farrowing Pens. *IET Image Process.* **2014**, *8*, 56–68. [[CrossRef](#)]
11. Tu, G.; Karstoft, H.; Pedersen, L.; Jorgensen, E. Foreground Detection using Loopy Belief Propagation. *Biosyst. Eng.* **2013**, *116*, 88–96. [[CrossRef](#)]
12. Nilsson, M.; Herlin, A.; Ardo, H.; Guzhva, O.; Astrom, K.; Bergsten, C. Development of Automatic Surveillance of Animal Behaviour and Welfare using Image Analysis and Machine Learned Segmentation Techniques. *Animal* **2015**, *9*, 1859–1865. [[CrossRef](#)]

13. Kashiha, M.; Bahr, C.; Ott, S.; Moons, C.; Niewold, T.; Tuytens, F.; Berckmans, D. Automatic Monitoring of Pig Locomotion using Image Analysis. *Livest. Sci.* **2014**, *159*, 141–148. [[CrossRef](#)]
14. Oczak, M.; Maschat, K.; Berckmans, D.; Vranken, E.; Baumgartner, J. Automatic Estimation of Number of Piglets in a Pen during Farrowing, using Image Analysis. *Biosyst. Eng.* **2016**, *151*, 81–89. [[CrossRef](#)]
15. Ahrendt, P.; Gregersen, T.; Karstoft, H. Development of a Real-Time Computer Vision System for Tracking Loose-Housed Pigs. *Comput. Electron. Agric.* **2011**, *76*, 169–174. [[CrossRef](#)]
16. Khoramshahi, E.; Hietaoja, J.; Valros, A.; Yun, J.; Pastell, M. Real-Time Recognition of Sows in Video: A Supervised Approach. *Inf. Process. Agric.* **2014**, *1*, 73–82. [[CrossRef](#)]
17. Nasirahmadi, A.; Hensel, O.; Edwards, S.; Sturm, B. Automatic Detection of Mounting Behaviours among Pigs using Image Analysis. *Comput. Electron. Agric.* **2016**, *124*, 295–302. [[CrossRef](#)]
18. Nasirahmadi, A.; Hensel, O.; Edwards, S.; Sturm, B. A New Approach for Categorizing Pig Lying Behaviour based on a Delaunay Triangulation Method. *Animal* **2017**, *11*, 131–139. [[CrossRef](#)] [[PubMed](#)]
19. Nasirahmadi, A.; Edwards, S.; Matheson, S.; Sturm, B. Using Automated Image Analysis in Pig Behavioural Research: Assessment of the Influence of Enrichment Substrate Provision on Lying Behaviour. *Appl. Anim. Behav. Sci.* **2017**, *196*, 30–35. [[CrossRef](#)]
20. Navarro-Jover, J.; Alcaniz-Raya, M.; Gomez, V.; Balasch, S.; Moreno, J.; Grau-Colomer, V.; Torres, A. An Automatic Colour-based Computer Vision Algorithm for Tracking the Position of Piglets. *Span. J. Agric. Res.* **2009**, *7*, 535–549. [[CrossRef](#)]
21. Guo, Y.; Zhu, W.; Jiao, P.; Chen, J. Foreground Detection of Group-Housed Pigs based on the Combination of Mixture of Gaussians using Prediction Mechanism and Threshold Segmentation. *Biosyst. Eng.* **2014**, *125*, 98–104. [[CrossRef](#)]
22. Guo, Y.; Zhu, W.; Jiao, P.; Ma, C.; Yang, J. Multi-Object Extraction from Topview Group-Housed Pig Images based on Adaptive Partitioning and Multilevel Thresholding Segmentation. *Biosyst. Eng.* **2015**, *135*, 54–60. [[CrossRef](#)]
23. Buayai, P.; Kantanukul, T.; Leung, C.; Saikaew, K. Boundary Detection of Pigs in Pens based on Adaptive Thresholding using an Integral Image and Adaptive Partitioning. *CMU J. Nat. Sci.* **2017**, *16*, 145–155. [[CrossRef](#)]
24. Lu, M.; Xiong, Y.; Li, K.; Liu, L.; Yan, L.; Ding, Y.; Lin, X.; Yang, X.; Shen, M. An Automatic Splitting Method for the Adhesive Piglets Gray Scale Image based on the Ellipse Shape Feature. *Comput. Electron. Agric.* **2016**, *120*, 53–62. [[CrossRef](#)]
25. Lu, M.; He, J.; Chen, C.; Okinda, C.; Shen, M.; Liu, L.; Yao, W.; Norton, T.; Berckmans, D. An Automatic Ear Base Temperature Extraction Method for Top View Piglet Thermal Image. *Comput. Electron. Agric.* **2018**, *155*, 339–347. [[CrossRef](#)]
26. Jun, K.; Kim, S.; Ji, H. Estimating Pig Weights from Images without Constraint on Posture and Illumination. *Comput. Electron. Agric.* **2018**, *153*, 169–176. [[CrossRef](#)]
27. Kang, F.; Wang, C.; Li, J.; Zong, Z. A Multiobjective Piglet Image Segmentation Method based on an Improved Noninteractive GrabCut Algorithm. *Adv. Multimed.* **2018**, *2018*, 108876. [[CrossRef](#)]
28. Yang, A.; Huang, H.; Zhu, X.; Yang, X.; Chen, P.; Li, S.; Xue, Y. Automatic Recognition of Sow Nursing Behaviour using Deep Learning-based Segmentation and Spatial and Temporal Features. *Biosyst. Eng.* **2018**, *175*, 133–145. [[CrossRef](#)]
29. Yang, Q.; Xiao, D.; Lin, S. Feeding Behavior Recognition for Group-Housed Pigs with the Faster R-CNN. *Comput. Electron. Agric.* **2018**, *155*, 453–460. [[CrossRef](#)]
30. Kongsro, J. Estimation of Pig Weight using a Microsoft Kinect Prototype Imaging System. *Comput. Electron. Agric.* **2014**, *109*, 32–35. [[CrossRef](#)]
31. Lao, F.; Brown-Brandl, T.; Stinn, J.; Liu, K.; Teng, G.; Xin, H. Automatic Recognition of Lactating Sow Behaviors through Depth Image Processing. *Comput. Electron. Agric.* **2016**, *125*, 56–62. [[CrossRef](#)]
32. Stavrakakis, S.; Li, W.; Guy, J.; Morgan, G.; Ushaw, G.; Johnson, G.; Edwards, S. Validity of the Microsoft Kinect Sensor for Assessment of Normal Walking Patterns in Pigs. *Comput. Electron. Agric.* **2015**, *117*, 1–7. [[CrossRef](#)]
33. Zhu, Q.; Ren, J.; Barclay, D.; McCormack, S.; Thomson, W. Automatic Animal Detection from Kinect Sensed Images for Livestock Monitoring and Assessment. In Proceedings of the ICCIT, Liverpool, UK, 26–28 October 2015; pp. 1154–1157. [[CrossRef](#)]

34. Kulikov, V.; Khotskin, N.; Nikitin, S.; Lankin, V.; Kulikov, A.; Trapezov, O. Application of 3D Imaging Sensor for Tracking Minipigs in the Open Field Test. *J. Neurosci. Methods* **2014**, *235*, 219–225. [[CrossRef](#)] [[PubMed](#)]
35. Shi, C.; Teng, G.; Li, Z. An Approach of Pig Weight Estimation using Binocular Stereo System based on LabVIEW. *Comput. Electron. Agric.* **2016**, *129*, 37–43. [[CrossRef](#)]
36. Matthews, S.; Miller, A.; Plötz, T.; Kyriazakis, I. Automated Tracking to Measure Behavioural Changes in Pigs for Health and Welfare Monitoring. *Sci. Rep.* **2017**, *7*, 17582. [[CrossRef](#)] [[PubMed](#)]
37. Zheng, C.; Zhu, X.; Yang, X.; Wang, L.; Tu, S.; Xue, Y. Automatic Recognition of Lactating Sow Postures from Depth Images by Deep Learning Detector. *Comput. Electron. Agric.* **2018**, *147*, 51–63. [[CrossRef](#)]
38. Lee, J.; Jin, L.; Park, D.; Chung, Y. Automatic Recognition of Aggressive Pig Behaviors using Kinect Depth Sensor. *Sensors* **2016**, *16*, 631. [[CrossRef](#)] [[PubMed](#)]
39. Kim, J.; Chung, Y.; Choi, Y.; Sa, J.; Kim, H.; Chung, Y.; Park, D.; Kim, H. Depth-based Detection of Standing-Pigs in Moving Noise Environments. *Sensors* **2017**, *17*, 2757. [[CrossRef](#)] [[PubMed](#)]
40. Chung, Y.; Kim, H.; Lee, H.; Park, D.; Jeon, T.; Chang, H. A Cost-Effective Pigsty Monitoring System based on a Video Sensor. *KSII Trans. Internet Inf. Syst.* **2014**, *8*, 1481–1498.
41. Ju, M.; Choi, Y.; Seo, J.; Sa, J.; Lee, S.; Chung, Y.; Park, D. A Kinect-based Segmentation of Touching-Pigs for Real-Time Monitoring. *Sensors* **2018**, *18*, 1746. [[CrossRef](#)]
42. Zuo, S.; Jin, L.; Chung, Y.; Park, D. An Index Algorithm for Tracking Pigs in Pigsty. In Proceedings of the ICITMS, Hong Kong, China, 1–2 May 2014; pp. 797–803. [[CrossRef](#)]
43. Intel RealSense D435, Intel. Available online: <https://click.intel.com/intel-realsensetm-depth-camera-d435.html> (accessed on 28 February 2018).
44. Mallick, T.; Das, P.; Majumdar, A. Characterization of Noise in Kinect Depth Images: A Review. *IEEE Sens. J.* **2014**, *14*, 1731–1740. [[CrossRef](#)]
45. Singh, B.; Patel, S. Efficient Medical Image Enhancement using CLAHE and Wavelet Fusion. *Int. J. Comput. Appl.* **2017**, *167*, 1–5.
46. Eramian, M.; Mould, D. Histogram Equalization using Neighborhood Metrics. In Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05), Victoria, BC, Canada, 9–11 May 2005; pp. 397–404.
47. Otsu, N. A Threshold Selection Method from Gray-level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [[CrossRef](#)]
48. Nadimi, S.; Bhanu, B. Physics-based Models of Color and IR Video for Sensor Fusion. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'03), Tokyo, Japan, 1 August 2003; pp. 161–166.
49. Becker, S.; Scherer-Negenborn, N.; Thakkar, P.; Hübner, W.; Arens, M. The effects of camera jitter for background subtraction algorithms on fused infrared-visible video streams. In Proceedings of the Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII, Edinburgh, UK, 26–27 October 2016; p. 99950.
50. Yang, S.; Luo, B.; Li, C.; Wang, G.; Tang, J. Fast Grayscale-Thermal Foreground Detection with Collaborative Low-rank Decomposition. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2574–2585. [[CrossRef](#)]
51. Bouwmans, T.; Silva, C.; Marghes, C.; Zitouni, S.; Bhaskar, H.; Frelicot, C. On the Role and the Importance of Features for Background Modeling and Foreground Detection. *Comput. Sci. Rev.* **2018**, *28*, 26–91. [[CrossRef](#)]
52. Maddalena, L.; Petrosino, A. Background Subtraction for Moving Object Detection in RGB-D Data: A Survey. *J. Imaging* **2018**, *4*, 71. [[CrossRef](#)]
53. Open Source Computer Vision, OpenCV. Available online: <http://opencv.org> (accessed on 18 December 2016).
54. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242.
55. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
56. Chen, C.; Ross, A. A Multi-Task Convolutional Neural Network for Joint Iris Detection and Presentation Attack Detection. In Proceedings of the 2018 IEEE Winter Applications of Computer Vision Workshops (WACVW), Lake Tahoe, NV, USA, 15 March 2018; pp. 44–51.
57. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on CVPR, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

58. Buehler, M.; Iagnemma, K.; Singh, S. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 56.
59. He, Z.; Liang, Y.; Chen, L.; Ahmad, I.; Wu, D. Power-Rate-Distortion Analysis for Wireless Video Communication under Energy Constraint. *IEEE Trans. Syst. Video Technol.* **2005**, *15*, 645–658.
60. He, Z.; Cheng, W.; Zhao, X.; Moll, R.; Beringer, J.; Sartwell, J. Energy-Aware Portable Video Communication System Design for Wildlife Activity Monitoring. *IEEE Circuits Syst. Mag.* **2008**, *8*, 25–37. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).