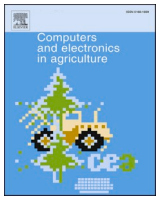
Computers and Electronics in Agriculture 184 (2021) 106095

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Original papers

Spatial segmentation for processing videos for farming automation☆

He Liu *, Amy R. Reibman, Aaron C. Ault, James V. Krogmeier

Purdue University, 501 Northwestern Ave, West Lafayette, IN 47907, USA

ARTICLE INFO

Keywords:
Spatial segmentation
Video processing
Agriculture

ABSTRACT

A camera mounted on the front of a large agricultural machine captures a rich collection of visual data. Powerful cues about the upcoming field can be extracted through video processing. However, to access these cues requires methods to focus only on a specific region of the video frame, for example, the region containing the vehicle attachment or the upcoming field. To separate these different spatial regions in farming videos, this paper presents a spatial segmentation method using a rapidly-trained classifier. This classifier is trained on low-level hand-crafted features with limited data and can be easily adapted to different farming applications. We consider two applications here: classifying farming activities and automatic control to lift the header of a combine harvester. We demonstrate experimentally that the segmentation algorithm enables activity classification accuracy of 87%, as well as a prediction error of about 1.3 s on the correct time to lift the combine header.

1. Introduction

Recently, new engineering technologies, such as sensing and robotics, have been applied to challenges with the agriculture industry (King et al., 2017). Traditional farming activities such as planting and harvesting also benefit from these technologies by using advanced farming machines (vehicles), including tractors and combine harvesters. Applications have been designed to improve different aspects of the farming machines including autonomous control. Completely autonomous systems are still difficult to build now, and our target is to develop systems to help and assist the control process for these machines.

A normal autonomous system requires sensors to collect signals from the surrounding environment. Compared to traditional sensors such as GPS and RADAR used in farming machines, cameras can provide a large amount of visual data efficiently, and the data can be easily interpreted for human analysis. With the help of image processing methods and video analytics, cameras have been applied in automation systems for farming vehicles (Boursianis et al., 2020; Gupta et al., 2020; Chen et al., 2017; Liu et al., 2019).

However, farming machines are more complex than automobiles. In addition to steering and speed control, machines like a tractor or a combine harvester have a tool, or attachment, whose interaction with the crops or the field must be controlled. Each farming activity requires a distinct attachment with its own requirements for controlling the interaction.

Fig. 1 shows three different farming activities and their attachments: corn chopping, wheat harvesting and tillage. The color, shape, and motion of the attachments are all distinctive, which make them effective cues to separate one activity from another. Automating the interaction between the attachment and the field requires identifying the activity and isolating the location of the attachment in the image. For example, when harvesting, the height of the attachment (called the header) should be adjusted based on the condition of the approaching field.

In addition, processing videos in practical farming vehicle applications is also challenging. Different applications have distinct requirements which determine where the camera should be located, and their processing techniques are not the same. In Kurita et al. (2012), cameras that are placed near the auger of a combine harvester are used to automate the unloading process, while in (Liu et al., 2018) dash cameras are mounted inside the cockpit of farming vehicles to capture the front view of the operator. But one common challenge for cameras is that the captured images normally include some unrelated areas which are not useful for further analysis (Cho et al., 2014). As a result, the most fundamental step for practical image analysis is to identify which region of the image is useful for the application. Unlike many video-based applications whose goal is to detect objects, instead we need to locate distinct regions of the image for further analysis.

Our video data are collected from multiple farms in the US between 2016 and 2018. For the purpose of control automation, the cameras are placed by farmers inside the cockpit of large agriculture vehicles to

☆ This work was supported by the Foundation for Food and Agricultural Research Grant #534662 and the Open Ag Technologies and Systems (OATS) Group.
* Corresponding author.
E-mail address: liu1433@purdue.edu (H. Liu).

https://doi.org/10.1016/j.compag.2021.106095
Received 21 May 2020; Received in revised form 27 February 2021; Accepted 7 March 2021

capture a view similar to that observed by a human operator. Notice that we refer to these captured videos as farming videos for the rest of this paper. Typical dashboard cameras (dash cams) are used for capture because they can be easily mounted on the windows of the farming machines, and the cameras are always pointed towards either the front side or back side of the vehicles based on the farming activities. Fig. 1 shows some example frames.

In our farming videos, there are normally three common regions captured from the cockpit. The first important region is the header or attachment region, which is normally connected to the front side. Different farming activities use various types of attachments. For example, the first row of Fig. 1 shows two different attachments used for corn chopping, although the attachment is blocked by the corn on the left image. The second important region is the upcoming field in front of the vehicle. There are many types of fields and each has a unique appearance. Our collection mainly covers three categories: soybean, wheat and corn. But for the same type of field, the color and shape can be different; see for example, the two types of wheat fields in the second row of Fig. 1. The third region includes all other parts such as sky or faraway objects that have no large motion. These low-motion regions are crucial in multi-camera systems because the feature points in this region can be used to connect different cameras. As a result, our target is to segment the farming videos into these three regions.

However, this farming video segmentation problem differs from the typical video segmentation such as (Liu et al., 2020; Karegowda et al., 2021). First, there are no particular objects to detect in farming videos, and our target is instead to divide the frame into different regions. In object detection, the appearance of the target (like shape and color), do not dramatically change over time. But for region detection, the content inside the region is not determined, and it also changes across time. Second, since the cameras are manually mounted by farmers, the capturing angles and scene structures of the videos are not controllable, shown in Fig. 1. This requires the segmentation method to be robust to all these challenges caused during capturing; for example, it should be able to quickly adapt to new viewing angles. Third, there are many practical constraints like time and hardware. The segmentation should process videos as quickly as possible because it needs to save time for further analysis, especially for real-time applications. It is also not practical to install powerful machines on farming vehicles, which limits the computational power of the method. Currently, we have not seen any related applications which address all the issues listed above.

In this work, we present a robust training-based segmentation method that effectively segments scenes captured from a large farming machine. The current work is both generic and effective when applied to each of the individual vision-based farming tasks that we have considered earlier (Liu et al., 2018; Liu et al., 2019). In addition, our new method overcomes two limitations of both of our previous solutions; these had used specifically-tuned threshold parameters and had focused on identifying only parts of the video frame (Liu et al., 2018; Liu et al., 2019). The current segmentation method is a training-based algorithm

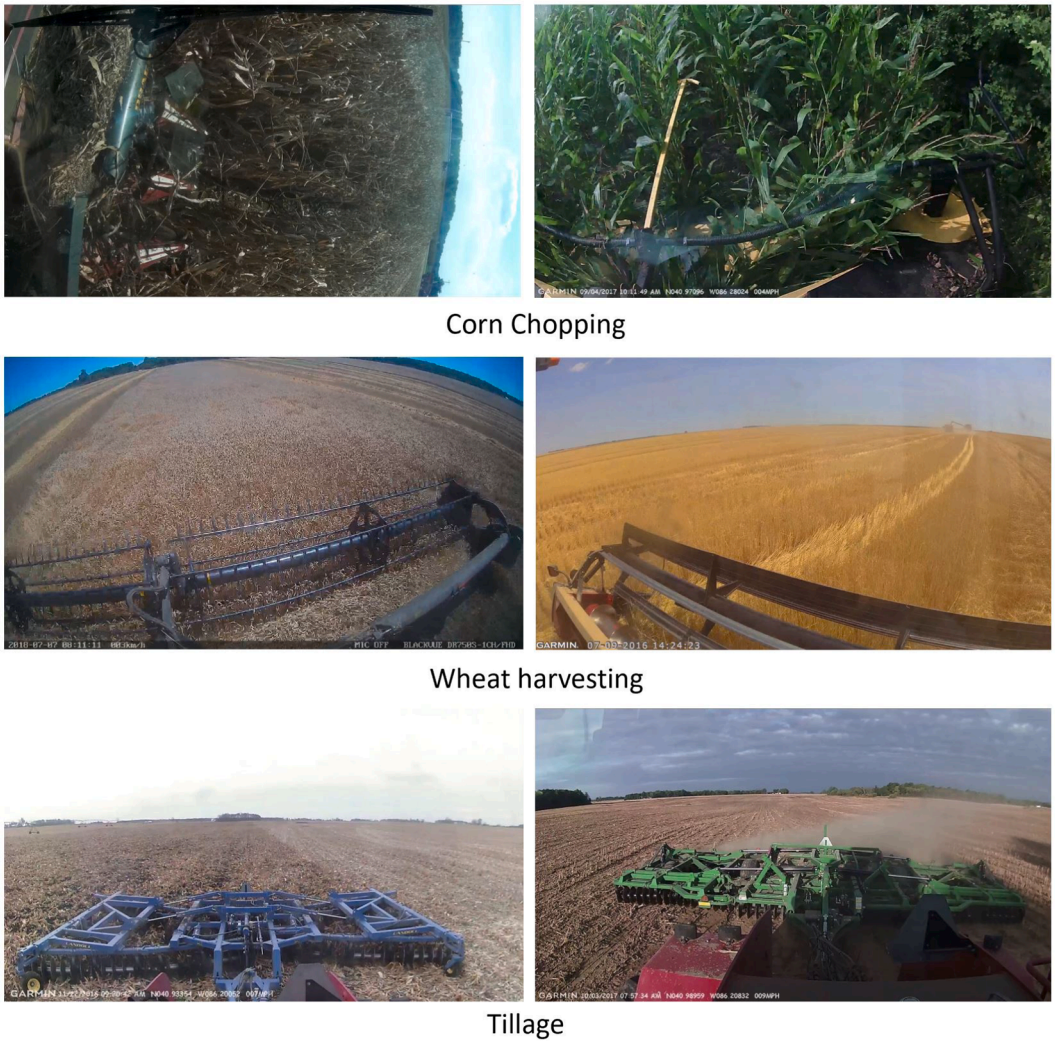


Fig. 1. Sample video frames captured by dash cameras mounted on different farming machines: chopping corn, harvesting, and tillage.

that segments all regions simultaneously and quickly adapts to different applications with a small number of training labels. We demonstrate here that our new method performs as well or better than our previous methods for each of the two tasks we considered before.

There are four main contributions in this work. First, we summarize the challenges associated with processing dash camera videos captured from farming vehicles. Second, a training-based spatial segmentation method is developed for farming videos which adapts easily to different farming applications. Third, we propose a generalized two-branch classification pipeline to improve video classification performance using domain knowledge. After that, we present another practical application which relies on the field region identified by the segmentation method, and it predicts the header-height[1] of a combine harvester.

This paper is organized as follows. We start by reviewing the related prior research on image and video processing in Section 2. Next, we introduce the farming videos and their unique challenges in Section 3, and propose the newly proposed spatial segmentation method in Section 4. Based on the new method, Section 5 and Section 6 describe the extended work from our previous research: farming activity classification and combine header-height prediction. Finally, Section 7 summarizes this work.

2. Previous work

2.1. Image and video segmentation

Image segmentation has been studied extensively. Typical image segmentation methods are based on color information (Felzenszwalb and Huttenlocher, 2004). But the color information is not robust enough on its own for segmenting outdoor farming videos (Liu et al., 2019). More recent methods apply Convolutional Neural Networks (CNN) to segment objects from the image; for example (Long et al., 2015) applies fully-connected networks. However, these methods are trained and tested with public datasets that do not contain relevant images related to agriculture or farming. Training a large image segmentation network for farming applications requires a large number of images and ground truth labels, and it is difficult to gather so much data.

Recently, CNN becomes popular for solving the Video Object Segmentation (VOS) problems (Pathak et al., 2017). However, CNN-based methods require large training datasets such as (Perazzi et al., 2016). These datasets include videos with clearly-labelled moving objects in the foreground. But for videos captured from farming vehicles, the foreground is not well-defined and it is hard to apply these methods to our segmentation problem. In addition, the videos from the popular datasets are not closely related to farming.

For practical applications, obtaining ground truth labels is time-consuming, so new methods are being developed to handle the label limitations. Semi-supervised methods train both labelled and unlabeled data together, while weakly-supervised methods use image-level labels on pixel-level segmentation tasks (Papandreou et al., 2015). These methods solve the problem of either lack of labels or low-quality labels, but they do not consider the computational load. As mentioned in Section 1, computational power is limited for practical farming video systems. As a result, although these previous work tackle some practical problems such as labelling, they are not suitable for farming applications.

2.2. Features for video classification

Farming applications require a system that can automatically classify videos into different categories or contexts. Traditional image and video classification algorithms extract features from the data and train classifiers with ground truth labels. Their performance largely depends on where the features are sampled and the corresponding feature descriptors. The popular feature extraction methods include traditional 2D key points such as SIFT, and 3D cuboids (Dollár et al., 2005). A trajectory-based (Wang and Schmid, 2013) feature sampling method can be more effective to capture the motions in video streams.

Feature descriptors are generated based on the sampled positions. Popular image feature descriptors include color features such as Content-Based Image Retrieval (CBIR) (Yue et al., 2011), and texture features such as the Gray Level Co-occurrence Matrix (GLCM) (Baraldi and Parmiggiani, 1995). Video feature descriptors are mainly extracted from 3D volumes based on the optical flow.

Recently, Convolutional Neural Networks (CNNs) have been widely applied to image and video classification, such as the PlacesCNN (Zhou et al., 2018) for image scene classification. However, most of the learning-based classification algorithms are not trained on farming-related data, and it is hard to find large-scale public datasets of farming images or videos. Moreover, these methods require significant computational power which is hard to applied in practical farming vehicles.

2.3. Farming video applications and CNNs

There are many video-related applications for farming vehicles. Video-based algorithms have been designed to detect the lateral cutting edges in corn chopping (Benson et al., 2003), and to adjust the tractor steering using video motion information for the corn (Sainz-Costa et al., 2011).

CNNs are powerful methods that have been applied to solve a variety of problems, including agricultural applications (Kamilaris and Prenafeta-Boldú, 2018; Kounalakis et al., 2019). However, our proposed method is not related to CNNs. First, the hardware constraint limits the computational power of this application, so it is not suitable to use a CNN. More importantly, the visual differences between our target regions can be easily characterized using simple classifiers. Therefore, there is no reason to apply more powerful methods, and the experiment in Section 4 shows that our hand-crafted features achieve better segmentation results. Also, using models like CNNs are more likely to cause over-fitting problems, especially when the training data is not extensive. In this paper, we train a light-weighted classifier with simple video features which can be easily incorporated into practical farming applications.

3. Farming videos and challenges

This section introduces the background of the practical vision-based farming application. We first talk about video data collection and then discuss some general challenges associated with processing these videos.

3.1. Farming video collection

As mentioned in Section 1, we use the term farming videos to indicate the videos that are captured from the cockpit of a machine such as a combine harvester or a tractor. However, the video collection process is not trivial. In general, it is time-consuming to collect videos in an outdoor environment for everyday farming activities (Liu et al., 2018). To the best of our knowledge, there are no embedded camera systems available for farming vehicles that can transfer videos directly for analysis, and there are few public farming datasets available. A multi-sensor dataset is published from Kragh et al. (2017) to detect obstacles on the farm, but their video data only lasts for two hours.

In addition, our target is to build a general segmentation system for different farming activities, and we need to collect various types of videos from different farming events. This is because the training-based

[1] In this paper, we refer the term header-height to the sum of the reel-height and actual header-height, and which equals to the distance of rotating reel to the ground in a combine harvester.

systems have poor performance on data they have not been seen before, and poor data variation causes troubles. However in practice, the timing of when most of the farming activities occur depends on factors such as human labor and weather, and most of these tasks are finished within a short period of time. This means that the farming videos must be collected within a limited time period, for example within several hours a day for a few days a year. Such videos contain the same repeating actions under the same environment for a long time, and special situations like field anomalies or machine breakdown rarely occur.

In summary, the video capturing environment makes it difficult to build one general segmentation method which works for all farming applications with limited computational power. However, it is possible to build a system with a quick learning process using limited data, so that it can be easily adapted to different farming applications. Our proposed segmentation method in Section 5 is designed based on this idea. It is trained on low-level hand-crafted features with limited data and provides robust segmentation on videos from different farming activities.

3.2. Challenges in processing farming videos

As mentioned in Section 1, our goal is to segment regions instead of particular objects. The structure and object in each region can change due to vehicle movement, and the image distortion also challenges the segmentation because of the outdoor environment (Liu et al., 2018). Outdoor illumination changes quickly and influences the color of every region in the video. Some regions could be blocked by shadows and window glare due to the direction of movement and sunshine. Apart from color, the outdoor regions also suffer from noise and blur distortion, which are caused by the dust in the field or foggy weather.

Motion information is robust to outdoor illumination changes, but the variety of motion from a farming vehicle is also challenging. For normal vehicle-mounted cameras, two types of motions are summarized in Liu et al. (2018): camera-induced motion and object motion in Liu et al. (2018). But both motion types are more complicated in farming videos compared to those in the videos from automobiles. Camera-induced motions are generated by the moving cameras, and farming vehicles are very shaky when moving in the field, which causes noises when analyzing the motions. Independent motions are caused by movements other than the camera, and most of these motions on farming videos come from the interaction between the machine and the crops. These motions are difficult to model and challenge the segmentation process.

Both the image distortion and complicated motions Based on such practical challenges, we propose our classification algorithm. We extract hand-crafted features of both color and motion information and train a classifier to segment images, which is effective shown in later experiments.

4. Training-based spatial segmentation

In this section, we first describe the problem of segmenting farming videos and then review the motion consistency measure from our previous work (Liu et al., 2019). Next we explain how to apply this measure to design new pixel-wise features for our proposed training-based segmentation method. After that, the comparison experiment of different segmentation methods is presented.

4.1. Motion consistency and segmentation

Both color and motion information are widely used in video spatial segmentation. In an outdoor farming environment, however, relying only on color features is insufficient due to the illumination variation and its affect on color. In particular, video motions are more effective to separate the moving objects. However, simple separation between the foreground moving objects and the background scene is not enough, because some object movement comes from both the camera motion and object motion. Most camera motions are consistent with the vehicle, such as forward, backward, and turning motions. Even though the fields in the farm are uneven and the vehicle can shake dramatically, we assume these camera motions are consistent over time because of the slow driving speed. Unlike camera motions, the object motions from the attachment regions are independent of the vehicle movement, and they are not consistent over time. They vary due to the different farming activities and the types of the corresponding machine. Normally, the motions in harvesting and chopping videos are chaotic, but tillage and spraying activities have weaker interactions and their motions are smaller.

To separate the camera motions and object motions, in Liu et al. (2019) we design a motion consistency measure C. This measure assumes the structure of video frames does not have sudden changes in a short period of time, which is valid because of low speed of farming vehicles. The measure is computed based on the optical flow over a block of consecutive video frames with length T:

$$C = \frac{1}{T}\sum_{t=1}^{T} \bullet\left[\frac{du_t}{dt} < \theta\right] \bullet \left[\frac{dv_t}{dt} < \theta\right] \tag{1}$$

where (u_t, v_t) is the optical flow at frame t, $\bullet$ represents the indicator function and θ is the minimum motion threshold. Here the optical flow values are computed based on every pixel position across frames. As a result, this measure shows the temporal consistency all over the frame.

This motion consistency measure is used to partition the frame in our previous work (Liu et al., 2019). However, that system depends on fixed threshold values and is not robust when applied in different farming activities. In this paper, we address this disadvantage by training a classifier with a limited number of labels, which makes it robust when applied to general farming videos.

4.2. Training-based segmentation

In order to separate spatial regions of general farming videos, we propose a training-based segmentation method which uses a classifier to separate pixel positions. Every pixel position in the frame is a classification unit and represented by a feature vector. Both color and motion information are used to form the feature vector, but their spatial positions are not included because the region positions of farming videos vary between different capture angles. The color features improve the spatial smoothness of the segmentation as shown later in Fig. 4. The color features are extracted using the Content-Based Image Retrieval (CBIR) method (Yue et al., 2011), which is the histogram of the pixels in the HSV color space over a set of pixel values, with 8 bins of illuminance channel and 6 bins of two color channels. The motion features are hand-crafted with length 14D, and are described in detail below. In total, each pixel position has a feature vector of 14D color feature and 14D motion feature; a summary of the features is shown in Table 1.

The motion features are extracted from the magnitudes of the optical flows over a video block. The feature vector includes two parts: a measure of the motion magnitude and the motion temporal consistency. The motion magnitudes are directly characterized by six percentiles ranging from 1% to 90% and the differences between two neighbor percentile values. The motion temporal consistency is summarized by

Table 1
The components of our hand-crafted features.

Feature		Dimension
CBIR color feature	luminance hist	8D
	color hist	6D
Motion feature	OF mag %	6D
	OF mag diff %	6D
	S_total	1D
	S_sbs	1D

two values: the total sum of motion S_{total} and the sum of Step-By-Step motion S_{sbs}, which are defined in Eq. (2) and (3).

$$S_{total} = \frac{1}{T}\sqrt{\left(\sum_{t=1}^{T} u_t\right)^2 + \left(\sum_{t=1}^{T} v_t\right)^2} \tag{2}$$

$$S_{sbs} = \frac{1}{T}\sqrt{\sum_{t=1}^{T}\left(\left(\frac{du_t}{dt}\right)^2 + \left(\frac{dv_t}{dt}\right)^2\right)} \tag{3}$$

The optical flow operations between different frames are at pixel level. This total sum of motion S_{total} describes the net movement of a pixel position during a period T, which is similar to the cumulative distance (Poleg et al., 2014) for video blocks. Here the net movement is computed as the sum of optical flows over time, where the periodical rotations are cancelled and the overall forward motion is preserved. The step-by-step motion sum S_{sbs} records the sum of the motion changes, which is computed as the sum of all flow differences (du_t/dt, dv_t/dt) between neighboring frames. By accumulating the differences, the rotating header region which is ignored in S_{total} can be highlighted. As a result, comparing these two types of motion sums can separate the camera motion and object motion, especially when the camera motions are basically forward or backward. Both consistency values and their difference are normalized and added to the motion feature vector.

Fig. 2 visualizes these two indicators. The main region in the sum of motion S_{total} identifies the field region and part of the rotating header, and the main region in the step-by-step motion S_{sbs} mainly indicates the rotating header region. Their difference mask shown in the bottom right image highlights the upcoming field region on the left side. Note that there is a smaller and brighter region in middle of the difference mask. It represents the conveyor belt (Fig. 6), which is also highlighted because the belt motions are also consistent.

Using these features, spatial segmentation can be achieved by classifying all pixels into different regions. Based on the training data, here we use a Random Forest (RF) classifier for two reasons. First, the classification is performed on each pixel position, which means the size of the training data is huge, and a RF classifier is much faster to train than other classifiers such as SVM. Another reason is that our feature vector contains multiple types of information like motions and color, and some features may be more useful. However, most linear classifiers treat all features equally without a preference, while the RF can focus more on the distinguishable parts of the feature vector (Pal, 2005).

In the next section, we compare the segmentation results using three different choices of features: color only, motion only and both features concatenated together. Every feature group is individually trained with a RF classifier.

4.3. Methods comparison experiment

4.3.1. Experiment preparation

This section compares different segmentation methods using farming videos and the manually-labelled ground truth. In total, 229 wheat and bean harvesting video clips are prepared in this experiment. These video clips are selected and pre-processed from the video dataset that we collected from the farms. During pre-processing, the raw videos are downsampled to the resolution of 480 × 272 and temporally segmented into video blocks with a length of 30 frames (1 s). The temporal segmentation enables us to assume that the video structure remains the same during a short period of time.

The pre-processed video clips are hand-labelled into the three spatial regions introduced in Section 1: the upcoming field region, the header (attachment) region, and low motion region (including sky and part of the body of vehicle). Each video is only labelled with one ground truth mask for all frames because the frame structure has little variation. As a result, there is no guarantee that a given pixel position has the same label across the entire second. Because of this, we only label the pixel positions that maintain a single region consistently across time. In other words, not all the pixels in the frame are labeled, but if the mask indicates that a pixel belongs to one category, this position will always belong to that category across the entire clip. As a result, there are some gaps (the black regions in the ground truth label in Fig. 4) in the labels, and this could influence the calculated segmentation performance as discussed below.

4.3.2. Implementation details

In our experiment, we train three RF classifiers using different features: the color, motion, and both features together. In a real application,

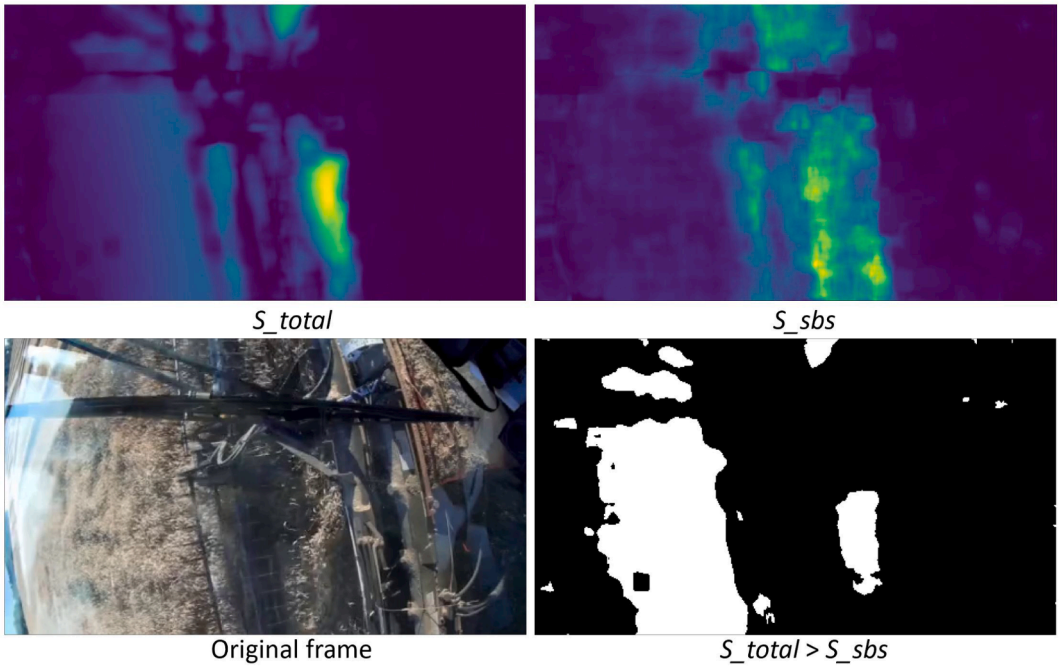


Fig. 2. Motion difference measure, with the white region in the bottom right mask shows where $S_{total} > S_{sbs}$. Notice the sky is on the left side.

the number of ground truth labels for segmentation is limited. So we choose only one sample video clip from the 229 video clips to be the training data for all three classifiers. Here one clip is sufficient to provide training data for pixel-level classification. In the actual training, 5000 pixel positions are randomly selected for each segmentation region. Notice all these clips are chosen from several farms, which means the one selected training clip could be similar to some of the testing clips, and we consider this issue when reporting the experiment results.

In this experiment, we compare our proposed methods with 5 different segmentation methods. Since there are no previous segmentation methods which solve a similar problem, here we test our videos on two popular VOS algorithms: Non-Local Consensus Voting (NLCV) (Faktor and Irani, 2014) and Saliency Aware Video Object Segmentation (SAVOS) (Wang et al., 2017). In addition, we compare with our two previous hand-crafted segmentation methods (Liu et al., 2018; Liu et al., 2019). Note that because the method in Liu et al. (2019) is only designed for the header region, in this experiment we extend it to create method (Liu et al., 2019)* by adding extra thresholds to separate all three regions. Furthermore, we also compare with a CNN-based method Deeplab (Chen et al., 2017). Although CNN-based methods are not practical to use for farming application, we still want to explore the potentials of these methods in the future. In this experiment, we fine-tune the last layer of the pre-trained Deeplab (Chen et al., 2017) network which is denoted by Deeplab (Chen et al., 2017)*. Here the training data are the same limited ground truth labels used to train the random forest classifiers which applies 100 estimators with maximum tree depth of 2, so we can obtain a fair comparison. The quantitative comparison results are presented in Table 2 and Fig. 3, and two sets of visual segmentation results are presented in Fig. 4.

4.3.3. Quantitative comparison results

All 228 testing video clips are quantitatively measured with the ground truth masks. The segmentation measures are based on all three regions, together with the overall frame. For each region, the Intersection Over Union (IOU) percentages are computed for all the testing videos, and we report their mean and standard deviation (STD) values. Here, the STD value is reported because the method is expected to generate robust segmentation on videos from different scenes. Methods with a higher mean percentage and a smaller STD value show better and robust performance. We first show the general performance comparison between different methods using a cumulative accuracy curve, and then the detailed numerical measures are explained later.

A cumulative IOU percentage plot of the accuracy on the overall frame measure is presented in Fig. 3. Each line represents the IOU measures of one method, and its detailed mean and STD values are shown in the last two rows in Table 2. The x axis of Fig. 3 shows the number of the test videos and y is the normalized sum of all IOU percentages at this number of testing videos. For example, when x value is 100, the corresponding y value is the sum of all testing videos from 1 to 100. In general, a line that reaches higher y values means this method has better accuracy, and the straightness of a line represents the robustness across different input videos. Notice among all testing videos, there are some clips which are similar to the training data, and we use a black vertical line to separate them: testing clips on the left side (first 42 clips) have similar structure as the training video, while the others do not.

From this figure, we can see that two proposed random forest methods *RF_motion* (pink) and *RF_both* (gray) are above the other lines and are relatively straighter than the others. Methods like (Liu et al., 2019)* and Deeplab* are not as robust. Our previous fixed threshold method (Liu et al., 2019)* (red) is consistently below the new training-based method. Specifically for Deeplab* (purple), applied to the first 42 testing videos, reaches similar performance with other methods, but its performance starts to decrease after the vertical separation line (better viewed in the zoomed-in box) when the testing data have different scene structures.

The detailed numerical measures of all the methods are reported in Table 2. Each row shows the IOU of a target region and each column represents a segmentation method. In general, the RF method with both color and motion features achieves better performance than other methods, with both high mean accuracy and stable STD values.

Among all the methods, three major comparisons from this table are discussed below. First, considering the mean accuracy, the improved hand-crafted method (Liu et al., 2019)* using fixed thresholds is slightly better than all training-based methods, but its STD value is much larger. This means fixed threshold values can be adjusted to make the average accuracy high, but the performance is not robust when applied to different video inputs. Second, comparing the Deeplab (Chen et al., 2017)* method with others, we can see it provides reasonable performance for the field region, but fails on other regions. This is mainly because the color of the field is much more similar across different testing videos. In addition, it has a high STD, which indicates a lack of robustness, potentially caused by the limited number of training clips. Third, the methods from the last two columns have similar performance: one uses only motion features and the other uses both motion and color features. But the STD values for the second method is slightly lower. This shows that adding color features has the potential to improve the robustness for segmentation.

4.3.4. Qualitative comparison results

In this section, we select two examples to show the visual segmentation results in Fig. 4, which present some segmentation results from two testing videos. The example clip (a) is captured from the same farming vehicle as the training data, but example (b) is from a different scene. The blue, green, and red regions respectively represent the header (attachment) region, the upcoming field region, and the low-motion region. The black region represents unlabeled gaps in the ground truth or pixels that are not labeled by the algorithm.

In this figure, the SAVOS (Wang et al., 2017 and Liu et al., 2018) only segment the header region, and SAVOS is not accurate enough because of the poor performance of the super-pixel segmentation. Notice the Deeplab (Chen et al., 2017)* method works well on the left example because this testing video is similar to the training data, but it fails on the right one because of the scene difference. This scene change also

Table 2
The segmentation comparison with different methods, presented by the mean and standard deviation (STD) of IOU percentages. Higher mean percentage and lower STD values means a better performance. The bold number indicates the best performance of each row.

Measured regions		NLCV (Faktor and Irani, 2014)	SAVOS (Wang et al., 2017)	Liu et al. (2018)	Liu et al. (2019)*	Deeplab (Chen et al., 2017)*	RF_color	RF_motion	RF_both
header	mean	0.17	0.379	0.608	**0.617**	0.499	0.375	0.614	0.615
	STD	0.126	0.136	0.107	0.133	0.185	0.159	**0.092**	**0.092**
field	mean	N/A	N/A	N/A	**0.646**	0.63	0.278	0.622	0.619
	STD	N/A	N/A	N/A	0.28	0.218	0.204	0.209	**0.203**
low_motion	mean	N/A	N/A	N/A	0.35	0.457	0.313	0.493	**0.511**
	STD	N/A	N/A	N/A	0.252	0.289	**0.164**	0.197	0.189
overall_frame	mean	0.146	0.198	0.291	0.652	0.668	0.649	0.714	**0.717**
	STD	0.053	0.051	0.094	0.118	0.161	0.136	0.095	**0.092**

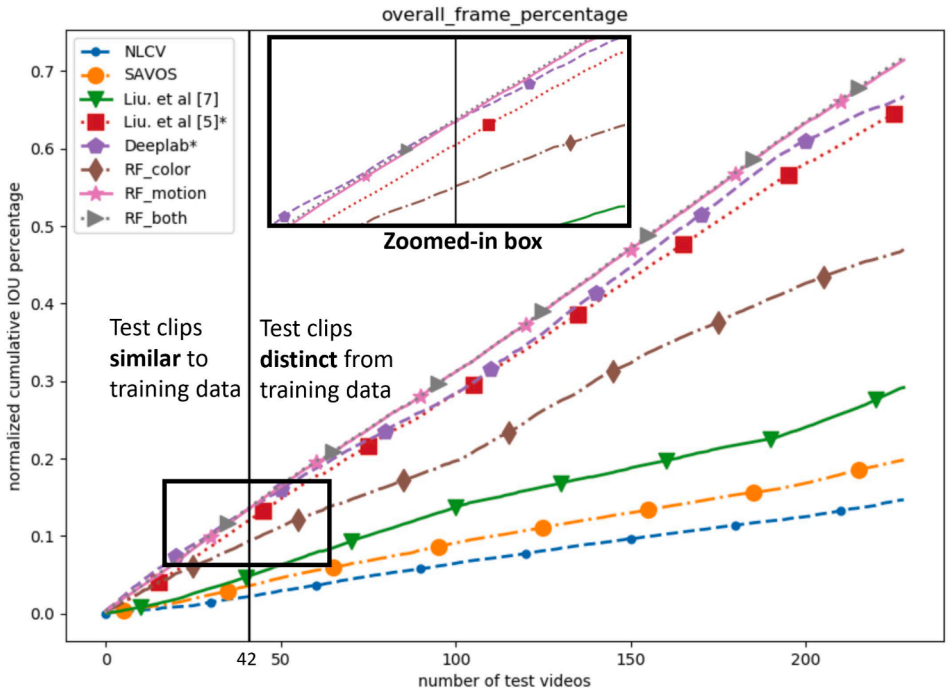


Fig. 3. The normalized cumulative IOU percentages on the overall frame measure. X axis is the number of videos, Y axis is the normalized sum of previous IOU percentages. For the vertical line, the left side shows the clips similar to the training data, the right side shows testing clips different from the training data. A zoomed-in plot is provided in the black box.

explains the drop of its curve in Fig. 3 after video number 40.

Our previous methods (Liu et al., 2018 and Liu et al., 2019) are specifically designed with fixed threshold parameters to locate only the header (blue) region and field (green) region respectively. The improved (Liu et al., 2019)* can perform multiple-region segmentation based on fixed threshold values. But these fixed values are not robust so the method misses some regions and has unlabelled black areas for the left video. Considering our proposed RF methods, their performances largely depend on the features they used. Color features over-segment the left clip, but fail on the right one because of the color difference. Motion features are more robust in general, but adding color features together corrects some miss-classified regions shown in both examples.

4.3.5. *General discussion*

In general, the new training-based segmentation method has four advantages for farming video segmentation. First, our method is robust to different farming applications. Using color and motion features can achieve segmentation in real-time with machines equipped with normal computational power and storage, such as a single-board computer. Second, with a quick training process, no fixed thresholds are required, and it is capable of handling practical challenges. Third, unlike complicated models such as a CNN, this classifier uses few labels for training, but achieves similar or better segmentation results. In addition, the new classifier segments all three spatial regions in one step, which can be directly applied to applications focusing on different regions. In the next two sections, we introduce two farming applications which use different regions based on the segmentation.

5. Generalized two-branch video classification

This section proposes a generalized two-branch video classification system that we then applied to classify farming activities. We first discuss the motivation of classifying farming videos, and then introduce the generalized classification system. Next we show how to incorporate segmentation to this two-branch system on farming videos, and finally a validation experiment is presented.

5.1. *Farming activity classification*

Many different farming activities may happen on the field and each activity needs a specific rule or strategy for further processing. This requires the collected videos to be categorized first. However, when performing their daily work, farmers cannot spare the time to identify the current activity, nor provide labels for the system design. So it is worthwhile to design a video classification system to detect what activity is happening in each video.

In video classification, motion is useful but can be chaotic in farming videos. Many of the public activity classification datasets are captured from static cameras (Chaquet et al., 2013), where all motion features are generated by the foreground objects such as human motion. However, the dash camera videos captured from farming vehicles have strong camera motions which are not effective for distinguishing among farming activities. Many of the camera motion estimation methods apply feature-point matching to cancel the global motions (Wang and Schmid, 2013). But in farming videos, there are few robust feature points available.

5.2. *Generalized two-branch classification pipeline*

In real applications, video classification methods that use domain knowledge typically perform better than methods that are designed for a general scenario (Onofri et al., 2016). Based on the domain knowledge, the specific targeted methods can be designed to extract the unique domain features that better distinguish the different categories. Such specifically-designed systems have been developed in different industries (Petscharnig and Schöffmann, 2018). However, these unique characteristics are not always applicable to all input data, and there are always exceptions. For example, the object motions of farming videos are useful to recognize farming activities, but some raw videos may be static or have little object motion. As a result, classifying videos with only the specific domain features is not enough.

In Liu et al. (2018), we proposed a two-branch classification pipeline, which applies two specific feature sampling strategies on different

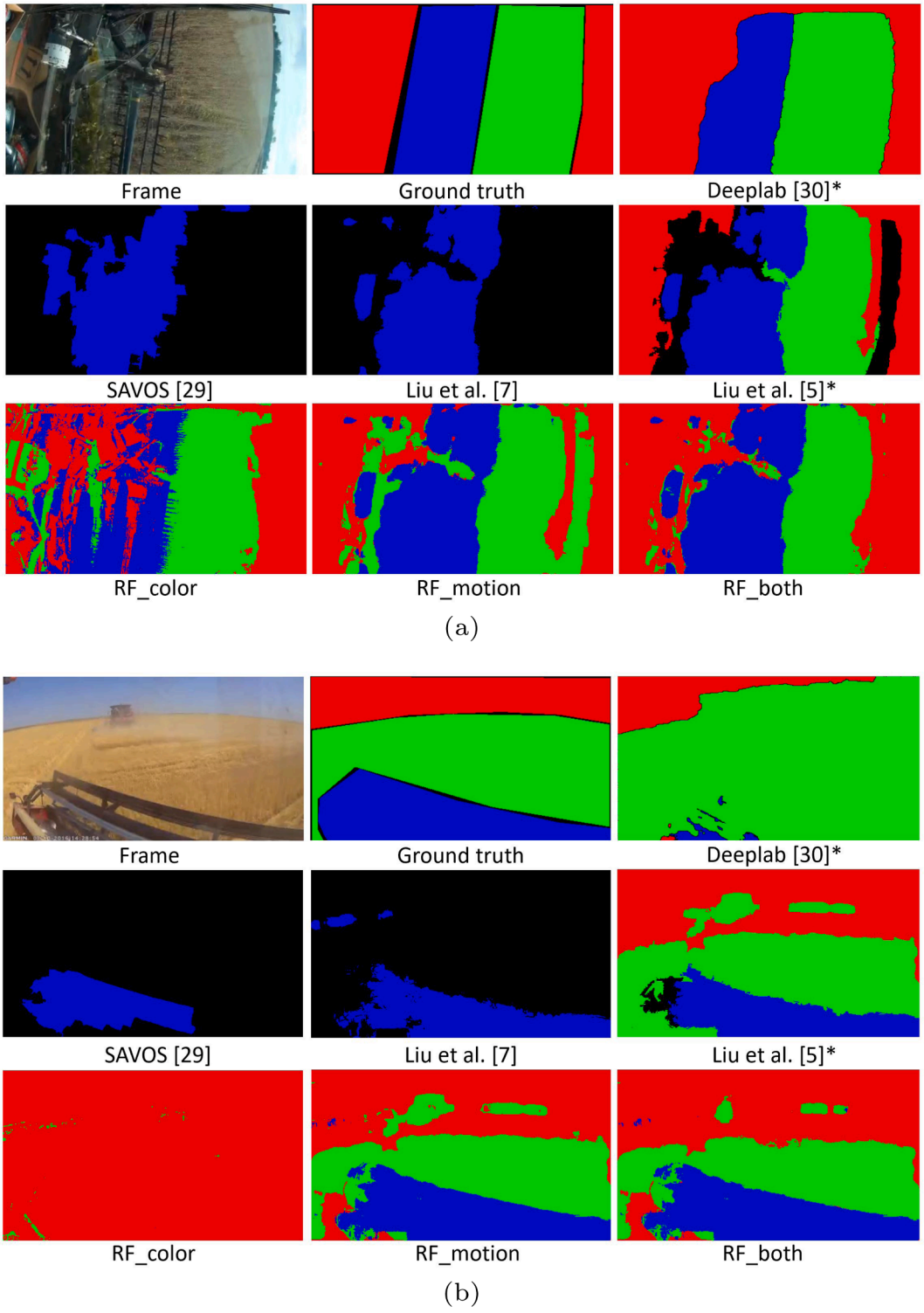


Fig. 4. Two segmentation results. Example (a) has the similar structure as the training data, and example (b) is from a different farm. The blue region is the header or attachment, green region is the upcoming field region, and red region is low-motion region. The black region is unlabeled.

videos. In this paper, we expand this framework to a generalized two-branch classification pipeline, which is shown in Fig. 5. The upper branch processes all input videos, but unlike our previous design, this branch can incorporate any general classification method. The second branch (shown in the red dashed box) selects those videos with unique domain-related features and processes them with specific methods. There is an activation scheme of the second branch to decide whether the input video has characteristics that will lead to better classification with the specifically-designed classifier (Branch 2). This scheme can be determined by different indicators based on domain knowledge, such as amount of object motion in farming video classification. Finally, the results from the general and specific classifiers are merged to improve the overall accuracy. A number of different score fusion methods can be applied to form the final decision, such as simple averaging and weighted sum. Depending on the reliability of the classifiers, it is also possible to train a classifier that learns the weights between multiple

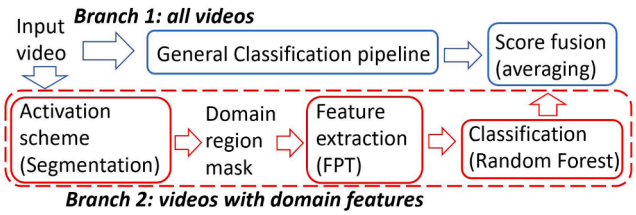


Fig. 5. The generalized two-branch video classification pipeline. The farming video classification is used in the second branch.

branches (Peng et al., 2016). The next subsection describes the specific choices we made to apply this pipeline for farming video classification.

5.3. Two-branch classification on farming videos

This subsection shows how we apply the two-branch pipeline for farming video classification. Farming videos have strong domain features, and the most effective cues to distinguish different farming activities are in the attachment region. In particular, the attachment interacts with the vehicle and the field in a way that is unique for each activity. As a result, the motion information from the attachment region is an effective choice for the feature in the second branch, and our proposed segmentation method can be applied as the activation scheme to sample these distinguishing features.

Based on this domain knowledge, we apply the generalized two-branch framework to classify the farming videos. The second branch in Fig. 5 is specifically designed to sample the object motions based on the output of the segmentation. The second branch is only activated if the segmentation method finds the object motion region, otherwise only the general classifier in Branch 1 is used. Then, features are extracted in the identified regions using the Fixed Position Trajectory (FPT) (Liu et al., 2018) methodology. FPT is based on the idea of optical flow stacking (Simonyan and Zisserman, 2014), tracks the motion at every fixed pixel position over a variable number of frames, and extracts features with the cuboid-based dense sampling method (Dollár et al., 2005). A random forest classifier processes these features, and we use direct averaging as the final score fusion method.

5.4. Experiment on farming video classification

This section presents a farming activity classification experiment that demonstrates that adding Branch 2 improves the overall performance of the classification system. Due to the limited farming data, only three farming activities are classified: tillage, corn chopping and wheat/bean harvesting, which are shown in Fig. 1. All the video clips used in this experiment have a fixed length of 5 s, and they are randomly selected from the raw farming videos. Each clip is manually labeled with a farming activity label. To measure the robustness of the system, we separate all the video clips into training and testing groups based on the time they were captured. The training clips are all selected from 2016 and 2017, and the testing clips are from other days during 2017 and 2018. For each activity, we select 500 clips for training and 500 clips for testing, which equals to 1500 training clips and 1500 testing clips in total.

In the experiment, we choose different general classifiers as the baseline (Branch 1 in Fig. 5) and compare them with the overall accuracy after adding Branch 2. Four general video classification systems are tested as Branch 1: the Improved Dense Trajectory (IDT) (Wang and Schmid, 2013), the dense-extracted features (Uijlings et al., 2015), the first branch classifier from Liu et al. (2018), and a video-based 3D Convolutional Neural Network (C3D) model (Hara et al., 2018). We implement the dense cuboid method (Uijlings et al., 2015) and our previous method (Liu et al., 2018). The original implementation of IDT feature extraction (Wang and Schmid, 2013) is directly used. For the CNN method (Hara et al., 2018), we apply the pre-trained C3D network and select the last layer from the network as the feature vector. So the farming videos are input to the network and the output features are used to train a random forest classifier. Note that this pre-trained model uses the ResNet architecture (He et al., 2016) that is trained on Kinetics human action dataset (Kay et al., 2017).

Next we describe the implementation details of Branch 2. Four segmentation methods from Section 4 are selected to activate Branch 2: the fine-tuned Deeplab (Chen et al., 2017)*, our previous method (Liu et al., 2018), the improved (Liu et al., 2019)*, and the new proposed *RF_motion*. These segmentation methods produce spatial masks, but the rest of the classification processes are the same. We extract video features including HOG, HOF and MBH from the masks using FPT. The Fisher Vector (FV) from Krapac et al. (2011) is used to encode the feature vector and a random forest classifier is trained as the classifier for this branch. The final decision is made by averaging the scores from both branches.

The classification results are shown in Table 3. In this table, the second column shows the classification results with Branch 1 only, and the right four columns present the overall accuracy after adding the second branch with different segmentation methods. Comparing these columns, it can be observed that the performance of all four general classifiers are improved after adding the second branch. The overall best performance is achieved by the C3D (Hara et al., 2018) method as a general classifier plus the cuboid-based method with *RF_motion* segmentation.

Comparing four segmentation methods for feature sampling in the second branch, we can see the proposed *RF_motion* method has the best performance. Note that each segmentation result activates a different subset of videos to be processed by the Branch 2 classifier. So we do not report accuracy for Branch 2 alone because it does not operate in isolation. But inaccurate segmentation can cause unnecessary computation such as the Deeplab (Chen et al., 2017)* method, which activates Branch 2 for almost all testing videos. It increases both time and computational requirements for this method relative to the other segmentation methods.

Comparing the four general classifiers in Table 3, the improvements provided by the second branch vary. This is mainly due to the feature sampling and extraction strategies used in the two branches. The features from Branch 1 are extracted from the whole frame, while features in Branch 2 are hand-crafted to concentrate more on object motions. However, the similarities between the feature-sampling strategies of two branches limit the improvements that can be obtained by adding Branch 2. From the table, two classifiers in Branch 1, (Liu et al., 2018) in row 3 and (Uijlings et al., 2015) in row 4, also depend on cuboid-based feature extraction methods and they provide limited improvement. The dense trajectories (Wang and Schmid, 2013) work poorly as Branch 1, but

Table 3
The comparison of video classification methods. Note the right four columns show the results by adding Branch 2, and each column uses different segmentation method. The C3D (Hara et al., 2018)* on the bottom row is used as feature extractor.

General Classifiers	Branch 1 only	Overall result adding Branch 2			
		(Chen et al., 2017)*	(Liu et al., 2018)	(Liu et al., 2019)*	RF_motion
IDT (Wang and Schmid, 2013)	0.591	0.661	0.618	0.619	0.754
Uijlings et al. (2015)	0.772	0.798	0.815	0.812	0.833
Liu et al. (2018)	0.768	0.792	0.80	0.819	0.834
C3D (Hara et al., 2018)*	0.818	0.869	0.826	0.830	**0.872**

Branch 2 helps to improve the overall accuracy. Comparing with traditional feature extraction methods, the pre-trained C3D (Hara et al., 2018) model better captures the general video characteristics, so adding the hand-crafted Branch 2 method compensates for the lack of domain-related farming features and achieves best performance.

In general, when applying this two-branch system to practical applications, the domain-related features would generally be hand-crafted by humans based on prior knowledge. As a result, choosing the general feature processing methods such as the cuboid-based feature extractor or learning-based feature extractor as the first branch can achieve better performance in the overall classification.

6. Combine header control

This section introduces another farming application. Based on analyzing the crop in the segmented field region, we predict the time when the header needs to be lifted on a combine harvester. We first introduce the basic background of a combine harvester and explain the workflow of the system. Finally, a validation experiment is presented.

6.1. Combine header control

A combine harvester is one of the most-widely used farming vehicles in agriculture. Fig. 6 shows some sample frames that were captured from the cockpit of a combine harvester. The red box indicates the reel or header and the green box is the conveyor belt that carries cut crops into the machine. The position of header (reel) can be adjusted as needed. The header should be low to the ground when there are crops to be cut, so no crops are left in the ground. But the header should be lifted when there are no crops in front to harvest, to prevent damage (Xie et al., 2013). As a result, the amount of uncut crops in front can be used to adjust the header-height. Most of the previous efforts (Xie and Alleyne, 2012) for combine header are mainly passive control and their common goal is to stabilize the header-height despite vehicle vibration. However, by incorporating visual information, the header-height can be actively predicted. While continuous height adjustment is possible, here we only consider the movement from low to high. In other words, we assume the combine harvester is harvesting crops with its header low, so there are crops in the upcoming field region by default, and our goal is to predict and automatically control when the header should be lifted.

The prediction is based on analyzing the upcoming field region, which is shown as the black box in Fig. 6. Estimating how the crop amount changes in the field region is the key step. However, designing this automatic prediction system is not trivial. First, it needs an accurate spatial segmentation of the upcoming field region as the target. Inaccurate field regions cause missing blocks which affects crop estimation. Second, comparing the left and right columns in Fig. 6, it is visually difficult to separate the uncut crop versus the empty field (including cut crops), or even to estimate of the amount of crops. Later in this section, two segmentation methods are evaluated for this task, and a texture-based crop-presence classifier is developed to estimate the fraction of remaining crops.

6.2. Field region analysis

The presented header prediction system is shown in Fig. 7. The input raw videos are processed based on a block of frames with length 30. After applying spatial segmentation, the upcoming field region is highlighted, and a crop-presence classifier is used to separate the uncut crop and empty field. Based on the classification results, each video frame

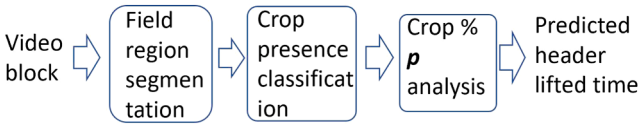


Fig. 7. The combine harvester header prediction pipeline.

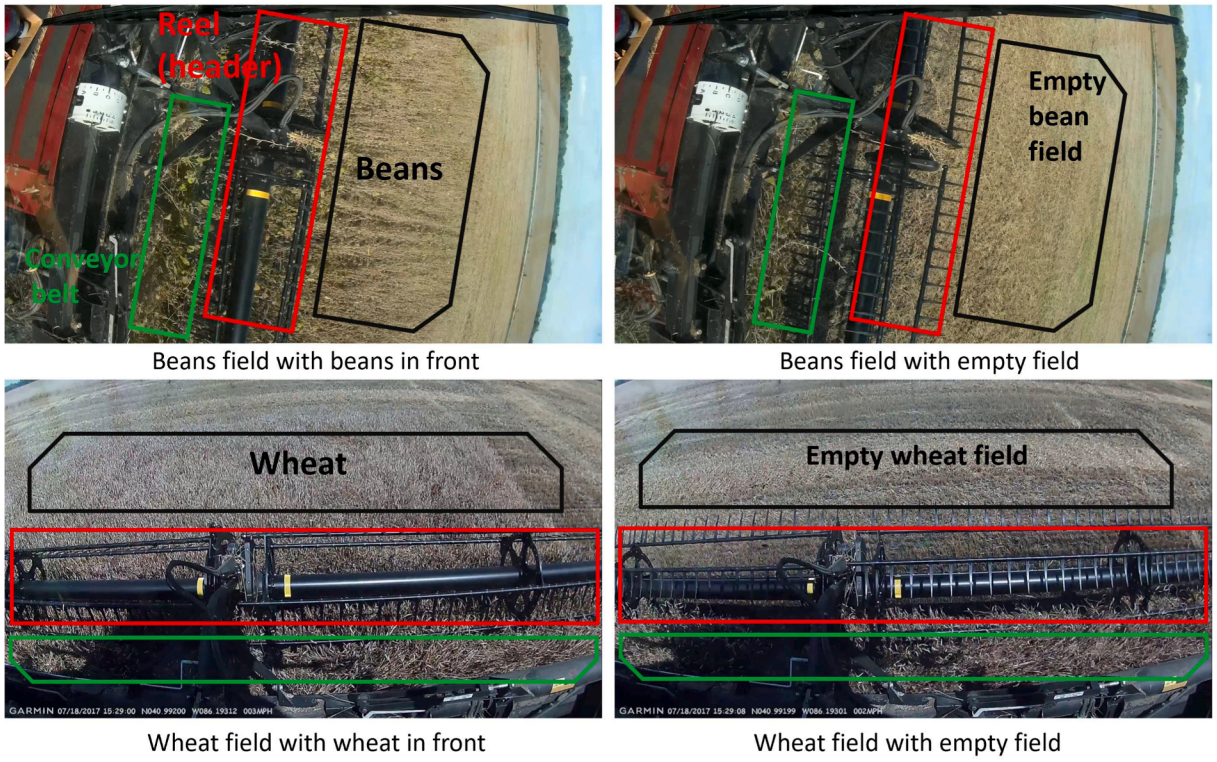


Fig. 6. Sample frames captured by dash cameras mounted on combine harvester in bean field (top) and wheat field (bottom). The left and right images show the crop field and empty field.

produces one crop percentage value p. By analyzing a series of percentage values, we can estimate the final time when the header should be lifted.

The goal of the crop-presence classifier is to separate the crops from the empty field and estimate the crops percentages. In our design, we segment the field regions into smaller units for the classifier and then calculate the total crops percentages. As a result, the field region is divided into squares with length L based on the segmentation, and all neighboring squares overlap. We use overlapping squares because each square might not provide a complete view of the crop, and over-segmenting the field region can preserve the shape of crops. This overlap also means that each pixel is covered by multiple squares, which improves the accuracy of crop amount estimation.

To classify the squares between crop and empty field, the GLCM (Baraldi and Parmiggiani, 1995) texture feature is used. This feature captures four possible directions of neighboring pixel pairs in each square. Histograms of contrast and homogeneity are measured as the feature vector. Notice that all the combine harvesting videos we collected have a resolution of 1920×1080, and these higher resolution images enable the texture feature to perform well. The texture features are used to train a random forest classifier, and based on the classified squares, each pixel receives a probability representing how likely it is to be a crop instead of empty field. Then for each video frame, a final crop percentage value p is estimated using the weighted sum of all the probabilities of pixels in the upcoming field region.

After estimating all the frames in a video block, a series of crop percentage values p_t are generated. If the header of the combine harvester needs to be lifted, this percentage value should be decreasing. To capture the possible decreasing percentages, a Sigmoid function (4) is used to fit p_t with respect to time t:

$$\widehat{p}_t = \frac{s_0}{1 + e^{-\frac{s_1}{t - s_2}}} + s_3 \tag{4}$$

where all the s_i are parameters, $s \in \{0,1,2,3\}$. This Sigmoidal function is a natural choice given that the percentage of crop will monotonically decrease. The parameter s_2 controls the temporal position where the function decreases, s_1 indicates the dropping curvature, and s_0, s_3 normalize the range into 0 to 100. The fitted curve represents the decrease of crop percentage, which allows us to estimate the percentage value at a future time $\widehat{p}_{t+\Delta T}$. As a result, the time when the header should be lifted t_{up} can be estimated by Eq. (5),

$$t_{up} = arg(\widehat{p}_t \to 0) + T_{delay} \tag{5}$$

which predicts the time when the crop percentage will decrease to 0. The T_{delay} is a constant time and it represents the delay between the time when there are no crops in the camera until the combine actually harvests the crops.

6.3. Header prediction experiment

We use both wheat harvesting and soybean harvesting videos (shown in Fig. 1) in this header prediction experiment. The crop-presence classifier is trained from the manually labelled masks used in the segmentation experiment in Section 4. These ground truth masks are further labelled into crops and empty fields to train the crop-presence classifier. In the implementation, the square length L is 100 pixels, the overlap between squares is $L/2$, and the crop-presence classifier is trained from multiple harvesting clips.

When preparing the testing videos for header prediction, the raw harvesting clips are manually segmented into transition clips. All these clips contain the moment when the farmer lifts the header. Each transition clip is 10 s (300 frames) long with the transition happening around frame 250 to 280. We also select some normal harvesting videos (without header lifted) as negative clips. In total, we prepare 37 positive transition clips and 38 negative clips.

We begin by comparing the performance of different classifiers on classifying each square as having been cropped or not-yet cropped. Four methods are compared: a decision tree classifier, a Support Vector Machine (SVM) classifier, an Adaboost classifier, and our applied random forest classifier. Labelled crop and non-crop squares are partitioned into training and testing sets and we compute the prediction accuracy. In this experiment, the random forest method achieves 93.3%, followed by Adaboost classifier with 92.2%, decision tree with 89.8%, and the 60.3% for the SVM. Subsequent experiments all use the random forest classifier.

Next, we compare three segmentation methods from Section 4: the fine-tuned Deeplab (Chen et al., 2017)*, our previous method (Liu et al., 2019)*, and the proposed random forest method *RF_motion*. To measure the performance of this system, we first evaluate whether the system can make a correct decision when to lift the header. This measure is based on the difference between the average value of $\widehat{p}_t$ of the first N frames and the average value of the last N frames. Here N is the number of frames that the header is stabilized to either high or low position, which is 30 frames (1 s) in the experiment. The accuracy is computed by the ratio of correctly predicted clips. If the header needs to be lifted, the system also predicts the time when the header needs to be lifted. So another measure is the averaged error time between the actual rising time and the predicted T_{up}.

Fig. 8 shows the result of the crop-presence classifier using the segmentation in Liu et al. (2019)* on a soybean-harvesting video. The blue color indicates regions that are more likely to have crop and the orange regions represents the empty field. Note that this machine is driving toward to the right side, which is the end of a row in the field. As the frame number increases, the area of crop (blue) is shrinking to the left (frame 150 and 180 in Fig. 8), and disappears in the end. It can be observed that the classifier is applied on squares, and the light blue squares in frame 180 shows the uncertainty of the classifier. Each frame produces one percentage value p, and after fitting these values with time, the future crop percentage can be estimated.

On the same soybean harvesting clip, the crop percentage curves estimated by three segmentation methods are shown in Fig. 9. All three curves present the decreasing crop percentages, but the decreasing moment of the Deeplab* (Chen et al., 2017) curve is difficult to detect. One possible reason could be the segmentation method miss-classifies other blurry regions such as the sky or some far-away blurry fields. The texture features in those regions confuse the crop percentage estimation which leads to the flat curve. The other two methods provide clear decreasing curves which can be easily interpreted to create a header control signal.

The quantitative comparison between the segmentation methods is shown in Table 4. It can be observed that apart from the Deeplab (Chen et al., 2017)*, the other two methods have almost the same performance on both decision accuracy and averaged error time. But for real applications, no incorrect prediction is allowed. So for further improvement, this system needs to have more accurate spatial segmentation results for the field region. Also the crop-presence classifier needs to gain robustness by training data from various farms.

An extra experiment is performed to evaluate the sensitivity of the parameters of the fitting function in Eq. (4). All four s_i values are adjusted to $s_i \pm 5\%$ and we perform the same evaluation as in Table 4. The results show that only the variation of s_2 cause 1% variation compared to the original results in Table 4, and all other parameter variations result in no changes. This indicates that the method is robust to small perturbations generated by the parameters in the fitting model.

7. Conclusion

This paper introduces the role of spatial segmentation in processing farming videos. Unlike typical video object segmentation, the goal of segmenting farming videos is to partition the frame into different regions. The practical farming environment also requires the system to be

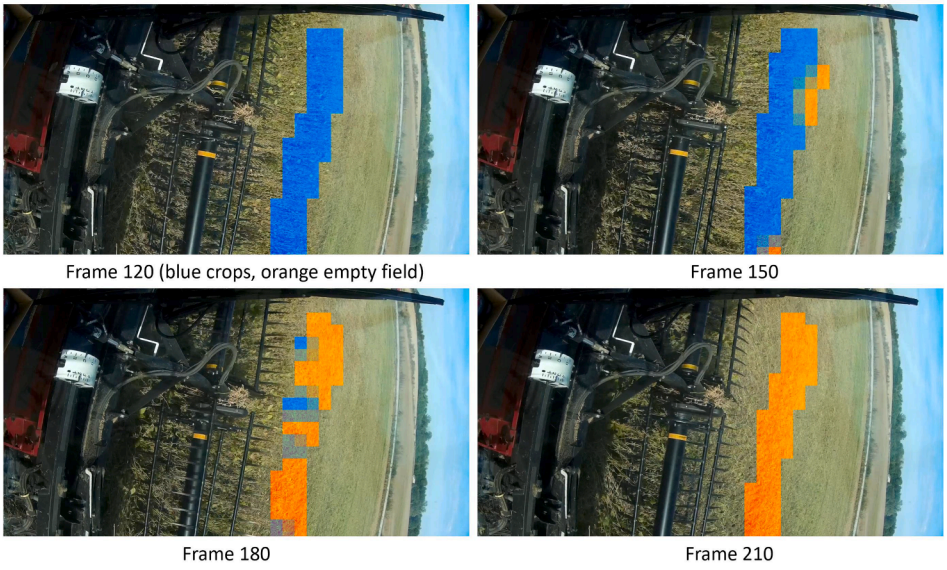


Fig. 8. Example output of the crop-presence classifier. The color represents the probability of crops: blue indicates crops, and orange indicates empty field.

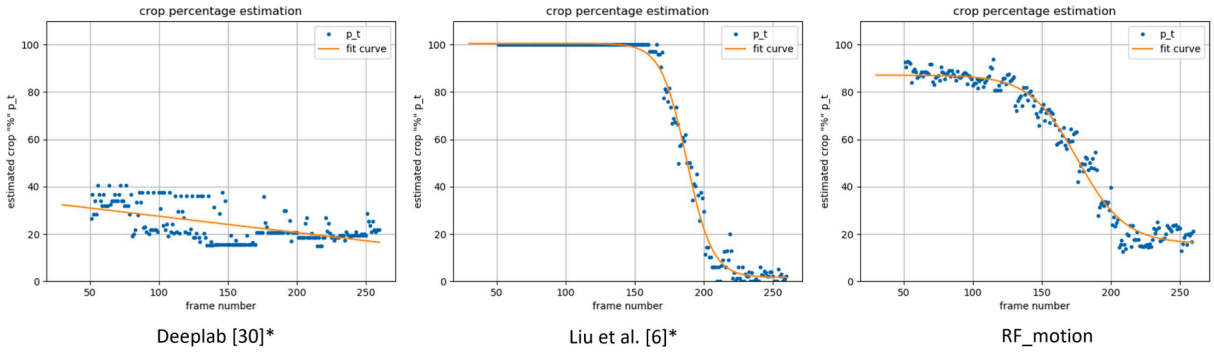


Fig. 9. The crop amount prediction comparison on an example video clip. The X axis shows the frame number, and Y axis is the estimated crop percentage.

Table 4
The comparison of three segmentation methods in combine header prediction. Notice all videos are 30 frame per second.

Method	Prediction accuracy	Averaged error frames
Deeplab (Chen et al., 2017)*	0.778	121.04
Liu et al. (2019)*	0.984	40.67
RF_motion	0.953	66.72

both computationally efficient and easily adaptable to different farming applications. A training-based segmentation method is introduced using a random forest classifier. It extracts basic color and motion features, and the system can be quickly trained on machines with limited computational power.

In addition, two video-based farming applications are presented, each focusing on a different region of the image. First, we develop a generalized two-branch pipeline for farming video classification. The system uses a general video classifier as Branch 1 and merges a specifically-designed classifier in Branch 2 based on domain knowledge. When applying this system to classify farming activity videos, the second branch selects features from the attachment region and improves the overall performance. In addition to farming, this two-branch pipeline can be further applied to video classification problems for other areas with domain knowledge involved.

Another application we consider is to predict and control the header-height of a combine harvester. This system uses the segmentation results and focuses on analyzing the upcoming field region. Based on crop presence classification, the system estimates the crop amount in the field, which can be further parsed to indicate and adjust the combine header-height. The results show that the crop percentages in the field regions can be successfully detected, and the sensitivity test shows the model to identify the decreasing percentage is robust to minor crop detection errors.

In this paper, we have considered data from only a few farms in the US. While performance is promising on these data, future work is needed on a wider variety of farms and crops, to validate and establish robustness of our methods. Future work will also incorporating our ideas into other vision-based harvesting applications, like fine-tuned control of the header height to maintain uniform stubble height despite varying terrain. Finally, we will apply our general processing pipeline to additional applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Baraldi, A., Parmiggiani, F., 1995. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. IEEE Trans. Geosci. Remote Sens. 33 (2), 293–304.

Benson, E., Reid, J., Zhang, Q., 2003. Machine vision-based guidance system for agricultural grain harvesters using cut-edge detection. Biosyst. Eng. 86 (4), 389–398.

Boursianis, A.D., Papadopoulou, M.S., Diamantoulakis, P., Liopa-Tsakalidi, A., Barouchas, P., Salahas, G., Karagiannidis, G., Wan, S., Goudos, S.K., 2020. Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: a comprehensive review. Internet Things 100187.

Chaquet, J.M., Carmona, E.J., Fernández-Caballero, A., 2013. A survey of video datasets for human action and activity recognition. Comput. Vis. Image Underst. 117 (6), 633–659.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Chen, J., Wang, Y., Wang, X., Wang, Y., Hu, R., 2017. Development and application of remote video monitoring system for combine harvester based on embedded linux. In: Seventh International Conference on Electronics and Information Engineering, vol. 10322. International Society for Optics and Photonics, p. 1032223.

Cho, H., Seo, Y.-W., Kumar, B.V., Rajkumar, R.R., 2014. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In: IEEE International Conference on Robotics and Automation, pp. 1836–1843.

Dollár, P., Rabaud, V., Cottrell, G., Belongie, S., 2005. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72.

Faktor, A., Irani, M., 2014. Video segmentation by non-local consensus voting. In: British Machine Vision Conference, vol. 2, p. 8.

Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. Int. J. Comput. Vision 59 (2), 167–181.

Gupta, M., Abdelsalam, M., Khorsandroo, S., Mittal, S., 2020. Security and privacy in smart farming: Challenges and opportunities. IEEE Access 8, 34564–34584.

Hara, K., Kataoka, H., Satoh, Y., 2018. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?, in. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6546–6555.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90.

Karegowda, A.G., Devika, G., Geetha, M., 2021. Deep learning solutions for agricultural and farming activities. In: Deep Learning Applications and Intelligent Decision Making in Engineering, IGI Global, pp. 256–287.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al., 2017. The kinetics human action video dataset, arXiv preprint arXiv:1705.06950.

King, A., et al., 2017. The future of agriculture. Nature 544 (7651), S21–S23.

Kounalakis, T., Triantafyllidis, G.A., Nalpantidis, L., 2019. Deep learning-based visual recognition of rumex for robotic precision farming. Comput. Electron. Agric. 165, 104973.

Kragh, M.F., Christiansen, P., Laursen, M.S., Larsen, M., Steen, K.A., Green, O., Karstoft, H., Jørgensen, R.N., 2017. Fieldsafe: dataset for obstacle detection in agriculture. Sensors 17 (11), 2579.

Krapac, J., Verbeek, J., Jurie, F., 2011. Modeling spatial layout with Fisher vectors for image categorization. In: IEEE International Conference on Computer Vision. IEEE, pp. 1487–1494.

Kurita, H., Iida, M., Suguri, M., Masuda, R., 2012. Application of image processing technology for unloading automation of robotic head-feeding combine harvester. Eng. Agric. Environ. Food 5 (4), 146–151.

Liu, H., Reibman, A.R., Ault, A.C., Krogmeier, J.V., 2018. Video classification of farming activities with motion-adaptive feature sampling. In: IEEE International Workshop on Multimedia Signal Processing, pp. 1–6.

Liu, H., Reibman, A.R., Ault, A.C., Krogmeier, J.V., 2019. Video-based prediction for header-height control of a combine harvester. In: IEEE Conference on Multimedia Information Processing and Retrieval, pp. 310–315.

Liu, H., Reibman, A.R., Boerman, J.P., 2020. Video analytic system for detecting cow structure. Comput. Electron. Agric. 178, 105761.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

Onofri, L., Soda, P., Pechenizkiy, M., Iannello, G., 2016. A survey on using domain and contextual knowledge for human activity recognition in video streams. Expert Syst. Appl. 63, 97–111.

Pal, M., 2005. Random forest classifier for remote sensing classification. Int. J. Remote Sens. 26 (1), 217–222.

Papandreou, G., Chen, L.-C., Murphy, K.P., Yuille, A.L., 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proceedings of the IEEE International Cnference on Computer Vision, pp. 1742–1750.

Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B., 2017. Learning features by watching objects move. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6024–6033.

Peng, X., Wang, L., Wang, X., Qiao, Y., 2016. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. Comput. Vis. Image Underst. 150, 109–125.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A., 2016. A benchmark dataset and evaluation methodology for video object segmentation, in. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 724–732.

Petscharnig, S., Schöffmann, K., 2018. Learning laparoscopic video shot classification for gynecological surgery. Multimedia Tools Appl. 77 (7), 8061–8079.

Poleg, Y., Arora, C., Peleg, S., 2014. Temporal segmentation of egocentric videos, in. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2537–2544.

Sainz-Costa, N., et al., 2011. Mapping wide row crops with video sequences acquired from a tractor moving at treatment speed. Sensors 11 (7), 7095–7109.

Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576.

Uijlings, J., Duta, I.C., Sangineto, E., Sebe, N., 2015. Video classification with densely extracted HOG/HOF/MBH features: an evaluation of the accuracy/computational efficiency trade-off. Int. J. Multimedia Informat. Retrieval 4 (1), 33–44.

Wang, H., Schmid, C., 2013. Action recognition with improved trajectories. In: Proceedings of the IEEE Conference on Computer Vision, pp. 3551–3558.

Wang, W., Shen, J., Yang, R., Porikli, F., 2017. Saliency-aware video object segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 40 (1), 20–33.

Xie, Y., Alleyne, A., 2012. Two degrees of freedom control for combine harvester header height control. In: ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference. American Society of Mechanical Engineers, pp. 539–547.

Xie, Y., Alleyne, A.G., Greer, A., Deneault, D., 2013. Fundamental limits in combine harvester header height control. J. Dyn. Syst. Meas. Contr. 135 (3), 034503.

Yue, J., Li, Z., Liu, L., Fu, Z., 2011. Content-based image retrieval using color and texture fused features. Mathe. Comput. Modell. 54 (3–4), 1121–1127.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A., 2018. Places: A 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. 40 (6), 1452–1464.