ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 167

International Scientific and Practical Conference "Digitization of Agriculture - Development Strategy" (ISPC 2019)

Detecting uncut crop edge with convolutional neural networks

Denis Protasov
Scientific and Production Association
of automatics named after academician N.A.Semikhatov
Ekaterinburg, Russia
protasovdenisn@gmail.com

***Abstract*—The paper proposes an approach to determining the boundary between cut and uncut crop, which is one of the most important tasks in creating a system of assistance to the combine operator. The approach is based on semantic segmentation of images that come from a camera installed in the combine cab. Semantic segmentation is performed using the ENet neural network, which is designed to work in real time. As a result of segmentation, 5 classes are recognized: cut crop, uncut crop, obstacles, harvester part, and background. The straight line of the boundary between cut and uncut crop is determined through linear regression. Testing of the performance of algorithm was carried out both on conventional CPU and on mobile ones. An implementation for mobile processors has been created that provides a performance of 3.61 FPS on an ARM Cortex-A53 processor. The algorithm's performance is sufficient to let the combine drive at 4 km/h.**

Keywords—advanced driver assistance systems, agricultural harvester, convolutional neural networks, semantic segmentation

I. INTRODUCTION

Human activities in the field of agriculture are gradually being automated. One of the promising areas of automation is the harvesting process. The increase in computing power has now made it possible to use computer vision algorithms to extract navigation parameters from a video stream in real time.

Currently, the harvesting is carried out using combine harvesters controlled by operators. The operator's task is to drive the combine along an edge between cut and uncut crop, avoid obstacles, move between work areas on a field, and control the unloading of the harvested crop. The computer vision subsystem is able to increase the accuracy of the movement of the combine along the border between cut and uncut crop in semi-automatic mode. The purpose of this paper is to describe an algorithm for detecting this border.

The process of selection of navigation parameters is difficult because of the features of the images, which include image irregularities, dynamically changing lighting, and a large number of objects. Also, to reduce the cost of the system as a whole, it is necessary to use a cost-effective low-cost CPU to perform all the calculations.

This paper discusses the use of convolutional neural network technology to perform semantic segmentation and subsequent determination of the boundary between the cut and uncut crop. The proposed solution is designed to work in real time on an inexpensive mobile processor.

II. BACKGROUND AND RELATED WORK

For the system of automatic driving in the field of agriculture, there are solutions using various sensors [1]. Ultrasonic sensors, lidars, sensors operating with GPS global positioning systems [2], accelerometers, odometers, and other kinds of sensors can be used. Also, in such systems, video cameras can be used [3]. Some papers discuss the joint use of video cameras and lidars [4],[5].

Recently, the use of cameras is becoming more and more popular in the systems associated with autonomous driving. First of all, this is due to the large amount of useful information that can be obtained from the camera; secondly, it is a result of increasing performance of mobile computing systems. As concerns useful information from the camera, one can obtain data on the types of objects in the field of view [6], their position relative to each other [7], and their sizes. In case of using a stereo camera, one can calculate the distance to objects [8].

In autonomous driving, the tasks of algorithms are to determine the line separating the cut and uncut crop or to determine the boundaries of the crop row located in front of the combine [9]. The main disadvantages of existing algorithms are their dependence on weather conditions and time of day, and also low performance..

Algorithms using convolutional neural networks exhibit relatively low sensitivity to slight visual noise in images [10]. In particular, such algorithms are able to cope with the task of semantic segmentation [11].

Using the results of semantic segmentation, the existing algorithm [12] selects navigation parameters from images. This approach uses a modified U-Net [13] architecture and works exclusively on GPU. This paper proposes the use of the ENet [14] architecture, which has a higher image processing speed without loss of segmentation quality.

The ENet neural network is designed to work with the CamVid [15], Cityscapes [16], and SUN RGBD [17] databases. The characteristics of the listed databases are presented in Table I.

TABLE I. DATASET CHARACTERISTICS

Dataset	Input resolution	Number of classes
CamVid	480×360	11
Cityscapes	1024×512	19
SUN RGBD	256×200	37

a b

Fig. 1. Frames obtained from Mystery (a) and Mivue (b) DVRs

The ENet architecture is presented in Table II. The neural network is an encoder-decoder structure. ENet consists of several stages, which are separated by horizontal lines in the table. The first stage consists of 5 blocks, while stages 2 and 3 have the same structure, except that stage 3 does not downsample the input at the beginning. The first three stages refer to the encoder, and the rest to the decoder.

ENet largely inherits the architecture of ResNet neural networks [18] which are reduced to a structure with one main branch and extensions, and then combined by elementwise addition. Each block consists of three convolutional layers: a 1×1 projection, a main convolutional layer, and a 1×1 expansion. As in the original article, these blocks are called bottleneck modules. Between all the convolutional layers there are the Batch Normalization [19] and PReLU [20] functions. If a downsampling operation is performed in the bottleneck block, the max pooling layer is added to the main branch. The last layer has a parameter C, which is equal to the number of classes.

TABLE II. ENET ARCHITECTURE [14]

Name	Type	Output size
initial		16 × 256 × 256
bottleneck1.0	downsampling	64 × 128 × 128
4 × bottleneck1.x		64 × 128 × 128
bottleneck2.0	downsampling	128 × 64 × 64
bottleneck2.1		128 × 64 × 64
bottleneck2.2	dilated 2	128 × 64 × 64
bottleneck2.3	asymmetric 5	128 × 64 × 64
bottleneck2.4	dilated 4	128 × 64 × 64
bottleneck2.5		128 × 64 × 64
bottleneck2.6	dilated 8	128 × 64 × 64
bottleneck2.7	asymmetric 5	128 × 64 × 64
bottleneck2.8	dilated 16	128 × 64 × 64
Repeat section 2, without bottleneck2.0		
bottleneck4.0	upsampling	64 × 128 × 128
bottleneck4.1		64 × 128 × 128
bottleneck4.2		64 × 128 × 128
bottleneck5.0	upsampling	16 × 256 × 256
bottleneck5.1		16 × 256 × 256
fullconv		C × 512 × 512

III. METHOD

The determination of the border between cut and uncut crop using the machine vision subsystem is carried out in following stages. At the first stage, a frame is obtained from the camera that is installed in the combine cab. The next step is semantic segmentation of the frame. The segmentation is performed for objects of five classes: obstacles, cut crop, uncut crop, harvester part, and background. The cut and uncut crop classes are directly used to solve the task. The obstacle class is necessary to determine the location of the obstacles relative to the combine; this class includes agricultural machinery, automobiles, wells, power line poles, people, and animals. The harvester part class has been added to ensure that the algorithm does not view parts of the combine as obstacles. The background class contains everything that does not gets into the main classes. The segmentation is performed by the ENet neural network. To train this neural network, we prepared our own database, which is described in Section IV. The training process is described in Section V. Lastly, based on the result of the analysis of the cut and uncut crop classes, their boundary is determined based on the segmented frame. The data obtained in the process of analyzing the position of the boundary can be used to issue a control signal to the main control unit responsible for the movement of the combine.

IV. DATASET

The author is not aware of any open-access database for combine harvesting, therefore, a private database was created for training the neural network. A video was recorded during the operation of the combine, which harvests corn of different planting densities.

Two DVRs were used to record the video: Mivue 518 and Mystery 800, which were located in the combine cab on either side of the driver. The resolutions of the DVRs were 1920×1080 and 848×480 pixels, respectively. The total amount of footage was 95.1 GB and the total recording duration was 30.6 hours. Sample images are provided in Fig.1.

The images obtained from the DVRs suffer from radial distortion, which can be seen on Fig.1. An algorithm based on papers [21] and [22] was used to eliminate this effect, which was implemented using the OpenCV computer vision library. The image with radial distortion removed is shown in Fig.2.

Fig. 2. Image with radial distortion

As the video was acquired, the following special scenes were noted: obstacles in the path of the combine, a bend in the line along which the harvest is made, moving to the next harvest area (Fig.3 a), work under power lines (Fig.3 b), work at the end of the field, work with interference in the camera's field of view, etc. As a result of the analysis of the video, a database has been formed that is suitable for training a neural network.

The harvested crop is thrown into the body of an auxiliary agricultural machine that moves near the combine in the process of work; to properly recognize such situation, images with different positions of the auxiliary machine relative to the combine were added to the dataset: only the cabin is visible, the cabin and a part of the body is visible, and the whole vehicle is visible. Since the body of the auxiliary machine periodically fills up during operation, scenes have been added that contain a change of one machine by another and scenes with several machines in the camera's field of view. We have also added scenes in which there were no auxiliary machines. In addition, the following scenes were also added to the database: crop gathering at the end of the field, curvilinear movement of the combine along the working area, crop gathering from the last section (left and right areas are cut, but not the central one), and the changing of the working area. Finally, scenes with a dynamic change in the illumination of the working area and with work in the evening were added to the database since harvesting is conducted throughout the whole day.

For image labeling, a program for semi-automatic segmentation was developed. The program allows to sequentially label the images frame by frame. First, the user labels the areas belonging to different classes using different color markers: obstacles (red), cut/uncut crop (yellow/green), harvester part (purple), background (blue) (Fig.4 a). Next, the areas belonging to different classes are automatically filled with solid color using the watershed algorithm (Fig.4 b).

To simplify the learning process of the neural network, the labeled images are converted to grayscale format, which reduces the number of color channels from three to one. When performing this operation, there were problems with the jpg format: color gradient on the border of two colors and the presence of noise in the image. This is typical of the jpg format, so it was decided to work with the png format. For a comparison of images in these formats, see Fig.5.

The resulting dataset for neural network training contains 2474 images. We divided it into three parts: training, validation, and test (70%, 20% and 10% respectively). Images from the training set were used to train the neural network. The validation set contained the images that were used to test the segmentation quality by the neural network in the training process. The final evaluation of segmentation quality was conducted on the test set.

V. TRAINING OF NEURAL NETWORK

To train our neural network, we used supervised learning. The ENet neural network architecture has been adjusted to work with images of different resolutions. The input of the neural network was fed a pair of images, the original photo and the labeled one. We used the Spatial Cross Entropy as the error function and Accuracy as a measure of the quality. The following hyperparameters were used during the training:

- Optimizer: adam
- Batch Size: 16
- L2 Weight Decay: 0.0002
- Learning Rate: 0.0005
- Momentum: 0.9

The neural network was trained on three Nvidia GTX 1080 Ti GPU using CUDA technology. The total amount of video memory required for training was 15163 MB. For input images at 640×480 resolution, the training time was 4 hours.

The training was carried out in two stages. The encoder was trained for 30 epochs. After that, the weights with the highest accuracy were selected. The best accuracy was at the 25th epoch of the training encoder. Then, the chosen weights were used to train the decoder, which was trained during 100 epochs, and the best accuracy was at the 95th epoch. The segmentation accuracy for different resolutions is given in Table III.

Figures 6-7 show the graphs of the dependence of the error value on the number of epochs for the encoder and decoder during training.

The result of the trained neural network is shown in Fig.8. The network has segmented the image into the key classes: cut crop, uncut crop, and obstacles (here, a truck); for convenience, this segmentation is shown superimposed over the original image.

TABLE III. COMPARISON OF THE ACCURACY OF DETERMINING THE CLASSES AT DIFFERENT RESOLUTIONS OF IMAGES

Resolution	**640×480**	**480×360**	**320×240**	**160×120**
Obst	98.07	97.55	97.64	96.63
Uncut crop	96.27	96.27	96.19	96.29
Cut crop	95.21	95.32	94.13	93.57
Harv. part	96.37	94.93	96.49	95.50
Background	99.27	99.21	99.24	98.60
Avg. acc.	97.04	96.65	96.74	96.12

Fig. 3. Moving to the next harvest area (a). Work under power lines (b) DVRs

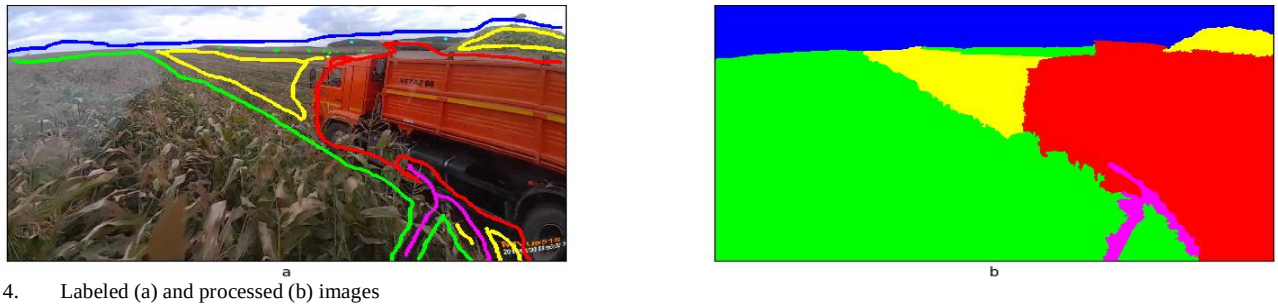

Fig. 4. Labeled (a) and processed (b) images

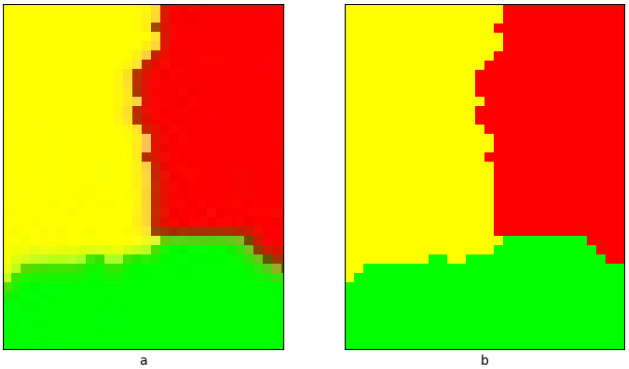

Fig. 5. Comparison of jpg (a) and png (b) images

VI. Algorithm for Detecting the Border Between Cut and Uncut Crop

The segmentation between the cut and uncut crop classes yields a curved line (Fig.8), however, for correct movement of the combine, a straight path is required. Therefore, the algorithm analyzes the extreme points of the boundary and forms a straight line (Fig.9).

The algorithm works as follows: first, key points are determined that belong to the boundary between cut and uncut crop (Fig.10). Then, linear regression is applied to the obtained points to build a straight line.

To test the accuracy of the algorithm, a dataset was composed of images of different resolutions. For each image, the boundary between the classes of interest was painted by hand by 5 people; it was then averaged. The quality of the algorithm was estimated by the modified mean squared error metric. The results of the test are presented in Table IV.

TABLE IV. The error of detecting the border between the cut and uncut crop

	640×480	480×360	320×240	160×120
Error	9.29	8.88	7.89	6.83

VII. Embedded Implementation

A distinctive feature of the algorithm is the ability to work without the use of GPU resources. Thus, the algorithm can work on personal computers, single-board computers, and mobile devices. To measure the performance of the algorithm, the number of frames processed per second (FPS) was counted. Measurements were taken from the moment of receipt of the frame from the camera to the moment of transfer of data on the position of the border to the main control computer. Testing of the performance was carried out both on conventional CPU and on mobile ones. The results of the test are presented in Table V.

To speed up the algorithm on ARM architecture processors, the OpenCV library was compiled with support for the NEON and VFPV3 command sets.

TABLE V. Performance comparison

CPU	640×480	480×360	320×240	160×120
Intel Core I5-7400	4.71	9.85	19.41	38.47
Intel Core I5-3230M	3.44	5.51	13.92	35.24
ARM Cortex-A53	0.28	0.46	1.13	3.61
ARM Cortex-A7	0.21	0.38	0.82	2.74

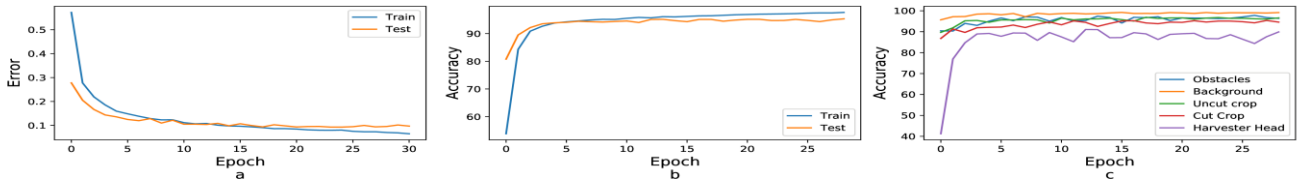

Fig. 6. Graphs of changes in loss (a), accuracy (b), and per class accuracy (c) during the training of encoder

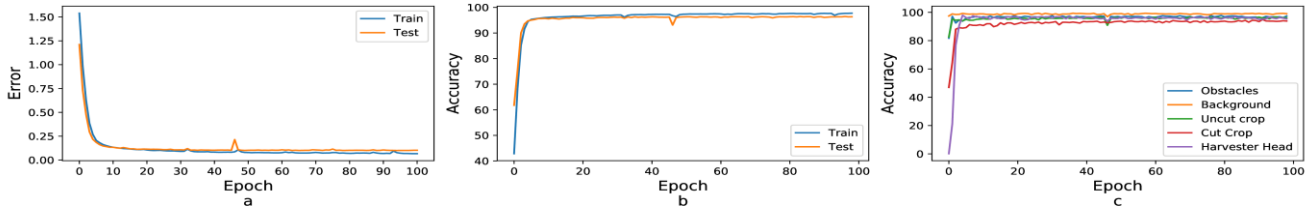

Fig. 7. Graphs of changes in loss (a), accuracy (b), and per class accuracy (c) during the training of decoder

Fig. 8. The result obtained from the neural network

Fig. 9. Border detection

Fig. 10. Key points detection

VIII. Discussion

Based on the obtained results, it was decided to use images in the 160×120 resolution. At this resolution, the accuracy is sufficient for determining the key classes and it is enough to form the line of motion of the combine.

During harvest, the combine can move at speeds from 3 to 7 km/h. The algorithm's performance on the ARM Cortex-A53 processor is sufficient to let the combine drive at 4 km/h. To drive the combine faster, a more powerful CPU, such as ARM Cortex-A15, can be used.

IX. Conclusion

The paper presents an approach to determining the boundary between the cut and uncut crop, which can be used in combine driver assistance systems. In contrast to existing system [12], our approach uses the ENet neural network, which is designed for real-time semantic segmentation. The

developed solution allows to get 3.61 FPS border detection performance on an ARM Cortex-A53 processor. The resulting performance is sufficient for the combine to work at a speed of 4 km/h. Further work directions include increasing the speed of segmentation, improving detection of obstacles in the path of the combine, and growing the database to ensure a stable operation of the algorithm in different conditions.

ACKNOWLEDGMENT

The author is grateful to Andrei Vladimirovich Sozykin and Boris Vladimirovich Shulgin for their assistance in writing this paper.

REFERENCES

[1] N. Shalal, T. Low, C. McCarthy, and N. Hancock, ``A review of autonomous navigation systems in agricultural environments," Soc. Agric. Eng. (SEAg): Innovative Agric. Tech. for a Sustainable Future. Barton, Australia, 2013.

[2] B. Thuilot, C. Cariou, L. Cordesses, and P. Martinet, ``Automatic guidance of a farm tractor along curved paths, using a unique CP-DGPS," Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, p. 674-679, 2001.

[3] R. Gottschalk, X. P. Burgos-Artizzu, A. Ribeiro, and G. Pajares, ``Real-time image processing for the guidance of a small agricultural field inspection vehicle," International journal of intelligent systems technologies and applications, 8: 434-443, 2010.

[4] V. Subramanian, T. F. Burks, and A. Arroyo, ``Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation," Computers and Electronics in Agriculture, 53: 130-143, 2006.

[5] F. Auat Cheein, G. Steiner, G. Perez Paina, and R. Carelli, ``Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection," Computers and Electronics in Agriculture, 78: 195-207, 2011.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ``ImageNet classification with deep convolutional neural networks," In Proceedings of the Advances in Neural Information Processing Systems, December 2012.

[7] Z. Zhong-Qiu, Z. Peng, X. Shou-tao, and W. Xindong, ``Object Detection with Deep Learning: A Review," arXiv preprint arXiv:1807.05511, 2018.

[8] H. Hirschmüller, ``Stereo Processing by Semi-Global Matching and Mutual Information," in IEEE Transactions on Pattern Analysis and Machine Intelligence. 30. 328-341, 2008.

[9] R. Gottschalk, X. P. Burgos-Artizzu, A. Ribeiro, and G. Pajares, ``Real-time image processing for the guidance of a small agricultural field inspection vehicle," International journal of intelligent systems technologies and applications, 8: 434-443, 2010.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ``Gradient-based learning applied to document recognition," Proceedings of the IEEE. 86(11): p. 2278-2324, 1998.

[11] J. Long, E. Shelhamer, and T. Darrell, ``Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[12] A.Y. Shkanaev, D.A. Krokhina, D.V. Polevoy, A. Panchenko, D. Sholomov, and R. Sadekov, ``Analysis of straw row in the image to control the trajectory of the agricultural combine harvester," Proceedings of SPIE - The International Society for Optical Engineering. 10696. 10.1117/12.2310143, 2018.

[13] O. Ronneberger, P. Fischer, and T. Brox, ``U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv preprint arXiv:1505.04597, 2015.

[14] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, ``ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," arXiv preprint arXiv:1606.02147, 2016.

[15] G. J. Brostow, J. Fauqueur, and R. Cipolla. ``Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters," 30(2):88–97, 2009.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, ``The cityscapes dataset for semantic urban scene understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[17] S. Song, S. P. Lichtenberg, and J. Xiao, ``Sun rgb-d: A rgb-d scene understanding benchmark suite," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 567–576, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385, 2015.

[19] S. Ioffe and C. Szegedy, ``Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, ``Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," pp. 1026–1034, 2015.

[21] Z. Zhang. ``A Flexible New Technique for Camera Calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

[22] Y. Bouguet, J. Matlab camera calibration toolbox. IEEE Transactions on Reliability - TR, 2005.